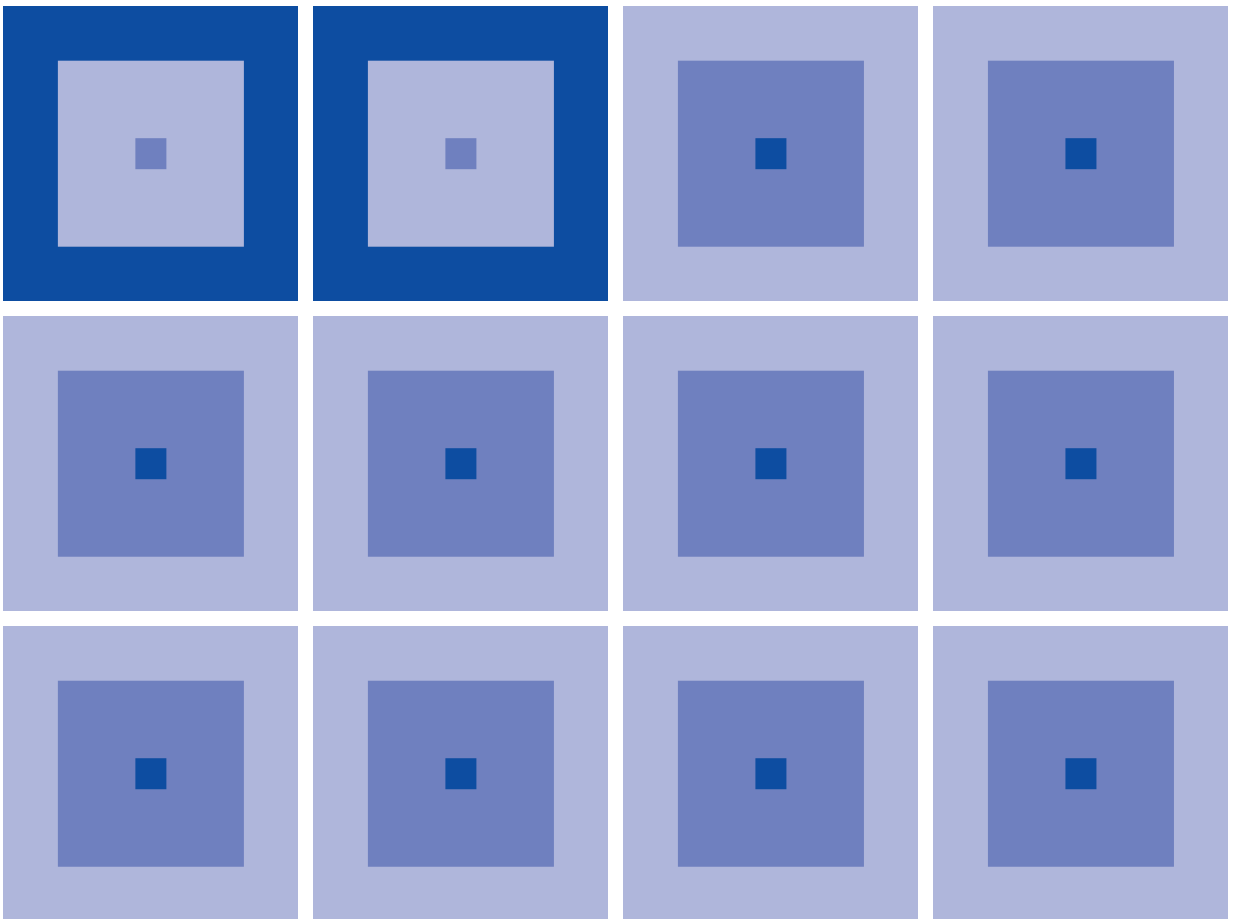


CMOS 32-BIT SINGLE CHIP MICROCOMPUTER

# S1C33301

## Technical Manual



**NOTICE**

---

*No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from another government agency.*

## Configuration of product number

### Devices

S1    C    33209    F    00E1    00

#### Packing specifications

[00 : Besides tape & reel  
 0A : TCP BL    2 directions  
 0B : Tape & reel BACK  
 0C : TCP BR    2 directions  
 0D : TCP BT    2 directions  
 0E : TCP BD    2 directions  
 0F : Tape & reel FRONT  
 0G : TCP BT    4 directions  
 0H : TCP BD    4 directions  
 0J : TCP SL    2 directions  
 0K : TCP SR    2 directions  
 0L : Tape & reel LEFT  
 0M : TCP ST    2 directions  
 0N : TCP SD    2 directions  
 0P : TCP ST    4 directions  
 0Q : TCP SD    4 directions  
 0R : Tape & reel RIGHT  
 99 : Specs not fixed

#### Specification

#### Package

[D: die form; F: QFP]

#### Model number

#### Model name

[C: microcomputer, digital products]

#### Product classification

[S1: semiconductor]

### Development tools

S5U1    C    33000    H2    1    00

#### Packing specifications

[00: standard packing]

#### Version

[1: Version 1]

#### Tool type

[Hx : ICE  
 Dx : Evaluation board  
 Ex : ROM emulation board  
 Mx : Emulation memory for external ROM  
 Tx : A socket for mounting  
 Cx : Compiler package  
 Sx : Middleware package]

#### Corresponding model number

[33L01: for S1C33L01]

#### Tool classification

[C: microcomputer use]

#### Product classification

[S5U1: development tool for semiconductor products]



# S1C33301 Technical Manual

<b>I S1C33301</b>	<b>I</b>
I-1 OUTLINE	Outline
I-2 BLOCK DIAGRAM	Block
I-3 PIN DESCRIPTION	Pin
I-4 POWER SUPPLY	Power
I-5 INTERNAL MEMORY	RAM
I-6 ELECTRICAL CHARACTERISTICS	E char
I-7 PACKAGE	Package
I-8 BASIC EXTERNAL WIRING DIAGRAM	Wiring
I-9 PRECAUTIONS ON MOUNTING	Mount
<b>II CORE BLOCK</b>	<b>II</b>
II-1 INTRODUCTION	Intro
II-2 CPU AND OPERATING MODE	CPU
II-3 INITIAL RESET	Reset
II-4 BCU (Bus Control Unit)	BCU
II-5 ITC (Interrupt Controller)	ITC
II-6 CLG (Clock Generator)	CLG
II-7 POWER-DOWN CONTROL	P down
II-8 DBG (Debug Unit)	DBG
<b>III PERIPHERAL BLOCK</b>	<b>III</b>
III-1 INTRODUCTION	Intro
III-2 CHIP ID / PIN STATUS CONTROL REGISTERS	PUP
III-3 PRESCALER	PSC
III-4 8-BIT PROGRAMMABLE TIMERS	8TM
III-5 16-BIT PROGRAMMABLE TIMERS	16TM
III-6 WATCHDOG TIMER	WDT
III-7 LOW-SPEED (OSC1) OSCILLATION CIRCUIT	OSC1
III-8 CLOCK TIMER	CTM
III-9 SERIAL INTERFACE	SIF
III-10 SERIAL INTERFACE with FIFO	FSIF
III-11 INPUT/OUTPUT PORTS	I/O
III-12 EXTENDED PORTS	Ext I/O
III-13 SMARTMEDIA INTERFACE	SMIF
<b>IV ANALOG BLOCK</b>	<b>IV</b>
IV-1 INTRODUCTION	Intro
IV-2 A/D CONVERTER	A/D
<b>V DMA BLOCK</b>	<b>V</b>
V-1 INTRODUCTION	Intro
V-2 HSDMA (High-Speed DMA)	HSDMA
V-3 IDMA (Intelligent DMA)	IDMA
<b>Appendix</b>	<b>APP</b>
I/O Map	I/O map



## CONTENTS

**I S1C33301**

<b>I-1 Outline</b> .....	<b>I-1-1</b>
Features .....	I-1-1
<b>I-2 Block Diagram</b> .....	<b>I-2-1</b>
C33 Macro Block Diagram .....	I-2-1
Functional Block Diagram .....	I-2-3
<b>I-3 Pin Description</b> .....	<b>I-3-1</b>
Pin Layout Diagram (plastic package) .....	I-3-1
Pin Functions .....	I-3-3
<b>I-4 Power Supply</b> .....	<b>I-4-1</b>
Power Supply Pins .....	I-4-1
Operating Voltage (V <sub>DD</sub> , V <sub>SS</sub> ) .....	I-4-1
Power Supply for I/O Interface (V <sub>DDE</sub> ) .....	I-4-1
Power Supply for Analog Circuits (AV <sub>DDE</sub> ) .....	I-4-2
Precautions on Power Supply .....	I-4-2
<b>I-5 Internal Memory</b> .....	<b>I-5-1</b>
ROM and Boot Address .....	I-5-2
RAM .....	I-5-2
External Address Bus .....	I-5-2
<b>I-6 Electrical Characteristics</b> .....	<b>I-6-1</b>
Absolute Maximum Rating .....	I-6-1
Recommended Operating Conditions .....	I-6-1
DC Characteristics .....	I-6-2
Current Consumption .....	I-6-3
A/D Converter Characteristics .....	I-6-4
Oscillation Characteristics .....	I-6-5
PLL Characteristics .....	I-6-5
Pin Characteristics .....	I-6-6
AC Characteristics .....	I-6-9
Symbol Description .....	I-6-9
AC Characteristics Measurement Condition .....	I-6-9
C33 Block AC Characteristic Tables .....	I-6-10
C33 Block AC Characteristic Timing Charts .....	I-6-12
<Reference> External Device Interface Timings .....	I-6-16
ROM and Burst ROM .....	I-6-17
SRAM (55ns) .....	I-6-19
SRAM (70ns) .....	I-6-21
8255A .....	I-6-23
<b>I-7 Package</b> .....	<b>I-7-1</b>
Plastic Package .....	I-7-1
Thermal Package Resistance .....	I-7-3
<b>I-8 Basic External Wiring Diagram</b> .....	<b>I-8-1</b>
<b>I-9 Precautions on Mounting</b> .....	<b>I-9-1</b>

**II CORE BLOCK**

---

<b>II-1 INTRODUCTION</b>	<b>II-1-1</b>
<b>II-2 CPU AND OPERATING MODE</b>	<b>II-2-1</b>
CPU .....	II-2-1
Standby Mode .....	II-2-1
HALT Mode .....	II-2-1
SLEEP Mode .....	II-2-1
Notes on Standby Mode .....	II-2-2
Debug Mode .....	II-2-2
Trap Table .....	II-2-2
<b>II-3 INITIAL RESET</b>	<b>II-3-1</b>
Pins for Initial Reset .....	II-3-1
Cold Start and Hot Start .....	II-3-1
Power-on Reset .....	II-3-2
Reset Pulse .....	II-3-2
Boot Address .....	II-3-2
Notes Related to Initial Reset .....	II-3-3
<b>II-4 BCU (Bus Control Unit)</b>	<b>II-4-1</b>
Pin Assignment for External System Interface .....	II-4-1
I/O Pin List .....	II-4-1
Combination of System Bus Control Signals .....	II-4-2
Memory Area .....	II-4-3
Memory Map .....	II-4-3
External Memory Map and Chip Enable .....	II-4-4
Using Internal Memory on External Memory Area .....	II-4-6
Exclusive Signals for Areas .....	II-4-6
Area 10 .....	II-4-7
Setting External Bus Conditions .....	II-4-8
Setting Device Type and Size .....	II-4-8
Setting SRAM Timing Conditions .....	II-4-9
Setting Timing Conditions of Burst ROM .....	II-4-10
Bus Operation .....	II-4-11
Data Arrangement in Memory .....	II-4-11
Bus Operation of External Memory .....	II-4-11
Bus Clock .....	II-4-15
Bus Speed Mode .....	II-4-16
Bus Clock Output .....	II-4-16
Bus Cycles in External System Interface .....	II-4-17
SRAM Read Cycles .....	II-4-17
SRAM Write Cycles .....	II-4-21
Burst ROM Read Cycles .....	II-4-24
Releasing External Bus .....	II-4-25
Power-down Control by External Device .....	II-4-25
I/O Memory of BCU .....	II-4-26
Programming Notes .....	II-4-38



<b>II-5 ITC (Interrupt Controller)</b> .....	<b>II-5-1</b>
Outline of Interrupt Functions .....	II-5-1
Maskable Interrupts .....	II-5-1
Interrupt Factors and Intelligent DMA .....	II-5-3
Nonmaskable Interrupt (NMI) .....	II-5-3
Interrupt Processing by the CPU .....	II-5-3
Clearing Standby Mode by Interrupts .....	II-5-4
Trap Table .....	II-5-5
Control of Maskable Interrupts .....	II-5-6
Structure of the Interrupt Controller .....	II-5-6
Processor Status Register (PSR) .....	II-5-6
Interrupt Factor Flag and Interrupt Enable Register .....	II-5-7
Interrupt Priority Register and Interrupt Levels .....	II-5-9
IDMA Invocation .....	II-5-10
HSDMA Invocation .....	II-5-12
I/O Memory of Interrupt Controller .....	II-5-13
Programming Notes .....	II-5-25
<b>II-6 CLG (Clock Generator)</b> .....	<b>II-6-1</b>
Configuration of Clock Generator .....	II-6-1
I/O Pins of Clock Generator .....	II-6-2
High-Speed (OSC3) Oscillation Circuit .....	II-6-2
PLL .....	II-6-3
Controlling Oscillation .....	II-6-3
Setting and Switching Over the CPU Operating Clock .....	II-6-4
Power-Control Register Protection Flag .....	II-6-5
Operation in Standby Mode .....	II-6-5
I/O Memory of Clock Generator .....	II-6-6
Programming Notes .....	II-6-9
<b>II-7 POWER-DOWN CONTROL</b> .....	<b>II-7-1</b>
<b>II-8 DBG (Debug Unit)</b> .....	<b>II-8-1</b>
Debug Circuit .....	II-8-1
I/O Pins of Debug Circuit .....	II-8-1

## III PERIPHERAL BLOCK

---

<b>III-1 INTRODUCTION</b>	<b>III-1-1</b>
<b>III-2 CHIP ID / PIN STATUS CONTROL REGISTERS</b>	<b>III-2-1</b>
Chip ID register .....	III-2-1
Pull-up Control Resistors .....	III-2-2
Driving Bus Signals Low .....	III-2-2
Setting BCU to Access to the Registers .....	III-2-3
I/O Memory for Chip ID and Pin Status Control .....	III-2-4
<b>III-3 PRESCALER</b>	<b>III-3-1</b>
Configuration of Prescaler .....	III-3-1
Source Clock .....	III-3-1
Selecting Division Ratio and Output Control for Prescaler .....	III-3-2
Source Clock Output to 8-Bit Programmable Timer .....	III-3-2
I/O Memory of Prescaler .....	III-3-3
Programming Notes .....	III-3-7
<b>III-4 8-BIT PROGRAMMABLE TIMERS</b>	<b>III-4-1</b>
Configuration of 8-Bit Programmable Timer .....	III-4-1
Output Pins of 8-Bit Programmable Timers .....	III-4-2
Uses of 8-Bit Programmable Timers .....	III-4-3
Control and Operation of 8-Bit Programmable Timer .....	III-4-5
Control of Clock Output .....	III-4-8
8-Bit Programmable Timer Interrupts and DMA .....	III-4-9
I/O Memory of 8-Bit Programmable Timers .....	III-4-11
Programming Notes .....	III-4-19
<b>III-5 16-BIT PROGRAMMABLE TIMERS</b>	<b>III-5-1</b>
Configuration of 16-Bit Programmable Timer .....	III-5-1
I/O Pins of 16-Bit Programmable Timers .....	III-5-2
Uses of 16-Bit Programmable Timers .....	III-5-3
Control and Operation of 16-Bit Programmable Timer .....	III-5-4
Controlling Clock Output .....	III-5-7
16-Bit Programmable Timer Interrupts and DMA .....	III-5-9
I/O Memory of 16-Bit Programmable Timers .....	III-5-12
Programming Notes .....	III-5-26
<b>III-6 WATCHDOG TIMER</b>	<b>III-6-1</b>
Configuration of Watchdog Timer .....	III-6-1
Control of Watchdog Timer .....	III-6-1
Operation in Standby Modes .....	III-6-2
I/O Memory of Watchdog Timer .....	III-6-3
Programming Notes .....	III-6-3

<b>III-7 LOW-SPEED (OSC1) OSCILLATION CIRCUIT</b>	<b>III-7-1</b>
Configuration of Low-Speed (OSC1) Oscillation Circuit	III-7-1
I/O Pins of Low-Speed (OSC1) Oscillation Circuit	III-7-1
Oscillator Types	III-7-2
Controlling Oscillation	III-7-2
Switching Over the CPU Operating Clock	III-7-3
Power-Control Register Protection Flag	III-7-3
Operation in Standby Mode	III-7-3
OSC1 Clock Output to External Devices	III-7-4
I/O Memory of Low-Speed (OSC1) Oscillation Circuit	III-7-5
Programming Notes	III-7-8
<b>III-8 CLOCK TIMER</b>	<b>III-8-1</b>
Configuration of Clock Timer	III-8-1
Control and Operation of the Clock Timer	III-8-2
Interrupt Function	III-8-5
Examples of Use of Clock Timer	III-8-6
I/O Memory of Clock Timer	III-8-7
Programming Notes	III-8-12
<b>III-9 SERIAL INTERFACE</b>	<b>III-9-1</b>
Configuration of Serial Interfaces	III-9-1
Features of Serial Interfaces	III-9-1
I/O Pins of Serial Interface	III-9-2
Setting Transfer Mode	III-9-3
Clock-Synchronized Interface	III-9-4
Outline of Clock-Synchronized Interface	III-9-4
Setting Clock-Synchronized Interface	III-9-5
Control and Operation of Clock-Synchronized Transfer	III-9-7
Asynchronous Interface	III-9-12
Outline of Asynchronous Interface	III-9-12
Setting Asynchronous Interface	III-9-13
Control and Operation of Asynchronous Transfer	III-9-16
IrDA Interface	III-9-21
Outline of IrDA Interface	III-9-21
Setting IrDA Interface	III-9-21
Control and Operation of IrDA Interface	III-9-23
Serial Interface Interrupts and DMA	III-9-24
I/O Memory of Serial Interface	III-9-27
Programming Notes	III-9-42
<b>III-10 SERIAL INTERFACE with FIFO</b>	<b>III-10-1</b>
Configuration of Serial Interface with FIFO	III-10-1
Features of Serial Interface with FIFO	III-10-1
I/O Pins of Serial Interface with FIFO	III-10-2
Setting BCU for using the Serial Interface with FIFO	III-10-3
Setting Transfer Mode	III-10-3
Baud-Rate Timer (Setting Baud Rate)	III-10-4

## CONTENTS

Clock-Synchronized Interface .....	III-10-6
Outline of Clock-Synchronized Interface .....	III-10-6
Setting Clock-Synchronized Interface .....	III-10-7
Control and Operation of Clock-Synchronized Transfer .....	III-10-8
Asynchronous Interface .....	III-10-14
Outline of Asynchronous Interface .....	III-10-14
Setting Asynchronous Interface .....	III-10-15
Control and Operation of Asynchronous Transfer .....	III-10-18
IrDA Interface .....	III-10-22
Outline of IrDA Interface .....	III-10-22
Setting IrDA Interface .....	III-10-22
Control and Operation of IrDA Interface .....	III-10-24
FIFO Serial Interface Interrupts and DMA .....	III-10-25
I/O Memory of Serial Interface with FIFO .....	III-10-27
Programming Notes .....	III-10-37
<b>III-11 INPUT/OUTPUT PORTS</b> .....	<b>III-11-1</b>
Input Ports (K Ports) .....	III-11-1
Structure of Input Port .....	III-11-1
Input-Port Pins .....	III-11-2
Notes on Use .....	III-11-2
I/O Memory of Input Ports .....	III-11-3
I/O Ports (P Ports) .....	III-11-4
Structure of I/O Port .....	III-11-4
I/O Port Pins .....	III-11-4
I/O Control Register and I/O Modes .....	III-11-5
I/O Memory of I/O Ports .....	III-11-6
Input Interrupt .....	III-11-11
Port Input Interrupt .....	III-11-11
Key Input Interrupt .....	III-11-12
Control Registers of the Interrupt Controller .....	III-11-14
I/O Memory for Input Interrupts .....	III-11-16
Programming Notes .....	III-11-22
<b>III-12 EXTENDED PORTS</b> .....	<b>III-12-1</b>
Structure of Extended I/O Port .....	III-12-1
Extended I/O Port Pins .....	III-12-2
I/O Control Register and I/O Modes .....	III-12-3
Setting BCU to Access to the P4–P6 and EFP Registers .....	III-12-3
I/O Memory of Extended I/O Ports .....	III-12-4
<b>III-13 SMARTMEDIA INTERFACE</b> .....	<b>III-13-1</b>
Configuration of SmartMedia Interface .....	III-13-1
Output Pins of SmartMedia Interface .....	III-13-1
Selecting a #CE Area .....	III-13-2
Setting BCU to Access to the SmartMedia I/F Registers .....	III-13-2
I/O Memory of SmartMedia Interface .....	III-13-3

## IV ANALOG BLOCK

---

<b>IV-1 INTRODUCTION</b> .....	<b>IV-1-1</b>
<b>IV-2 A/D CONVERTER</b> .....	<b>IV-2-1</b>
Features and Structure of A/D Converter .....	IV-2-1
I/O Pins of A/D Converter .....	IV-2-2
Setting A/D Converter .....	IV-2-3
Control and Operation of A/D Conversion .....	IV-2-7
A/D Converter Interrupt and DMA .....	IV-2-10
I/O Memory of A/D Converter .....	IV-2-12
Programming Notes .....	IV-2-25

## V DMA BLOCK

---

<b>V-1 INTRODUCTION</b> .....	<b>V-1-1</b>
<b>V-2 HSDMA (High-Speed DMA)</b> .....	<b>V-2-1</b>
Functional Outline of HSDMA .....	V-2-1
I/O Pins of HSDMA .....	V-2-2
Programming Control Information .....	V-2-3
Setting the Registers in Dual-Address Mode .....	V-2-3
Setting the Registers in Single-Address Mode .....	V-2-6
Enabling/Disabling DMA Transfer .....	V-2-8
Trigger Factor .....	V-2-9
Operation of HSDMA .....	V-2-10
Operation in Dual-Address Mode .....	V-2-10
Operation in Single-Address Mode .....	V-2-13
Timing Chart .....	V-2-14
Interrupt Function of HSDMA .....	V-2-15
I/O Memory of HSDMA .....	V-2-16
Programming Notes .....	V-2-35
<b>V-3 IDMA (Intelligent DMA)</b> .....	<b>V-3-1</b>
Functional Outline of IDMA .....	V-3-1
Programming Control Information .....	V-3-1
IDMA Invocation .....	V-3-5
Operation of IDMA .....	V-3-8
Linking .....	V-3-12
Interrupt Function of Intelligent DMA .....	V-3-13
I/O Memory of Intelligent DMA .....	V-3-14
Programming Notes .....	V-3-17

## Appendix

---

<b>I/O Map</b> .....	<b>Appendix-1</b>
----------------------	-------------------



**S1C33301 Technical Manual**

**I S1C33301**





# I-1 Outline

The S1C33301 is a Seiko Epson original 32-bit microcomputer. It features high speed, low power consumption, and low-voltage operation, and is ideal for portable products that require high-speed data processing. The S1C33301 consists of a C33 STD 32-bit RISC type CPU as its core, peripheral circuits including a bus control unit, DMA controller, interrupt controller, timers, serial interface with FIFO, A/D converter, and SmartMedia interface, and also RAM. A high-speed oscillation circuit, PLL, and a low-speed oscillation circuit are also included, supporting advanced operation, power-saving operation, and high-performance realtime clock functions. Use of the internal MAC (multiplication and accumulation) function in combination with the A/D converter also facilitates the design of systems requiring DSP functions, such as speech recognition and synthesis applications.

Table I.1.1 Model Lineup

Model	Package	Internal RAM	Internal ROM	Data bus I/F	I/O port	A/D input
S1C33301F00A***	QFP15-128pin	8K bytes	None	LVC MOS/LVTTL	13 (I)/47 (I/O)	8 ch.
S1C33301B00A***	PFPGA-121pin	8K bytes	None	LVC MOS/LVTTL	9 (I)/44 (I/O)	4 ch.

Note: The end of the S1C33301 subcode is not related to model identification.

## Features

### Core CPU

Seiko Epson original 32-bit RISC CPU C33 STD built-in

- Basic instruction set: 105 instructions (16-bit fixed size)
- Sixteen 32-bit general-purpose register
- 32-bit ALU and 8-bit shifter
- Multiplication/division instructions and MAC (multiplication and accumulation) instruction are available
- 20 ns of minimum instruction execution time at 50 MHz operation

### Internal memory

RAM: 8K bytes

### Internal peripheral circuits

High-speed (OSC3) oscillation circuit: Crystal/ceramic oscillator 5 MHz min. to 33 MHz max.  
External clock input 2 MHz min. to 33 MHz max.

Low-speed (OSC1) oscillation circuit: Crystal oscillator or external clock input 32.768 kHz typ.

PLL: Doubles the high-speed (OSC3) oscillation clock frequency  
PLL input clock 15 MHz min. to 25 MHz max.  
PLL output clock 30 MHz min. to 50 MHz max.

Timers: 8-bit timer 6 channels  
16-bit timer 6 channels  
Watchdog timer (16-bit timer 0's function)  
Clock timer 1 channel (with alarm function)

Serial interface: 4 channels  
Clock-synchronous system, asynchronous system and IrDA 1.0 interface are selectable  
Ch.0 is selectable between a built-in buffer type (a 4 bytes of receive-data buffer and a 2 bytes of transmit-data buffer) and no buffer type

A/D converter: 10 bits × 8 channels (QFP15-128pin)  
10 bits × 4 channels (PFPGA-121pin)

DMA controller: High-speed DMA 4 channels  
Intelligent DMA 128 channels

## I-1 S1C33301: OUTLINE

Interrupt controller:	Possible to invoke DMA	
	Input interrupt	10 types (programmable)
	DMA controller interrupt	5 types
	16-bit programmable timer interrupt	12 types
	8-bit programmable timer interrupt	6 types
	Serial interface interrupt	15 types
	A/D converter interrupt	1 type
	Clock timer interrupt	1 type
General-purpose input and output ports:	Shared with the I/O pins for internal peripheral circuits	
	Input port	13 bits (QFP15-128pin) 9 bits (PFBGA-121pin)
	I/O port	47 bits (QFP15-128pin) 44 bits (PFBGA-121pin)
SmartMedia interface:	Outputs #SMRE and #SMWE signals that allow direct connection of a SmartMedia	

### External bus interface

BCU (bus control unit) built-in

- 26-bit address bus (internal 28-bit processing)
- 16-bit data bus  
Data size is selectable from 8 bits and 16 bits in each area.
- Little-endian memory access; big-endian may be set in each area.
- Memory mapped I/O
- Chip enable and wait control circuits built-in
- Supports burst ROM.

### Operating conditions and power consumption

Operating voltage:	Core (V <sub>DD</sub> )	1.65 V to 1.95 V (1.8 V ±0.15 V)
	I/O (V <sub>DDE</sub> )	2.7 V to 3.6 V (3.0/3.3 V ±0.3 V)
Operating clock frequency:	CPU clock	50 MHz max. <sup>Note 1</sup>
	Bus (BCU) clock	40 MHz max.
Operating temperature:		-40 to 85°C
Power consumption:	During SLEEP	8 μW typ.
	During HALT	18 mW typ. (50 MHz)
	During execution	40 mW typ. (50 MHz) <sup>Note 2</sup>

### Supply form

QFP15-128pin or PFBGA-121pin plastic package

- Notes:**
1. Set the #X2SPD pin to "0" when running the CPU with a 40 MHz or more system clock. Also make sure that the internal bus operating clock frequency does not exceed 40 MHz.
  2. The values of power consumption during execution were measured when a test program that consisted of 55% load instructions, 23% arithmetic operation instructions, 1% mac instruction, 12% branch instructions and 9% ext instruction was being continuously executed.

## I-2 Block Diagram

### C33 Macro Block Diagram

The S1C33301 consists of five major blocks: C33 Core Block, C33 Peripheral Block, C33 Analog Block, C33 DMA Block and C33 Internal Memory Block. Although the S1C33301 contains one more extended peripheral block, this manual explains the circuits as C33 peripheral block functions.

Figure I.2.1 shows the configuration of the C33 macro blocks.

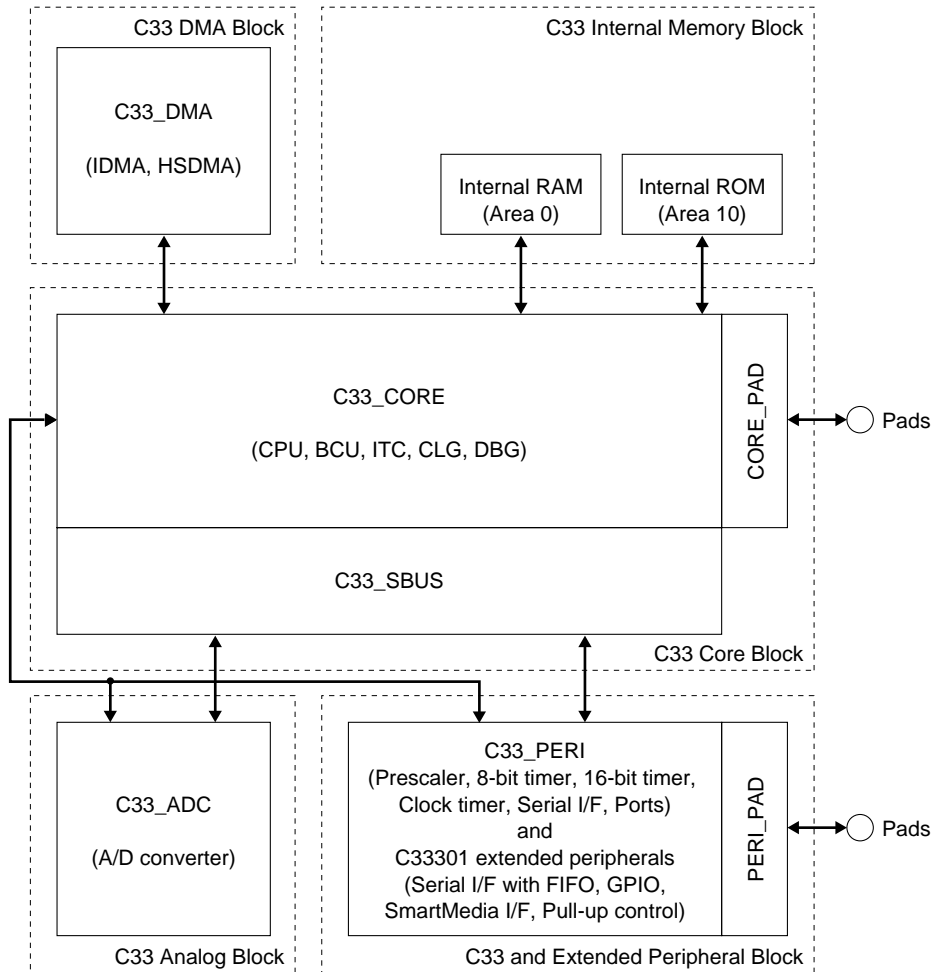


Figure I.2.1 C33 Macro Block Configuration

**Note:** Internal ROM is not provided in the S1C33301.

**C33 Core Block**

CPU	C33 STD 32-bit RISC type CPU
BCU (Bus Control Unit)	26-bit external address bus and 16-bit data bus All the BCU functions can be used.
ITC (Interrupt Controller)	50 types of interrupts are available.
CLG (Clock Generator)	OSC3 oscillation circuit (33 MHz Max.), PLL and OSC1 oscillation circuit (32.768 kHz Typ.) built-in
DBG (Debug Unit)	Functional block for debugging with the S5U1C33000H (In-Circuit Debugger for S1C33 Family)

Refer to Chapter II, "Core Block", for details.

**C33 Peripheral Block****C33 standard macro block**

Prescaler	Programmable clock generator for peripheral circuits
8-bit programmable timer	6 channels with clock output function
16-bit programmable timer	6 channels with event counter, clock output and watchdog timer functions
Serial interface	4 channels (asynchronous mode, clock synchronous mode and IrDA are selectable.)
Input and I/O ports	13 bits of input ports and 29 bits of I/O ports (used for peripheral I/O)
Clock timer	1 channel with alarm function

**S1C33301 extended peripheral circuits**

Serial interface with FIFO	1 channel (asynchronous mode, clock synchronous mode and IrDA are selectable.) Can be used as an alternative to serial interface Ch. 0 in the standard macro block.
SmartMedia interface	Generates the #SMWE and #SMRE signals using the BCU signals to interface directly with SmartMedia cards.
I/O ports	I/O ports are extended to 47 bits (including 29 bits of standard macro I/O ports).
Pull-up control circuit	Controls the pull-up resistors at the bus and I/O port pins.
Chip ID registers	Can be used to identify models and chip versions.

Refer to Chapter III, "Peripheral Block", for details.

**C33 Analog Block**

A/D converter	10-bit A/D converter with 8 input channels The S1C33301 allows use of extended functions.
---------------	--

Refer to Chapter IV, "Analog Block", for details of this block.

**C33 DMA Block**

HSDMA (High-Speed DMA)	4 channels
IDMA (Intelligent DMA)	128 channels

Refer to Chapter V, "DMA Block", for details.

**C33 Memory Block**

RAM	An 8KB SRAM built-in
-----	----------------------

Refer to Section I-5, "Internal Memory", for details.

# Functional Block Diagram

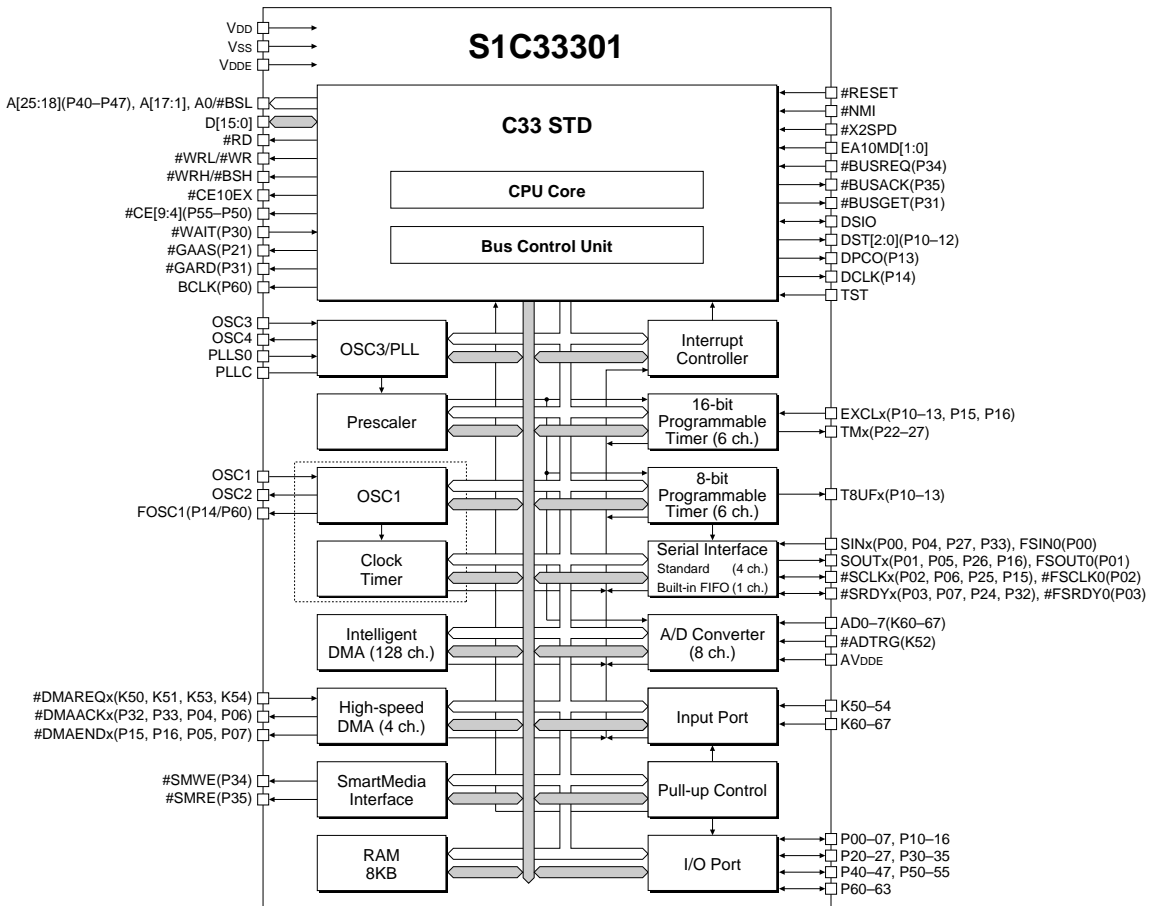


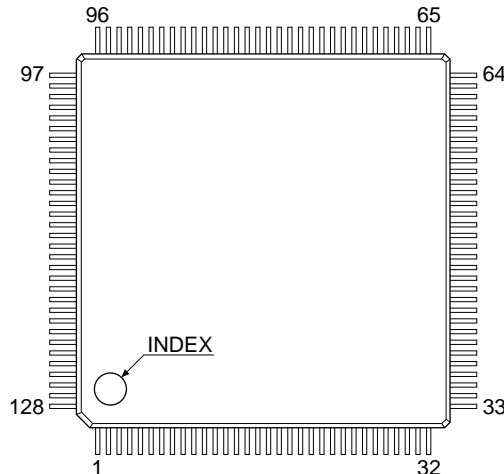
Figure I.2.2 S1C33301 Functional Block Diagram

THIS PAGE IS BLANK.

# I-3 Pin Description

## Pin Layout Diagram (plastic package)

QFP15-128pin

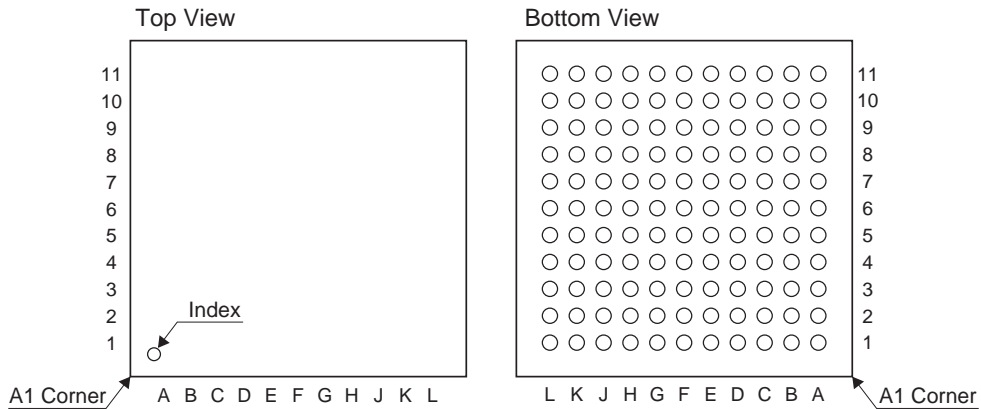


No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
1	<b>A0</b> (#BSL)	33	<b>#RESET</b>	65	<b>#CE6</b> (#CE7&8/P52)	97	<b>P24</b> (TM2/#SRDY2)
2	<b>A1</b>	34	<b>P30</b> (#WAIT/#CE4&5)	66	<b>#CE7</b> (#RAS0/#CE13/#RAS2/P53)	98	<b>P25</b> (TM3/#SCLK2)
3	<b>Vss</b>	35	<b>P31</b> (#BUGGET/#GARD)	67	<b>#CE8</b> (#RAS1/#CE14/#RAS3/P54)	99	<b>P26</b> (TM4/SOUT2)
4	<b>A2</b>	36	<b>P32</b> (#DMAACK0/#SRDY3)	68	<b>#CE9</b> (#CE17/#CE17&18/P55)	100	<b>P27</b> (TM5/SIN2)
5	<b>A3</b>	37	<b>P33</b> (#DMAACK1/SIN3)	69	<b>Vss</b>	101	<b>Vss</b>
6	<b>A4</b>	38	<b>Vss</b>	70	<b>D8</b>	102	<b>K50</b> (#DMAREQ0)
7	<b>A5</b>	39	<b>OSC4</b>	71	<b>D9</b>	103	<b>K51</b> (#DMAREQ1)
8	<b>A6</b>	40	<b>OSC3</b>	72	<b>D10</b>	104	<b>K52</b> (#ADTRG)
9	<b>A7</b>	41	<b>#X2SPD</b>	73	<b>D11</b>	105	<b>K53</b> (#DMAREQ2)
10	<b>VDD</b>	42	<b>P34</b> (#BUSREQ/#CE6/#SMWE)	74	<b>VDDE</b>	106	<b>K54</b> (#DMAREQ3)
11	<b>A8</b>	43	<b>P35</b> (#BUSACK/#SMRE)	75	<b>D12</b>	107	<b>DSIO</b>
12	<b>A9</b>	44	<b>D0</b>	76	<b>D13</b>	108	<b>DST0</b> (P10/EXCL0/T8UF0)
13	<b>A10</b>	45	<b>D1</b>	77	<b>D14</b>	109	<b>DST1</b> (P11/EXCL1/T8UF1)
14	<b>A11</b>	46	<b>VDDDE</b>	78	<b>D15</b>	110	<b>DST2</b> (P12/EXCL2/T8UF2)
15	<b>A12</b>	47	<b>D2</b>	79	<b>A22</b> (P43)	111	<b>VDDDE</b>
16	<b>A13</b>	48	<b>D3</b>	80	<b>A23</b> (P42)	112	<b>DPCO</b> (P13/EXCL3/T8UF3)
17	<b>A14</b>	49	<b>D4</b>	81	<b>#LCAS</b> (P41/A24)	113	<b>DCLK</b> (P14/FOSC1)
18	<b>VDDDE</b>	50	<b>D5</b>	82	<b>#HCAS</b> (P40/A25)	114	<b>P15</b> (EXCL4/#DMAEND0/#SCLK3)
19	<b>A15</b>	51	<b>D6</b>	83	<b>P00</b> (SIN0/FSIN0)	115	<b>P16</b> (EXCL5/#DMAEND1/SOUT3)
20	<b>A16</b>	52	<b>D7</b>	84	<b>P01</b> (SOUT0/FSOUT0)	116	<b>Vss</b>
21	<b>A17</b>	53	<b>VDD</b>	85	<b>P02</b> (#SCLK0/#FSCLK0)	117	<b>OSC2</b>
22	<b>A18</b> (P47)	54	<b>#WRL</b> (#WR/#WE)	86	<b>P03</b> (#SRDY0/#FSRDY0)	118	<b>OSC1</b>
23	<b>A19</b> (P46)	55	<b>#WRH</b> (#BSH)	87	<b>VDD</b>	119	<b>VDD</b>
24	<b>A20</b> (P45)	56	<b>#RD</b>	88	<b>P04</b> (SIN1/#DMAACK2)	120	<b>K60</b> (AD0)
25	<b>A21</b> (P44)	57	<b>#CE10EX</b> (#CE9&10EX)	89	<b>P05</b> (SOUT1/#DMAEND2)	121	<b>K61</b> (AD1)
26	<b>#NMI</b>	58	<b>BCLK</b> (P60/FOSC1)	90	<b>P06</b> (#SCLK1/#DMAACK3)	122	<b>K62</b> (AD2)
27	<b>EA10MD0</b>	59	<b>P61</b>	91	<b>P07</b> (#SRDY1/#DMAEND3)	123	<b>K63</b> (AD3)
28	<b>EA10MD1</b>	60	<b>P62</b>	92	<b>Vss</b>	124	<b>K64</b> (AD4)
29	<b>PLLS0</b>	61	<b>P63</b>	93	<b>P20</b> (#DRD)	125	<b>K65</b> (AD5)
30	<b>PLLCC</b>	62	<b>Vss</b>	94	<b>P21</b> (#DWE/#GAAS)	126	<b>K66</b> (AD6)
31	<b>Vss</b>	63	<b>#CE4</b> (#CE11/#CE11&12/P50)	95	<b>P22</b> (TM0)	127	<b>K67</b> (AD7)
32	<b>TST</b>	64	<b>#CE5</b> (#CE15/#CE15&16/P51)	96	<b>P23</b> (TM1)	128	<b>AVDDDE</b>

Boldface: Default setting

Figure I.3.1 Pin Layout Diagram (QFP15-128pin)

PFBGA-121pin



11	V <sub>SS</sub>	<b>P23</b> (TM1)	<b>P20</b> (#DRD)	V <sub>DD</sub>	<b>P01</b> (SOUT0/ FSOUT0)	<b>D15</b>	<b>D13</b>	<b>D10</b>	<b>D8</b>	<b>#CE7</b> (#RAS0/#CE13/ #RAS2/P53)	V <sub>SS</sub>	
10	<b>P25</b> (TM3/ #SCLK2)	<b>P24</b> (TM2/ #SRDY2)	<b>P21</b> (#DWE/ #GAAS)	<b>P05</b> (SOUT1/ #DMAEND2)	<b>P02</b> (#SCLK0/ #FSCLK0)	<b>A22</b> (P43)	V <sub>DD</sub> E	<b>D9</b>	<b>#CE8</b> (#RAS1/ #CE14/ #RAS3/P54)	<b>#CE6</b> (#CE7&8/P52)	<b>#CE5</b> #CE15/ #CE15&16/ P51)	
9	<b>K50</b> (#DMAREQ0)	<b>P26</b> (TM4/SOUT2)	<b>P22</b> (TM0)	<b>P06</b> (#SCLK1/ #DMAACK3)	<b>P03</b> (#SRDY0/ #FSRDY0)	<b>A23</b> (P42)	<b>D12</b>	<b>D11</b>	<b>#CE9</b> (#CE17/ #CE17&18/ P55)	<b>#CE4</b> (#CE11/ #CE11&12/ P50)	<b>BCLK</b> (P60/FOSC1)	
8	<b>K52</b> (#ADTRG)	<b>K51</b> (#DMAREQ1)	<b>P27</b> (TM5/SIN2)	<b>P07</b> (#SRDY1/ #DMAEND3)	<b>P04</b> (SIN1/ #DMAACK2)	<b>#LCAS</b> (P41/A24)	<b>D14</b>	<b>#RD</b>	<b>#CE10EX</b> (#CE9&10EX)	<b>#WRH</b> (#BSH)	<b>#WRL</b> (#WR/#WE)	
7	<b>DSIO</b>	<b>K54</b> (#DMAREQ3)	<b>K53</b> (#DMAREQ2)	<b>DST0</b> (P10/EXCL0/ T8UF0)	V <sub>SS</sub>	<b>#HCAS</b> (P40/A25)	<b>D5</b>	V <sub>SS</sub>	<b>D7</b>	<b>D6</b>	V <sub>DD</sub>	
6	V <sub>DD</sub> E	<b>DST1</b> (P11/EXCL1/ T8UF1)	<b>DCLK</b> (P14/FOSC1)	<b>DP00</b> (P13/EXCL3/ T8UF3)	<b>DST2</b> (P12/EXCL2/ T8UF2)	<b>P00</b> (SIN0/FSIN0)	<b>D0</b>	<b>D1</b>	<b>D4</b>	<b>D2</b>	<b>D3</b>	
5	<b>OSC2</b>	V <sub>SS</sub>	<b>P16</b> (EXCL5/ #DMAEND1/ SOUT3)	<b>P15</b> (EXCL4/ #DMAEND0/ #SCLK3)	<b>A10</b>	V <sub>DD</sub> E	<b>A19</b> (P46)	<b>P34</b> (#BUSREQ/ #CE6/ #SMWE)	<b>P35</b> (#BUSACK/ #SMRE)	<b>#X2SPD</b>	V <sub>DD</sub> E	
4	<b>OSC1</b>	<b>K60</b> (AD0)	V <sub>SS</sub>	V <sub>DD</sub>	<b>A8</b>	<b>A13</b>	<b>A18</b> (P47)	<b>P30</b> (#WAIT/ #CE4&5)	<b>P31</b> (#BUSGET/ #GARD)	<b>P33</b> (#DMAACK1/ SIN3)	<b>OSC3</b>	
3	<b>K61</b> (AD1)	<b>K63</b> (AD3)	<b>A2</b>	<b>A5</b>	<b>A6</b>	<b>A12</b>	<b>A17</b>	<b>#NMI</b>	<b>#RESET</b>	<b>P32</b> (#DMAACK0/ #SRDY3)	<b>OSC4</b>	
2	<b>AV<sub>DD</sub>E</b>	<b>K62</b> (AD2)	<b>A0</b> (#BSL)	<b>A3</b>	V <sub>DD</sub>	<b>A11</b>	<b>A15</b>	<b>A21</b> (P44)	<b>EA10MD0</b>	<b>PLLS0</b>	<b>TST</b>	
1	V <sub>SS</sub>	<b>A1</b>	<b>A4</b>	<b>A7</b>	<b>A9</b>	<b>A14</b>	<b>A16</b>	<b>A20</b> (P45)	<b>EA10MD1</b>	<b>PLLC</b>	V <sub>SS</sub>	
		A	B	C	D	E	F	G	H	J	K	L

**Boldface:** Default setting

Figure I.3.2 Pin Layout Diagram (PFBGA-121pin)



## Pin Functions

Table I.3.1 List of Pins for Power Supply System

Pin name	Pin No.		I/O	Pull-up	Function
	QFP	FPBGA			
V <sub>DD</sub>	10,53, 87,119	D4,D11, E2,L7	–	–	Power supply (+) for the internal logic
V <sub>SS</sub>	3,31,38, 62,69, 92,101, 116	A1,A11, B5,C4, E7,H7, L1,L11	–	–	Power supply (-); GND
V <sub>DDE</sub>	18,46, 74,111	A6,F5, G10,L5	–	–	Power supply (+) for the I/O block
AV <sub>DDE</sub>	128	A2	–	–	Analog system power supply (+)

Table I.3.2 List of Pins for External Bus Interface Signals

Pin name	Pin No.		I/O	Pull-up	Function
	QFP	FPBGA			
A0 #BSL	1	C2	I/O	*1	A0: Address bus (A0) when SBUSST(D3/0x4812E) = "0" (default) #BSL: Bus strobe (low byte) signal when SBUSST(D3/0x4812E) = "1"
A[17:1]	21–19, 17–11, 9–4,2	G3,G1, G2,F1, F4,F3, F2,E5, E1,E4, D1,E3, D3,C1, D2,C3, B1	O	*1	Address bus (A1–A17)
A18 P47	22	G4	I/O	*1	A18: Address bus (A18) when EFP47[1:0](D[7:6]/0x300049) = "00" (default) P47: I/O port when EFP47[1:0](D[7:6]/0x300049) = "01"
A19 P46	23	G5	I/O	*1	A19: Address bus (A19) when EFP46[1:0](D[5:4]/0x300049) = "00" (default) P46: I/O port when EFP46[1:0](D[5:4]/0x300049) = "01"
A20 P45	24	H1	I/O	*1	A20: Address bus (A20) when EFP45[1:0](D[3:2]/0x300049) = "00" (default) P45: I/O port when EFP45[1:0](D[3:2]/0x300049) = "01"
A21 P44	25	H2	I/O	*1	A21: Address bus (A21) when EFP44[1:0](D[1:0]/0x300049) = "00" (default) P44: I/O port when EFP44[1:0](D[1:0]/0x300049) = "01"
A22 P43	79	F10	I/O	*1	A22: Address bus (A22) when EFP43[1:0](D[7:6]/0x300048) = "00" (default) P43: I/O port when EFP43[1:0](D[7:6]/0x300048) = "01"
A23 P42	80	F9	I/O	*1	A23: Address bus (A23) when EFP42[1:0](D[5:4]/0x300048) = "00" (default) P42: I/O port when EFP42[1:0](D[5:4]/0x300048) = "01"
#LCAS P41 A24	81	F8	I/O (H)	*1	#LCAS: DRAM column address strobe (low byte) signal when EFP41[1:0](D[3:2]/0x300048) = "00" (default) P41: I/O port when EFP41[1:0](D[3:2]/0x300048) = "01" A24: Address bus (A24) when EFP41[1:0](D[3:2]/0x300048) = "10"
#HCAS P40 A25	82	F7	I/O (H)	*1	#HCAS: DRAM column address strobe (high byte) signal when EFP40[1:0](D[1:0]/0x300048) = "00" (default) P40: I/O port when EFP40[1:0](D[1:0]/0x300048) = "01" A25: Address bus (A25) when EFP40[1:0](D[1:0]/0x300048) = "10"
D[15:0]	78–75, 73–70, 52–47, 45,44	F11,G8, G11,G9, H9,H11, H10,J11, J7,K7, G7,J6, L6,K6, H6,G6	I/O	*2	Data bus (D0–D15)
#CE10EX #CE9&10EX	57	J8	O	*1	#CE10EX: Area 10 chip enable when CEFUNC[1:0](D[A:9]/0x48130) = "00" or "01" (default) #CE9&10EX: Areas 9&10 chip enable when CEFUNC[1:0](D[A:9]/0x48130) = "1x"
#CE9 #CE17 #CE17&18 P55	68	J9	I/O	*1	#CE9: Area 9 chip enable when EFP55[1:0](D[3:2]/0x30004B) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "00" (default) #CE17: Area 17 chip enable when EFP55[1:0](D[3:2]/0x30004B) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "01" #CE17&18: Areas 17&18 chip enable when EFP55[1:0](D[3:2]/0x30004B) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "1x" P55: I/O port when EFP55[1:0](D[3:2]/0x30004B) = "01"

I-3 S1C33301: PIN DESCRIPTION

Pin name	Pin No.		I/O	Pull-up	Function
	QFP	FPBGA			
#CE8 #RAS1 #CE14 #RAS3 P54	67	J10	I/O	*1	#CE8: Area 8 chip enable when EFP54[1:0](D[1:0]/0x30004B) = "00", CEFUNC[1:0](D[A:9]/0x48130) = "00" and A8DRA(D8/0x48128) = "0" (default) #RAS1: Area 8 DRAM row strobe when EFP54[1:0](D[1:0]/0x30004B) = "00", CEFUNC[1:0](D[A:9]/0x48130) = "00" and A8DRA(D8/0x48128) = "1" #CE14: Area 14 chip enable when EFP54[1:0](D[1:0]/0x30004B) = "00", CEFUNC[1:0](D[A:9]/0x48130) = "01" and A14DRA(D8/0x48122) = "0" #RAS3: Area 14 DRAM row strobe when EFP54[1:0](D[1:0]/0x30004B) = "00", CEFUNC[1:0](D[A:9]/0x48130) = "01" and A14DRA(D8/0x48122) = "1" P54: I/O port when EFP54[1:0](D[1:0]/0x30004B) = "01"
#CE7 #RAS0 #CE13 #RAS2 P53	66	K11	I/O	*1	#CE7: Area 7 chip enable when EFP53[1:0](D[7:6]/0x30004A) = "00", CEFUNC[1:0](D[A:9]/0x48130) = "00" and A7DRA(D7/0x48128) = "0" (default) #RAS0: Area 7 DRAM row strobe when EFP53[1:0](D[7:6]/0x30004A) = "00", CEFUNC[1:0](D[A:9]/0x48130) = "00" and A7DRA(D7/0x48128) = "1" #CE13: Area 13 chip enable when EFP53[1:0](D[7:6]/0x30004A) = "00", CEFUNC[1:0](D[A:9]/0x48130) = "01" and A13DRA(D7/0x48122) = "0" #RAS2: Area 13 DRAM row strobe when EFP53[1:0](D[7:6]/0x30004A) = "00", CEFUNC[1:0](D[A:9]/0x48130) = "01" and A13DRA(D7/0x48122) = "1" P53: I/O port when EFP53[1:0](D[7:6]/0x30004A) = "01"
#CE6 #CE7&8 P52	65	K10	I/O	*1	#CE6: Area 6 chip enable when EFP52[1:0](D[5:4]/0x30004A) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "00" or "01" (default) #CE7&8: Areas 7&8 chip enable when EFP52[1:0](D[5:4]/0x30004A) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "1x" P52: I/O port when EFP52[1:0](D[5:4]/0x30004A) = "01"
#CE5 #CE15 #CE15&16 P51	64	L10	I/O	*1	#CE5: Area 5 chip enable when EFP51[1:0](D[3:2]/0x30004A) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "00" (default) #CE15: Area 15 chip enable when EFP51[1:0](D[3:2]/0x30004A) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "01" #CE15&16: Areas 15&16 chip enable when EFP51[1:0](D[3:2]/0x30004A) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "1x" P51: I/O port when EFP51[1:0](D[3:2]/0x30004A) = "01"
#CE4 #CE11 #CE11&12 P50	63	K9	I/O	*1	#CE4: Area 4 chip enable when EFP50[1:0](D[1:0]/0x30004A) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "00" (default) #CE11: Area 11 chip enable when EFP50[1:0](D[1:0]/0x30004A) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "01" #CE11&12: Areas 11&12 chip enable when EFP50[1:0](D[1:0]/0x30004A) = "00" and CEFUNC[1:0](D[A:9]/0x48130) = "1x" P50: I/O port when EFP50[1:0](D[1:0]/0x30004A) = "01"
#RD	56	H8	O	*1	Read signal
#WRL #WR #WE	54	L8	O	*1	#WRL: Write (low byte) signal when SBUSST(D3/0x4812E) = "0" (default) #WR: Write signal when SBUSST(D3/0x4812E) = "1" #WE: DRAM write signal
#WRH #BSH	55	K8	O	*1	#WRH: Write (high byte) signal when SBUSST(D3/0x4812E) = "0" (default) #BSH: Bus strobe (high byte) signal when SBUSST(D3/0x4812E) = "1"
BCLK P60 FOSC1	58	L9	I/O	*1	BCLK: Bus clock output when EFP60[1:0](D[1:0]/0x30004C) = "00" (default) P60: I/O port when EFP60[1:0](D[1:0]/0x30004C) = "01" FOSC1: OSC1 clock output when EFP60[1:0](D[1:0]/0x30004C) = "10"

Table I.3.3 List of Port Pins

Pin name	Pin No.		I/O	Pull-up	Function
	QFP	FPBGA			
<b>K50</b> #DMAREQ0	102	A9	I	*1	K50: Input port when CFK50(D0/0x402C0) = "0" (default) #DMAREQ0: HSDMA Ch. 0 request input when CFK50(D0/0x402C0) = "1"
<b>K51</b> #DMAREQ1	103	B8	I	*1	K51: Input port when CFK51(D1/0x402C0) = "0" (default) #DMAREQ1: HSDMA Ch. 1 request input when CFK51(D1/0x402C0) = "1"
<b>K52</b> #ADTRG	104	A8	I	*1	K52: Input port when CFK52(D2/0x402C0) = "0" (default) #ADTRG: A/D converter trigger input when CFK52(D2/0x402C0) = "1"
<b>K53</b> #DMAREQ2	105	C7	I	*1	K53: Input port when CFK53(D3/0x402C0) = "0" (default) #DMAREQ2: HSDMA Ch. 2 request input when CFK53(D3/0x402C0) = "1"
<b>K54</b> #DMAREQ3	106	B7	I	*1	K54: Input port when CFK54(D4/0x402C0) = "0" (default) #DMAREQ3: HSDMA Ch. 3 request input when CFK54(D4/0x402C0) = "1"
<b>K60</b> AD0	120	B4	I	*1	K60: Input port when CFK60(D0/0x402C3) = "0" (default) AD0: A/D converter Ch. 0 input when CFK60(D0/0x402C3) = "1"
<b>K61</b> AD1	121	A3	I	*1	K61: Input port when CFK61(D1/0x402C3) = "0" (default) AD1: A/D converter Ch. 1 input when CFK61(D1/0x402C3) = "1"
<b>K62</b> AD2	122	B2	I	*1	K62: Input port when CFK62(D2/0x402C3) = "0" (default) AD2: A/D converter Ch. 2 input when CFK62(D2/0x402C3) = "1"
<b>K63</b> AD3	123	B3	I	*1	K63: Input port when CFK63(D3/0x402C3) = "0" (default) AD3: A/D converter Ch. 3 input when CFK63(D3/0x402C3) = "1"
<b>K64</b> AD4	124	-	I	*1	K64: Input port when CFK64(D4/0x402C3) = "0" (default) AD4: A/D converter Ch. 4 input when CFK64(D4/0x402C3) = "1"
<b>K65</b> AD5	125	-	I	*1	K65: Input port when CFK65(D5/0x402C3) = "0" (default) AD5: A/D converter Ch. 5 input when CFK65(D5/0x402C3) = "1"
<b>K66</b> AD6	126	-	I	*1	K66: Input port when CFK66(D6/0x402C3) = "0" (default) AD6: A/D converter Ch. 6 input when CFK66(D6/0x402C3) = "1"
<b>K67</b> AD7	127	-	I	*1	K67: Input port when CFK67(D7/0x402C3) = "0" (default) AD7: A/D converter Ch. 7 input when CFK67(D7/0x402C3) = "1"
<b>P00</b> SIN0 FSIN0	83	F6	I/O	*1	P00: I/O port when CFP00(D0/0x402D0) = "0" and EFP00[1:0](D[1:0]/0x300040) = "00" (default) SIN0: Serial I/F Ch. 0 data input when CFP00(D0/0x402D0) = "1" and EFP00[1:0](D[1:0]/0x300040) = "00" FSIN0: Serial I/F with FIFO Ch.0 data input when EFP00[1:0](D[1:0]/0x300040) = "01"
<b>P01</b> SOUT0 FSOUT0	84	E11	I/O	*1	P01: I/O port when CFP01(D1/0x402D0) = "0" and EFP01[1:0](D[3:2]/0x300040) = "00" (default) SOUT0: Serial I/F Ch. 0 data output when CFP01(D1/0x402D0) = "1" and EFP01[1:0](D[3:2]/0x300040) = "00" FSOUT0: Serial I/F with FIFO Ch.0 data output when EFP01[1:0](D[3:2]/0x300040) = "01"
<b>P02</b> #SCLK0 #FSCLK0	85	E10	I/O	*1	P02: I/O port when CFP02(D2/0x402D0) = "0" and EFP02[1:0](D[5:4]/0x300040) = "00" (default) #SCLK0: Serial I/F Ch. 0 clock input/output when CFP02(D2/0x402D0) = "1" and EFP02[1:0](D[5:4]/0x300040) = "00" #FSCLK0: Serial I/F with FIFO Ch.0 data input/output when EFP02[1:0](D[5:4]/0x300040) = "01"
<b>P03</b> #SRDY0 #FSRDY0	86	E9	I/O	*1	P03: I/O port when CFP03(D3/0x402D0) = "0" and EFP03[1:0](D[7:6]/0x300040) = "00" (default) #SRDY0: Serial I/F Ch. 0 ready signal input/output when CFP03(D3/0x402D0) = "1" and EFP03[1:0](D[7:6]/0x300040) = "00" #FSRDY0: Serial I/F with FIFO Ch.0 ready signal input/output when EFP03[1:0](D[7:6]/0x300040) = "01"
<b>P04</b> SIN1 #DMAACK2	88	E8	I/O	*1	P04: I/O port when CFP04(D4/0x402D0) = "0" and CFEX4(D4/0x402DF) = "0" (default) SIN1: Serial I/F Ch. 1 data input when CFP04(D4/0x402D0) = "1" and CFEX4(D4/0x402DF) = "0" #DMAACK2: HSDMA Ch. 2 acknowledge output when CFEX4(D4/0x402DF) = "1"
<b>P05</b> SOUT1 #DMAEND2	89	D10	I/O	*1	P05: I/O port when CFP05(D5/0x402D0) = "0" and CFEX5(D5/0x402DF) = "0" (default) SOUT1: Serial I/F Ch. 1 data output when CFP05(D5/0x402D0) = "1" and CFEX5(D5/0x402DF) = "0" #DMAEND2: HSDMA Ch. 2 end-of-transfer signal output when CFEX5(D5/0x402DF) = "1"

**I-3 S1C33301: PIN DESCRIPTION**

Pin name	Pin No.		I/O	Pull-up	Function
	QFP	FPBGA			
<b>P06</b> #SCLK1 #DMAACK3	90	D9	I/O	*1	P06: I/O port when CFP06(D6/0x402D0) = "0" and CFEX6(D6/0x402DF) = "0" (default) #SCLK1: Serial I/F Ch. 1 clock input/output when CFP06(D6/0x402D0) = "1" and CFEX6(D6/0x402DF) = "0" #DMAACK3: HSDMA Ch. 3 acknowledge output when CFEX6(D6/0x402DF) = "1"
<b>P07</b> #SRDY1 #DMAEND3	91	D8	I/O	*1	P07: I/O port when CFP07(D7/0x402D0) = "0" and CFEX7(D7/0x402DF) = "0" (default) #SRDY1: Serial I/F Ch. 1 ready signal input/output when CFP07(D7/0x402D0) = "1" and CFEX7(D7/0x402DF) = "0" #DMAEND3: HSDMA Ch. 3 end-of-transfer signal output when CFEX7(D7/0x402DF) = "1"
<b>P15</b> EXCL4 #DMAEND0 #SCLK3	114	D5	I/O	*1	P15: I/O port when CFP15(D5/0x402D4) = "0" (default) EXCL4: 16-bit timer 4 event counter input when CFP15(D5/0x402D4) = "1" and IOC15(D5/0x402D6) = "0" #DMAEND0: HSDMA Ch. 1 end-of-transfer signal output when CFP15(D5/0x402D4) = "1" and IOC15(D5/0x402D6) = "1" #SCLK3: Serial I/F Ch. 3 clock input/output when SSCLK3(D2/0x402D7) = "1" and CFP15(D5/0x402D4) = "0"
<b>P16</b> EXCL5 #DMAEND1 SOUT3	115	C5	I/O	*1	P16: I/O port when CFP16(D6/0x402D4) = "0" (default) EXCL5: 16-bit timer 5 event counter input when CFP16(D6/0x402D4) = "1" and IOC16(D6/0x402D6) = "0" #DMAEND1: HSDMA Ch. 1 end-of-transfer signal output when CFP16(D6/0x402D4) = "1" and IOC16(D6/0x402D6) = "1" SOUT3: Serial I/F Ch. 3 data output when SSOUT3(D1/0x402D7) = "1" and CFP16(D6/0x402D4) = "0"
<b>P20</b> #DRD	93	C11	I/O	*1	P20: I/O port when CFP20(D0/0x402D8) = "0" (default) #DRD: DRAM read signal output for successive RAS mode when CFP20(D0/0x402D8) = "1"
<b>P21</b> #DWE #GAAS	94	C10	I/O	*1	P21: I/O port when CFP21(D1/0x402D8) = "0" and CFEX2(D2/0x402DF) = "0" (default) #DWE: DRAM write signal output for successive RAS mode when CFP21(D1/0x402D8) = "1" and CFEX2(D2/0x402DF) = "0" #GAAS: Area address strobe output for GA when CFEX2(D2/0x402DF) = "1"
<b>P22</b> TM0	95	C9	I/O	*1	P22: I/O port when CFP22(D2/0x402D8) = "0" (default) TM0: 16-bit timer 0 output when CFP22(D2/0x402D8) = "1"
<b>P23</b> TM1	96	B11	I/O	*1	P23: I/O port when CFP23(D3/0x402D8) = "0" (default) TM1: 16-bit timer 1 output when CFP23(D3/0x402D8) = "1"
<b>P24</b> TM2 #SRDY2	97	B10	I/O	*1	P24: I/O port when CFP24(D4/0x402D8) = "0" (default) TM2: 16-bit timer 2 output when CFP24(D4/0x402D8) = "1" #SRDY2: Serial I/F Ch. 2 ready signal input/output when SSRDY2(D3/0x402DB) = "1" and CFP24(D4/0x402D8) = "0"
<b>P25</b> TM3 #SCLK2	98	A10	I/O	*1	P25: I/O port when CFP25(D5/0x402D8) = "0" (default) TM3: 16-bit timer 3 output when CFP25(D5/0x402D8) = "1" #SCLK2: Serial I/F Ch. 2 clock input/output when SSCLK2(D2/0x402DB) = "1" and CFP25(D5/0x402D8) = "0"
<b>P26</b> TM4 SOUT2	99	B9	I/O	*1	P26: I/O port when CFP26(D6/0x402D8) = "0" (default) TM4: 16-bit timer 4 output when CFP26(D6/0x402D8) = "1" SOUT2: Serial I/F Ch. 2 data output when SSOUT2(D1/0x402DB) = "1" and CFP26(D6/0x402D8) = "0"
<b>P27</b> TM5 SIN2	100	C8	I/O	*1	P27: I/O port when CFP27(D7/0x402D8) = "0" (default) TM5: 16-bit timer 5 output when CFP27(D7/0x402D8) = "1" SIN2: Serial I/F Ch. 2 data input when SSIN2(D0/0x402DB) = "1" and CFP27(D7/0x402D8) = "0"
<b>P30</b> #WAIT #CE4&5	34	H4	I/O	*1	P30: I/O port when CFP30(D0/0x402DC) = "0" (default) #WAIT: Wait cycle request input when CFP30(D0/0x402DC) = "1" #CE4&5: Areas 4&5 chip enable when CFP30(D0/0x402DC) = "1" and IOC30(D0/0x402DE) = "1"
<b>P31</b> #BUSGET #GARD	35	J4	I/O	*1	P31: I/O port when CFP31(D1/0x402DC) = "0" and CFEX3(D3/0x402DF) = "0" (default) #BUSGET: Bus status monitor signal output for bus release request when CFP31(D1/0x402DC) = "1" and CFEX3(D3/0x402DF) = "0" #GARD: Area read signal output for GA when CFEX3(D3/0x402DF) = "1"

Pin name	Pin No.		I/O	Pull-up	Function
	QFP	FPBGA			
<b>P32</b> #DMAACK0 #SRDY3	36	K3	I/O	*1	P32: I/O port when CFP32(D2/0x402DC) = "0" (default) #DMAACK0: HSDMA Ch. 0 acknowledge output when CFP32(D2/0x402DC) = "1" #SRDY3: Serial I/F Ch. 3 ready signal input/output when SSRDY3(D3/0x402D7) = "1" and CFP32(D2/0x402DC) = "0"
<b>P33</b> #DMAACK1 SIN3	37	K4	I/O	*1	P33: I/O port when CFP33(D3/0x402DC) = "0" (default) #DMAACK1: HSDMA Ch. 1 acknowledge output when CFP33(D3/0x402DC) = "1" SIN3: Serial I/F Ch. 3 data input when SSIN3(D0/0x402D7) = "1" and CFP33(D3/0x402DC) = "0"
<b>P34</b> #BUSREQ #CE6 #SMWE	42	H5	I/O	*1	P34: I/O port when CFP34(D4/0x402DC) = "0" and EFP34[1:0](D[1:0]/0x300047) = "00" (default) #BUSREQ: Bus release request input when CFP34(D4/0x402DC) = "1" and EFP34[1:0](D[1:0]/0x300047) = "00" #CE6: Area 6 chip enable when CFP34(D4/0x402DC) = "1", IOC34(D4/0x402DE) = "1" and EFP34[1:0](D[1:0]/0x300047) = "00" #SMWE: SmartMedia write signal when EFP34[1:0](D[1:0]/0x300047) = "01"
<b>P35</b> #BUSACK #SMRE	43	J5	I/O	*1	P35: I/O port when CFP35(D5/0x402DC) = "0" and EFP35[1:0](D[3:2]/0x300047) = "00" (default) #BUSACK: Bus acknowledge output when CFP35(D5/0x402DC) = "1" and EFP35[1:0](D[3:2]/0x300047) = "00" #SMRE: SmartMedia read signal when EFP35[1:0](D[3:2]/0x300047) = "01"
<b>P61</b>	59	–	I/O	*1	I/O port
<b>P62</b>	60	–	I/O	*1	I/O port
<b>P63</b>	61	–	I/O	*1	I/O port

Table I.3.4 List of Pins for Debug Interface

Pin name	Pin No.		I/O	Pull-up	Function
	QFP	FPBGA			
<b>DST0</b> P10 EXCL0 T8UF0	108	D7	I/O (H)	*1	DST0: DST0 signal output when CFEX1(D1/0x402DF) = "1" (default) P10: I/O port when CFP10(D0/0x402D4) = "0" and CFEX1(D1/0x402DF) = "0" EXCL0: 16-bit timer 0 event counter input when CFP10(D0/0x402D4) = "1", IOC10(D0/0x402D6) = "0" and CFEX1(D1/0x402DF) = "0" T8UF0: 8-bit timer 0 output when CFP10(D0/0x402D4) = "1", IOC10(D0/0x402D6) = "1" and CFEX1(D1/0x402DF) = "0"
<b>DST1</b> P11 EXCL1 T8UF1	109	B6	I/O (H)	*1	DST1: DST1 signal output when CFEX1(D1/0x402DF) = "1" (default) P11: I/O port when CFP11(D1/0x402D4) = "0" and CFEX1(D1/0x402DF) = "0" EXCL1: 16-bit timer 1 event counter input when CFP11(D1/0x402D4) = "1", IOC11(D1/0x402D6) = "0" and CFEX1(D1/0x402DF) = "0" T8UF1: 8-bit timer 1 output when CFP11(D1/0x402D4) = "1", IOC11(D1/0x402D6) = "1" and CFEX1(D1/0x402DF) = "0"
<b>DST2</b> P12 EXCL2 T8UF2	110	E6	I/O (L)	*1	DST2: DST2 signal output when CFEX0(D0/0x402DF) = "1" (default) P12: I/O port when CFP12(D2/0x402D4) = "0" and CFEX0(D0/0x402DF) = "0" EXCL2: 16-bit timer 2 event counter input when CFP12(D2/0x402D4) = "1", IOC12(D2/0x402D6) = "0" and CFEX0(D0/0x402DF) = "0" T8UF2: 8-bit timer 2 output when CFP12(D2/0x402D4) = "1", IOC12(D2/0x402D6) = "1" and CFEX0(D0/0x402DF) = "0"
<b>DPCO</b> P13 EXCL3 T8UF3	112	D6	I/O (H)	*1	DPCO: DPCO signal output when CFEX1(D1/0x402DF) = "1" (default) P13: I/O port when CFP13(D3/0x402D4) = "0" and CFEX1(D1/0x402DF) = "0" EXCL3: 16-bit timer 3 event counter input when CFP13(D3/0x402D4) = "1", IOC13(D3/0x402D6) = "0" and CFEX1(D1/0x402DF) = "0" T8UF3: 8-bit timer 3 output when CFP13(D3/0x402D4) = "1", IOC13(D3/0x402D6) = "1" and CFEX1(D1/0x402DF) = "0"
<b>DCLK</b> P14 FOSC1	113	C6	I/O (H)	*1	DCLK: DCLK signal output when CFEX0(D0/0x402DF) = "1" (default) P14: I/O port when CFP14(D4/0x402D4) = "0" and CFEX0(D0/0x402DF) = "0" FOSC1: OSC1 clock output when CFP14(D4/0x402D4) = "1" and CFEX0(D0/0x402DF) = "0"
<b>DSIO</b>	107	A7	I/O	Pull-up	Serial I/O pin for debugging This pin is used to communicate with the debugging tool S5U1C33000H/S5U1C33001H.

Table I.3.5 List of Other Pins

Pin name	Pin No.		I/O	Pull-up	Function											
	QFP	FPBGA														
<b>OSC1</b>	118	A4	I	–	Low-speed (OSC1) oscillation input (32 kHz crystal oscillator or external clock input)											
<b>OSC2</b>	117	A5	O	–	Low-speed (OSC1) oscillation output											
<b>OSC3</b>	40	L4	I	–	High-speed (OSC3) oscillation input (crystal/ceramic oscillator or external clock input)											
<b>OSC4</b>	39	L3	O	–	High-speed (OSC3) oscillation output											
<b>PLLS0</b>	29	K2	I	–	PLL set-up pins 0: PLL not used, 1: Frequency-doubling mode											
<b>PLLC</b>	30	K1	–	–	Capacitor connecting pin for PLL											
<b>EA10MD1</b>	28	J1	I	Pull-up	Area 10 boot mode selection <table border="1"> <thead> <tr> <th>EA10MD1</th> <th>EA10MD0</th> <th colspan="2">Mode</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>External ROM mode</td> <td rowspan="2">Connect both the pins to V<sub>DD</sub> during normal operation.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Internal ROM mode</td> </tr> </tbody> </table>	EA10MD1	EA10MD0	Mode		1	1	External ROM mode	Connect both the pins to V <sub>DD</sub> during normal operation.	1	0	Internal ROM mode
EA10MD1	EA10MD0	Mode														
1	1	External ROM mode	Connect both the pins to V <sub>DD</sub> during normal operation.													
1	0	Internal ROM mode														
<b>EA10MD0</b>	27	J2	I	Pull-up												
<b>#X2SPD</b>	41	K5	I	–	Clock doubling mode set-up pin 1: CPU clock = bus clock × 1, 0: CPU clock = bus clock × 2 Be sure to set A1X1MD (D3/0x4813A) to "1" when using the chip in x2 speed mode (#X2SPD = 0).											
<b>#NMI</b>	26	H3	I	Pull-up	NMI request input pin											
<b>#RESET</b>	33	J3	I	Pull-up	Initial reset input pin											
<b>TST</b>	32	L2	I	Pull-down	Test input pin Connect to V <sub>SS</sub> during normal operation.											

\*1: A pull-up resistor is built in and can be enabled/disabled using the I/O register. (default: pull-up is enabled)

\*2: A bus hold latch is attached.

**Notes:** • "#" in the pin names indicates that the signal is low active.

- Boldface letters in the pin name and I/O columns indicate the default pin (signal) names and default signal directions, respectively.
- (H) and (L) in the I/O column indicate the default output level and are described only for the pins of which the output level is fixed at high or low after an initial reset.
- The K67–K64 and P63–P61 pins do not exist in the PFBGA-121 pin package. When this package is used, do not disable the pull-up resistors for these pins (leave the pull-up control registers default settings).

## I-4 Power Supply

This section explains the operating voltage of the S1C33301.

### Power Supply Pins

The S1C33301 has the power supply pins shown in Table I.4.1.

Table I.4.1 Power Supply Pins

Pin name	Pin No.		Function
	QFP	FPBGA	
VDD	10,53,87,119	D4,D11,E2,L7	Power supply (+) for the internal logic
VSS	3,31,38,62,69, 92,101,116	A1,A11,B5,C4, E7,H7,L1,L11	Power supply (-); GND
VDDE	18,46,74,111	A6,F5,G10,L5	Power supply (+) for the I/O block
AVDDE	128	A2	Analog system power supply (+)

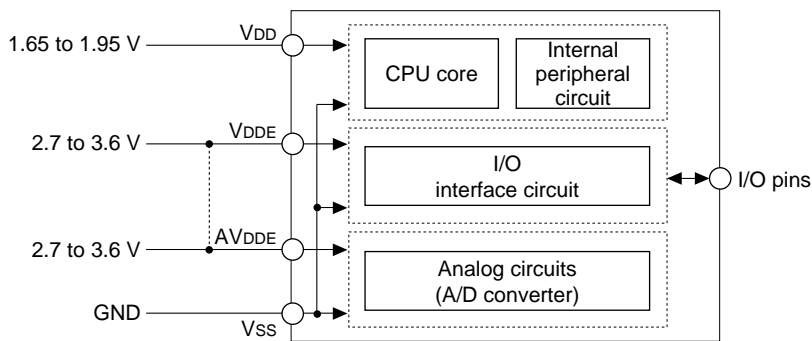


Figure I.4.1 Power Supply System

### Operating Voltage ( $V_{DD}$ , $V_{SS}$ )

The core CPU and internal peripheral circuits operate with a voltage supplied between the  $V_{DD}$  and  $V_{SS}$  pins. The following operating voltage can be used:

$$V_{DD} = 1.65 \text{ V to } 1.95 \text{ V} \quad (1.8 \text{ V} \pm 0.15 \text{ V}, V_{SS} = \text{GND})$$

**Note:** The S1C33301 has 4  $V_{DD}$  pins and 8  $V_{SS}$  pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.

### Power Supply for I/O Interface ( $V_{DDE}$ )

The  $V_{DDE}$  voltage is used for interfacing with external I/O signals. For the output interface of the S1C33301, the  $V_{DDE}$  voltage is used as high level and the  $V_{SS}$  voltage as low level.

The  $V_{SS}$  pin is used for the ground common with  $V_{DD}$ .

The following voltage is enabled for  $V_{DDE}$ :

$$V_{DDE} = 2.7 \text{ V to } 3.6 \text{ V} \quad (3.0/3.3 \text{ V} \pm 0.3 \text{ V}, V_{SS} = \text{GND})$$

**Notes:**

- The S1C33301 has 4  $V_{DDE}$  pins. Be sure to supply a voltage to all the pins. Do not open any of them.

- When an external clock is input to the OSC1 or OSC3 pin, the clock signal level must be  $V_{DD}$ .

## Power Supply for Analog Circuits (AVDDE)

The analog power supply pin (AVDDE) is provided separately from the VDD and VDDE pins in order that the digital circuits do not affect the analog circuit (A/D converter). The AVDDE pin is used to supply an analog power voltage and the VSS pin is used as the analog ground.

The following voltage is enabled for AVDDE:

$$AVDDE = 2.7 \text{ V to } 3.6 \text{ V (} 3.0/3.3 \text{ V} \pm 0.3 \text{ V, VSS} = \text{GND)}$$

**Note:** Be sure to supply VDDE to the AVDDE pin when the analog circuit is not used.

Noise on the analog power lines decrease the A/D converting precision, so use a stabilized power supply and make the board pattern with consideration given to that.

## Precautions on Power Supply

### Power-on sequence

In order to operate the device normally, supply power in accordance with the following timing.

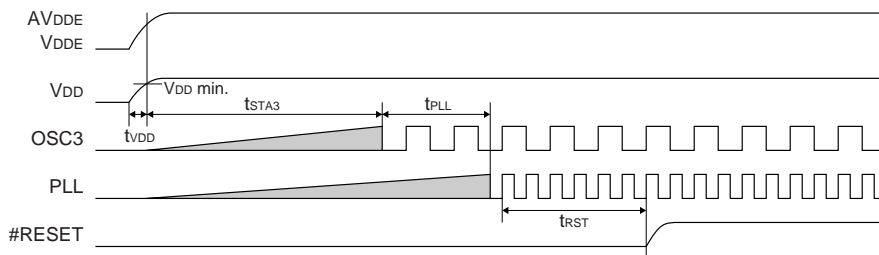


Figure I.4.2 Power-On Sequence

- (1)  $t_{VDD}$ : Elapsed time until the power supply stabilizes after power-on  
Supply power in the following sequence (or simultaneously).  
Power-on: VDD (Internal) → VDDE and AVDDE (I/O) → Apply the input signal
- (2)  $t_{STA3}$ : Time at which OSC3 oscillation starts
- (3)  $t_{PLL}$ : Time at which PLL locks up
- (4)  $t_{RST}$ : Minimum reset pulse width  
Time at which the clock supplied to the C33 STD core CPU stabilizes plus at least six clocks; Keep the #RESET signal low.

### Power-off sequence

Shut off the power supply in the following sequence (or simultaneously).

Power-off: Turn off the input signal → VDDE and AVDDE (I/O) → VDD (Internal)

### Latch-up

The CMOS device may be in the latch-up condition. This is the phenomenon caused by conduction of the parasitic PNP junction (thyristor) contained in the CMOS IC, resulting in a large current between VDD and VSS and leading to breakage.

Latch-up occurs when the voltage applied to the input / output exceeds the rated value and a large current flows into the internal element, or when the voltage at the VDD pin exceeds the rated value and the internal element is in the breakdown condition. In the latter case, even if the application of a voltage exceeding the rated value is instantaneous, the current remains high between VDD and VSS once the device is in the latch-up condition. As this may result in heat generation or smoking, the following points must be taken into consideration:

- (1) The voltage level at the input / output must not exceed the range specified in the electrical characteristics. In other words, it must be below the power-supply voltage and above VSS. The power-on timing should also be taken into consideration.
- (2) Abnormal noise must not be applied to the device.
- (3) The potential at the unused input should be fixed at VDD, VDDE, AVDDE, or VSS.
- (4) No outputs should be shorted.



## I-5 Internal Memory

This section explains the internal memory configuration.

Figure I.5.1 shows the S1C33301 memory map.

Area	Address	
Areas 18–11	0xFFFFFFF	External Memory
	0x1000000	
Area 10	0x0FFFFFF	External Memory
	0x0C00000	
Areas 9–7	0x0BFFFFFF	External Memory
	0x0400000	
Area 6	0x03FFFFFF	Internal I/O (16-bit device)
	0x0380000	
	0x037FFFF	
	0x0300000	Internal I/O (8-bit device) S1C33301 extended peripheral circuits
Areas 5–4	0x02FFFFFF	
	0x0100000	External Memory
Area 3	0x00FFFFFF	
	0x0080000	(Reserved) For middleware use
Area 2	0x007FFFF	
	0x0060000	(Reserved) For CPU, debug mode
Area 1	0x005FFFF	
	0x0050000	(Mirror of internal peripheral circuits)
	0x004FFFF	
	0x0040000	Internal peripheral circuits
	0x003FFFF	
	0x0030000	(Mirror of internal peripheral circuits)
Area 0	0x002FFFF	
	0x0002000	(Mirror of internal RAM)
	0x0001FFF	
	0x0000000	Internal RAM (8KB)

Figure I.5.1 Memory Map

Area 2 is used in debug mode only and it cannot be accessed in user mode (normal program execution status).

Area 6 is configured for external memory access at initial reset. The BCU register for Area 6 must be set up in accordance with the procedure described below before the S1C33301 extended peripheral circuits mapped in Area 6 can be accessed.

1. A6IO (D9) / Access control register (0x48132) = "1"  
This sets Area 6 so that the internal device can be accessed.
2. A6EC (D1) / Access control register (0x48132) = "0"  
This sets Area 6 so that it can be accessed in the little endian format.
3. A6WT[2:0] (D[A:8]) / Areas 6–4 setup register (0x4812A) = "001"  
This sets Area 6 so that it can be accessed with 1 wait state inserted.

## ***ROM and Boot Address***

---

The S1C33301 does not have a built-in ROM. The boot address is fixed at 0x0C00000, and so external ROM/Flash should be used in Area 10.

Refer to Section II-4, "BCU (Bus Control Unit)", for setting up Area 10.

## ***RAM***

---

The S1C33301 has a built-in 8KB RAM. The RAM is allocated to Area 0, address 0x0000000 to address 0x0001FFF.

The internal RAM is a 32-bit sized device and data can be read/written in 1 cycle regardless of data size (byte, half-word or word).

## ***External Address Bus***

---

The C33 STD Core CPU can process up to 28 bits of address for the internal bus. The S1C33301 external bus allows output of the low-order 26 bits (max.).

# I-6 Electrical Characteristics

## Absolute Maximum Rating

(V<sub>SS</sub>=0V)

Item	Symbol	Condition	Rated value	Unit	*
Internal logic power voltage	V <sub>DD</sub>		-0.3 to +2.5	V	
I/O power voltage	V <sub>DDE</sub>		-0.3 to +4.0	V	
Analog power voltage	AV <sub>DDE</sub>		-0.3 to +4.0	V	
Input voltage	V <sub>I</sub>		-0.3 to V <sub>DDE</sub> +0.5	V	
Analog input voltage	AV <sub>IN</sub>		-0.3 to AV <sub>DDE</sub> +0.3	V	
High-level output current	I <sub>OH</sub>	1 pin	-10	mA	
		Total of all pins	-40	mA	
Low-level output current	I <sub>OL</sub>	1 pin	10	mA	
		Total of all pins	40	mA	
Storage temperature	T <sub>STG</sub>		-65 to +150	°C	

E char

## Recommended Operating Conditions

(V<sub>SS</sub>=0V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Internal logic power voltage	V <sub>DD</sub>		1.65	1.80	1.95	V	
I/O power voltage	V <sub>DDE</sub>		2.70	–	3.60	V	
Analog power voltage	AV <sub>DDE</sub>		2.70	–	3.60	V	
Input voltage	HV <sub>I</sub>		V <sub>SS</sub>	–	V <sub>DDE</sub>	V	
	LV <sub>I</sub>		V <sub>SS</sub>	–	V <sub>DD</sub>	V	
Analog input voltage	AV <sub>IN</sub>		V <sub>SS</sub>	–	AV <sub>DDE</sub>	V	
CPU operating clock frequency	f <sub>CPU</sub>		–	–	50	MHz	
Bus operating clock frequency	f <sub>BUS</sub>		–	–	40	MHz	
OSC3 oscillation frequency	f <sub>OSC3</sub>		5	–	33	MHz	
OSC3 external input clock frequency	f <sub>ECLK</sub>		2	–	33	MHz	
OSC1 oscillation frequency	f <sub>OSC1</sub>		–	32.768	–	kHz	
Operating temperature	T <sub>a</sub>		-40	25	85	°C	
Input rise time (normal input)	t <sub>ri</sub>		–	–	50	ns	
Input fall time (normal input)	t <sub>fi</sub>		–	–	50	ns	
Input rise time (Schmitt input)	t <sub>ri</sub>		–	–	5	ms	
Input fall time (Schmitt input)	t <sub>fi</sub>		–	–	5	ms	

## DC Characteristics

(Unless otherwise specified:  $V_{DDE}=2.7V$  to  $3.6V$ ,  $V_{DD}=1.65V$  to  $1.95V$ ,  $T_a=-40^{\circ}C$  to  $+85^{\circ}C$ )

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*	
Input leakage current	ILI		-5	—	5	$\mu A$		
Off-state leakage current	IOZ		-5	—	5	$\mu A$		
High-level output voltage	VOH	$I_{OH}=-1.7mA$ (2mA Type), $I_{OH}=-3.5mA$ (4mA Type), $V_{DD}=\text{Min.}$	$V_{DD}$ -0.4	—	—	V		
Low-level output voltage	VOL	$I_{OL}=1.7mA$ (2mA Type), $I_{OL}=3.5mA$ (4mA Type), $V_{DD}=\text{Min.}$	—	—	0.4	V		
High-level input voltage	VIH	LVTTL level, $V_{DDE}=\text{Max.}$	2.0	—	—	V		
Low-level input voltage	VIL	LVTTL level, $V_{DDE}=\text{Min.}$	—	—	0.7	V		
Positive trigger input voltage	VT+	LVC MOS Schmitt	1.2	—	2.7	V		
Negative trigger input voltage	VT-	LVC MOS Schmitt	0.5	—	1.8	V		
Hysteresis voltage	VH	LVC MOS Schmitt	0.2	—	—	V		
Pull-up resistor	RPU	$V_I=0V$	100k $\Omega$ Type	50	100	288	k $\Omega$	
			50k $\Omega$ Type	25	50	144	k $\Omega$	
Pull-down resistor	RPD	$V_I=V_{DDE}$ (TST)	25	50	144	k $\Omega$		
High-level latching current	IBHH	Pins with bus-hold latch, $V_I=1.9V$ , $V_{DDE}=\text{Min.}$	—	—	-20	$\mu A$		
Low-level latching current	IBHL	Pins with bus-hold latch, $V_I=0.8V$ , $V_{DDE}=\text{Min.}$	—	—	17	$\mu A$		
High-level reversal current	IBHHO	Pins with bus-hold latch, $V_I=0.8V$ , $V_{DDE}=\text{Max.}$	-350	—	—	$\mu A$		
Low-level reversal current	IBHLO	Pins with bus-hold latch, $V_I=1.9V$ , $V_{DDE}=\text{Max.}$	300	—	—	$\mu A$		
Input pin capacitance	CI	$f=1MHz$ , $V_{DDE}=0V$	—	—	8	pF		
Output pin capacitance	CO	$f=1MHz$ , $V_{DDE}=0V$	—	—	8	pF		
I/O pin capacitance	CIO	$f=1MHz$ , $V_{DDE}=0V$	—	—	8	pF		

**Note:** See "Pin Characteristics" in this section for pin characteristics.

## Current Consumption

### Operating current

(Unless otherwise specified:  $V_{DDE}=AV_{DDE}=2.7V$  to  $3.6V$ ,  $V_{DD}=1.65V$  to  $1.95V$ ,  $V_{SS}=0V$ ,  $T_a=-40^{\circ}C$  to  $+85^{\circ}C$ )

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Current consumption during CPU running	IDD1	20MHz	–	9	11	mA	1
		25MHz	–	11	14		
		33MHz	–	15	18		
		50MHz	–	22	27		
Current consumption in HALT mode	IDD2	20MHz	–	4	5	mA	2
		25MHz	–	5	7		
		33MHz	–	7	9		
		50MHz	–	10	13		
Current consumption in HALT2 mode	IDD3	20MHz	–	1.5	2.5	mA	3
		25MHz	–	2.0	3.0		
		33MHz	–	2.5	4.0		
		50MHz	–	4.0	5.5		

Current consumption measurement condition:  $V_{IH}=V_{DDE}$ ,  $V_{IL}=0V$ , output pins are open,  $V_{DDE}$  and  $AV_{DDE}$  currents are not included

Typ. value measurement condition:  $V_{DDE}=AV_{DDE}=3.3V$ ,  $V_{DD}=1.8V$ ,  $T_a=25^{\circ}C$  Typ.

\* note)

No.	OSC3	OSC1	CPU	Other peripheral circuits
1	On	Off	Normal operation *1	Stop
2	On	Off	HALT mode	Stop
3	On	Off	HALT2 mode	Stop

\*1: The values of current consumption while the CPU is operating were measured when a test program that consists of 55% load instructions, 23% arithmetic operation instructions, 1% mac instruction, 12% branch instructions and 9% ext instruction is being executed in the built-in ROM continuously.

### Current consumption in SLEEP mode

(Unless otherwise specified:  $V_{DDE}=AV_{DDE}=2.7V$  to  $3.6V$ ,  $V_{DD}=1.65V$  to  $1.95V$ ,  $V_{SS}=0V$ )

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
VDD current consumption in SLEEP mode	IDDSL	$T_a=25^{\circ}C$	–	1	10	$\mu A$	
		$T_a=85^{\circ}C$	–	10	100		
V <sub>DDE</sub> + AV <sub>DDE</sub> current consumption in SLEEP mode	IDDSE	$T_a=25^{\circ}C$	–	2	20	$\mu A$	
		$T_a=85^{\circ}C$	–	4	30		

Current consumption measurement condition:  $V_{IH}=V_{DDE}$ ,  $V_{IL}=0V$ , output pins are open

Typ. value measurement condition:  $V_{DDE}=AV_{DDE}=3.3V$ ,  $V_{DD}=1.8V$ , Typ.

### Peripheral circuit operating currents

(Unless otherwise specified:  $V_{DDE}=AV_{DDE}=3.3V$ ,  $V_{DD}=1.8V$ ,  $V_{SS}=0V$ ,  $T_a=25^{\circ}C$ )

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Clock timer operating current	ICT	OSC1 oscillation: 32kHz	–	1	–	$\mu A$	4
A/D converter operating current	IAD	When A/D converter is enabled	–	300	–	$\mu A$	5
OSC3 operating current	IOSC3	OSC3 oscillation: 33MHz	–	0.6	–	mA	4
PLL operating current	IPLL	PLL output clock: 50MHz	–	1	–	mA	4

\* note 4) V<sub>DD</sub> power current consumption

5) AV<sub>DDE</sub> power current consumption

## A/D Converter Characteristics

(Unless otherwise specified: V<sub>DDE</sub>=AV<sub>DDE</sub>=2.7V to 3.6V, V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, T<sub>a</sub>=-40°C to +85°C, ST[1:0]=11)

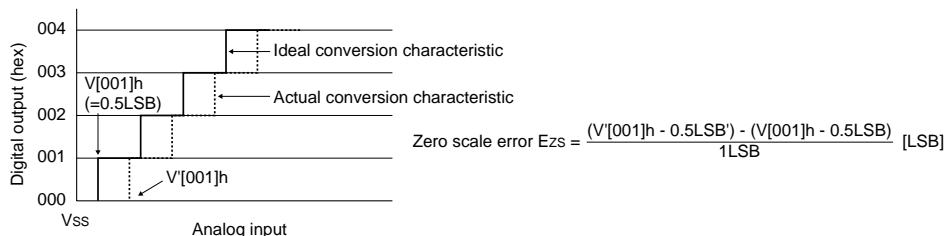
Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Resolution	—		—	10	—	bit	
Conversion time	—		10	—	1250	μs	1
Zero scale error	E <sub>Zs</sub>		-2	—	2	LSB	
Full scale error	E <sub>FS</sub>		-2	—	2	LSB	
Integral linearity error	E <sub>L</sub>		-3	—	3	LSB	
Differential linearity error	E <sub>D</sub>		-3	—	3	LSB	
Permissible signal source impedance	—		—	—	5	kΩ	
Analog input capacitance	—		—	—	45	pF	

\* note 1) Indicates the minimum value when A/D clock = 2MHz.  
Indicates the maximum value when A/D clock = 16kHz.

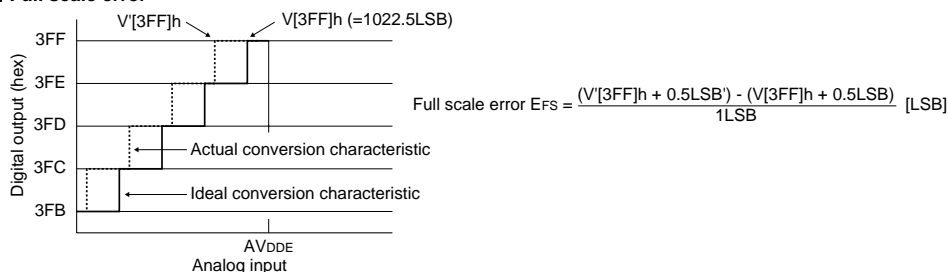
### A/D conversion error

V[001]h = Ideal voltage at zero-scale point (=0.5LSB)  
 V'[001]h = Actual voltage at zero-scale point  $1LSB = \frac{AV_{DDE} - V_{SS}}{2^{10} - 1}$   
 V[3FF]h = Ideal voltage at full-scale point (=1022.5LSB)  
 V'[3FF]h = Actual voltage at full-scale point  $1LSB' = \frac{V'[3FF]h - V'[001]h}{2^{10} - 2}$

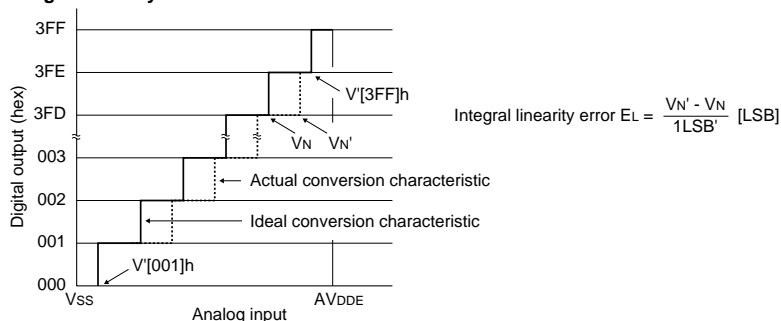
#### Zero scale error



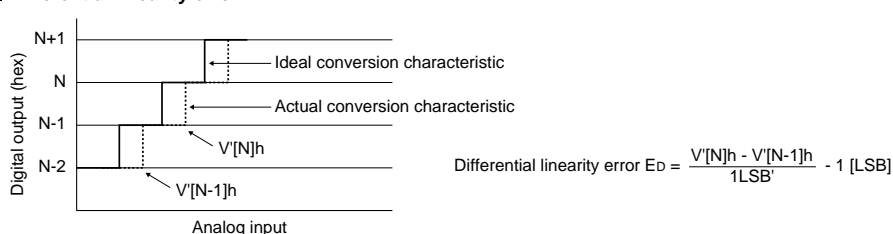
#### Full scale error



#### Integral linearity error



#### Differential linearity error



## Oscillation Characteristics

Oscillation characteristics change depending on conditions such as components used (oscillator, R<sub>f</sub>, R<sub>d</sub>, C<sub>G</sub>, C<sub>D</sub>) and board pattern. Use the following characteristics as reference values. In particular, when a ceramic or crystal oscillator is used, evaluate the components adequately under real operating conditions by mounting them on the board before the external register (R<sub>f</sub>, R<sub>d</sub>) and capacitor (C<sub>G</sub>, C<sub>D</sub>) values are finally decided.

### OSC1 crystal oscillation

(Unless otherwise specified: V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, T<sub>a</sub>=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t <sub>STA1</sub>				3	s	

### OSC3 crystal/ceramic oscillation

**Note:** A "crystal resonator that uses a fundamental" should be used for the OSC3 crystal oscillation circuit.

(Unless otherwise specified: V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, T<sub>a</sub>=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t <sub>STA3</sub>				25	ms	

## PLL Characteristics

### Setting the PLLS0 pin (recommended operating condition)

PLLS0	Mode	F <sub>in</sub> (OSC3 clock)	F <sub>out</sub>
1	Frequency doubling mode	15 to 25MHz	30 to 50MHz
0	PLL not used	—	—

### PLL characteristics

(Unless otherwise specified: V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, R<sub>1</sub>=4.7kΩ, C<sub>1</sub>=100pF, C<sub>2</sub>=3pF, T<sub>a</sub>=-40°C to +85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Lockup time	t <sub>PLL</sub>				1	ms	

## Pin Characteristics

Pin No.		Signal name	I/O cell name	Input/output characteristic	Output characteristic	I/O	Pull-up/down (resistor value)	Power source	Remarks
QFP	BGA								
1	C2	A0(#BSL)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
2	B1	A1	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
3	A1	V <sub>ss</sub>	POWER	–	–	–	–	–	
4	C3	A2	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
5	D2	A3	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
6	C1	A4	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
7	D3	A5	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
8	E3	A6	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
9	D1	A7	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
10	E2	V <sub>DD</sub>	POWER	–	–	–	–	–	
11	E4	A8	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
12	E1	A9	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
13	E5	A10	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
14	F2	A11	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
15	F3	A12	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
16	F4	A13	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
17	F1	A14	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
18	F5	V <sub>DDE</sub>	POWER	–	–	–	–	–	
19	G2	A15	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
20	G1	A16	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
21	G3	A17	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 3
22	G4	A18(P47)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
23	G5	A19(P46)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
24	H1	A20(P45)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
25	H2	A21(P44)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
26	H3	#NMI	HIBHP1TY	LVC MOS Schmitt	–	I	Pull-up(50k)	V <sub>DDE</sub>	
27	J2	EA10MD0	HIBHP2TY	LVC MOS Schmitt	–	I	Pull-up(100k)	V <sub>DDE</sub>	
28	J1	EA10MD1	HIBHP2TY	LVC MOS Schmitt	–	I	Pull-up(100k)	V <sub>DDE</sub>	
29	K2	PLLS0	HIBHY	LVC MOS Schmitt	–	I	–	V <sub>DDE</sub>	
30	K1	PLL C	LLINY	–	–	–	–	V <sub>DDE</sub>	
31	L1	V <sub>ss</sub>	POWER	–	–	–	–	–	
32	L2	TST	HIBHD1TY	–	–	I	Pull-down(50k)	V <sub>DDE</sub>	Test pin
33	J3	#RESET	HIBHP1TY	LVC MOS Schmitt	–	I	Pull-up(50k)	V <sub>DDE</sub>	
34	H4	P30(#WAIT/#CE4&5)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
35	J4	P31(#BUSGET/#GARD)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
36	K3	P32(#DMAACK0/#SRDY3)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
37	K4	P33(#DMAACK1/SIN3)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
38	H7	V <sub>ss</sub>	POWER	–	–	–	–	–	
39	L3	OSC4	LLOTY	–	–	O	–	V <sub>DD</sub>	
40	L4	OSC3	LLINY	–	–	I	–	V <sub>DD</sub>	note 4
41	K5	#X2SPD	HIBHTY	LVC MOS Schmitt	–	I	–	V <sub>DDE</sub>	
42	H5	P34(#BUSREQ/#CE6/#SMWE)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
43	J5	P35(#BUSACK/#SMRE)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
44	G6	D0	HBBT2BH TY	LV TTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
45	H6	D1	HBBT2BH TY	LV TTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
46	L5	V <sub>DDE</sub>	POWER	–	–	–	–	–	
47	K6	D2	HBBT2BH TY	LV TTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
48	L6	D3	HBBT2BH TY	LV TTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
49	J6	D4	HBBT2BH TY	LV TTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
50	G7	D5	HBBT2BH TY	LV TTL	4mA	I/O	–	V <sub>DDE</sub>	note 2



Pin No.		Signal name	I/O cell name	Input/output characteristic	Output characteristic	I/O	Pull-up/down (resistor value)	Power source	Remarks
QFP	BGA								
51	K7	D6	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
52	J7	D7	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
53	L7	V <sub>DD</sub>	POWER	–	–	–	–	–	
54	L8	#WRL(#WR/#WE)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
55	K8	#WRH(#BSH)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
56	H8	#RD	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
57	J8	#CE10EX(#CE9&10EX)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
58	L9	BCLK(P60/FOSC1)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
59	–	P61	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 6
60	–	P62	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 6
61	–	P63	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1, 6
62	L11	V <sub>SS</sub>	POWER	–	–	–	–	–	
63	K9	#CE4(#CE11/#CE11&12/P50)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
64	L10	#CE5(#CE15/#CE15&16/P51)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
65	K10	#CE6(#CE7&8/P52)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
66	K11	#CE7(#RAS0/#CE13/#RAS2/P53)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
67	J10	#CE8(#RAS1/#CE14/#RAS3/P54)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
68	J9	#CE9(#CE17/#CE17&18/P55)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
69	E7	V <sub>SS</sub>	POWER	–	–	–	–	–	
70	J11	D8	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
71	H10	D9	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
72	H11	D10	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
73	H9	D11	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
74	G10	V <sub>DDE</sub>	POWER	–	–	–	–	–	
75	G9	D12	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
76	G11	D13	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
77	G8	D14	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
78	F11	D15	HBBT2BH7Y	LVTTTL	4mA	I/O	–	V <sub>DDE</sub>	note 2
79	F10	A22(P43)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
80	F9	A23(P42)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
81	F8	#LCAS(P41/A24)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
82	F7	#HCAS(P40/A25)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
83	F6	P00(SIN0/FSIN0)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
84	E11	P01(SOUT0/FSOUT0)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
85	E10	P02(#SCLK0/#FSCLK0)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
86	E9	P03(#SRDY0/#FSRDY0)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
87	D11	V <sub>DD</sub>	POWER	–	–	–	–	–	
88	E8	P04(SIN1/#DMAACK2)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
89	D10	P05(SOUT1/#DMAEND2)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
90	D9	P06(#SCLK1/#DMAACK3)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
91	D8	P07(#SRDY1/#DMAEND3)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
92	A11	V <sub>SS</sub>	POWER	–	–	–	–	–	
93	C11	P20(#DRD)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
94	C10	P21(#DWE/#GAAS)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
95	C9	P22(TM0)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
96	B11	P23(TM1)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
97	B10	P24(TM2/#SRDY2)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
98	A10	P25(TM3/#SCLK2)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
99	B9	P26(TM4/SOUT2)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
100	C8	P27(TM5/SIN2)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1

**I-6 S1C33301: ELECTRICAL CHARACTERISTICS**

Pin No.		Signal name	I/O cell name	Input/output characteristic	Output characteristic	I/O	Pull-up/down (resistor value)	Power source	Remarks
QFP	BGA								
101	B5	Vss	POWER	–	–	–	–	–	
102	A9	K50(#DMAREQ0)	HIBHP2TY	LVC MOS Schmitt	–	I	Pull-up(100k)	V <sub>DDE</sub>	note 1
103	B8	K51(#DMAREQ1)	HIBHP2TY	LVC MOS Schmitt	–	I	Pull-up(100k)	V <sub>DDE</sub>	note 1
104	A8	K52(#ADTRG)	HIBHP2TY	LVC MOS Schmitt	–	I	Pull-up(100k)	V <sub>DDE</sub>	note 1
105	C7	K53(#DMAREQ2)	HIBHP2TY	LVC MOS Schmitt	–	I	Pull-up(100k)	V <sub>DDE</sub>	note 1
106	B7	K54(#DMAREQ3)	HIBHP2TY	LVC MOS Schmitt	–	I	Pull-up(100k)	V <sub>DDE</sub>	note 1
107	A7	DSIO	HBBH2BP1TY	LVC MOS Schmitt	4mA	I/O	Pull-up(50k)	V <sub>DDE</sub>	
108	D7	DST0(P10/EXCL0/T8UF0)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
109	B6	DST1(P11/EXCL1/T8UF1)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
110	E6	DST2(P12/EXCL2/T8UF2)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
111	A6	V <sub>DDE</sub>	POWER	–	–	–	–	–	
112	D6	DPCO(P13/EXCL3/T8UF3)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
113	C6	CLK(P14/FOSC1)	HBBH2BP2TY	LVC MOS Schmitt	4mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
114	D5	P15(EXCL4/#DMAEND0/#SCLK3)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
115	C5	P16(EXCL5/#DMAEND1/SOUT3)	HBBH1BP2TY	LVC MOS Schmitt	2mA	I/O	Pull-up(100k)	V <sub>DDE</sub>	note 1
116	C4	Vss	POWER	–	–	–	–	–	
117	A5	OSC2	LL0TY	–	–	O	–	V <sub>DD</sub>	
118	A4	OSC1	LL1NY	–	–	I	–	V <sub>DD</sub>	note 4
119	D4	V <sub>DD</sub>	POWER	–	–	–	–	–	
120	B4	K60(AD0)	HIBSP2TY	LVC MOS	–	I	Pull-up(100k)	AV <sub>DDE</sub>	note 1, 5
121	A3	K61(AD1)	HIBSP2TY	LVC MOS	–	I	Pull-up(100k)	AV <sub>DDE</sub>	note 1, 5
122	B2	K62(AD2)	HIBSP2TY	LVC MOS	–	I	Pull-up(100k)	AV <sub>DDE</sub>	note 1, 5
123	B3	K63(AD3)	HIBSP2TY	LVC MOS	–	I	Pull-up(100k)	AV <sub>DDE</sub>	note 1, 5
124	–	K64(AD4)	HIBSP2TY	LVC MOS	–	I	Pull-up(100k)	AV <sub>DDE</sub>	note 1, 5, 6
125	–	K65(AD5)	HIBSP2TY	LVC MOS	–	I	Pull-up(100k)	AV <sub>DDE</sub>	note 1, 5, 6
126	–	K66(AD6)	HIBSP2TY	LVC MOS	–	I	Pull-up(100k)	AV <sub>DDE</sub>	note 1, 5, 6
127	–	K67(AD7)	HIBSP2TY	LVC MOS	–	I	Pull-up(100k)	AV <sub>DDE</sub>	note 1, 5, 6
128	A2	AV <sub>DDE</sub>	POWER	LVC MOS	–	–	–	–	

note 1) The pull-up resistor can be enabled/disabled using the I/O register (pull-up is enabled by default).

note 2) A bus hold latch is attached.

note 3) This pin is set as an input pin during device testing. Normally it is an output pin.

note 4) The voltage applied to this pin must be  $0V \leq V_{IN} \leq V_{DD}$ .

note 5) The voltage applied to this pin must be  $0V \leq V_{IN} \leq AV_{DDE}$ .

note 6) Available in the QFP15-128pin package. These pins are not provided in the PFBGA-121pin package. The pull-up resistors for these pins must be enabled when the PFBGA-121pin package is used.

## AC Characteristics

### Symbol Description

#### tcyc: Bus-clock cycle time

- In x1 mode, tcyc = 50 ns (20 MHz) when the CPU is operated with a 20-MHz clock  
tcyc = 30 ns (33 MHz) when the CPU is operated with a 33-MHz clock
- In x2 mode, tcyc = 50 ns (20 MHz) when the CPU is operated with a 40-MHz clock  
tcyc = 40 ns (25 MHz) when the CPU is operated with a 50-MHz clock

#### WC: Number of wait cycles

Up to 7 cycles can be set for the number of cycles using the BCU control register. Furthermore, it can be extended to a desired number of cycles by setting the #WAIT pin from outside of the IC.

The minimum number of read cycles with no wait (0) inserted is 1 cycle.

The minimum number of write cycles with no wait cycle (0) inserted is 2 cycles. It does not change even if 1-wait cycle is set. The write cycle is actually extended when 2 or more wait cycles are set.

When inserting wait cycles by controlling the #WAIT pin from outside of the IC, pay attention to the timing of the #WAIT signal sampling. Read cycles are terminated at the cycle in which the #WAIT signal is negated.

Write cycles are terminated at the following cycle after the #WAIT signal is negated.

#### C1, C2, C3, Cn: Cycle number

C1 indicates the first cycle when the BCU transfers data from/to an external memory or another device. Similarly, C2 and Cn indicate the second cycle and nth cycle, respectively.

#### Cw: Wait cycle

Indicates that the cycle is wait cycle inserted.

### AC Characteristics Measurement Condition

Signal detection level: Input signal High level  $V_{IH} = V_{DDE} - 0.4 \text{ V}$   
Low level  $V_{IL} = 0.4 \text{ V}$

Output signal High level  $V_{OH} = 1/2 V_{DDE}$   
Low level  $V_{OL} = 1/2 V_{DDE}$

The following applies when OSC3 is external clock input:

Input signal High level  $V_{IH} = 1/2 V_{DD}$   
Low level  $V_{IL} = 1/2 V_{DD}$

Input signal waveform: Rise time (10% → 90%  $V_{DD}$ ) 5 ns  
Fall time (90% → 10%  $V_{DD}$ ) 5 ns

Output load capacitance:  $C_L = 50 \text{ pF}$

## C33 Block AC Characteristic Tables

### External clock input characteristics

(Note) These AC characteristics apply to input signals from outside the IC.

The OSC3 input clock must be within VDD to VSS voltage range.

(Unless otherwise specified: V<sub>DDE</sub>=2.7V to 3.6V, V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, T<sub>a</sub>=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
High-speed clock cycle time	t <sub>C3</sub>	30	500	ns	
OSC3 clock input duty	t <sub>C3ED</sub>	45	55	%	
OSC3 clock input rise time	t <sub>IF</sub>		5	ns	
OSC3 clock input fall time	t <sub>IR</sub>		5	ns	
BCLK high-level output delay time	t <sub>CD1</sub>		35	ns	
BCLK low-level output delay time	t <sub>CD2</sub>		35	ns	
Minimum reset pulse width	t <sub>RST</sub>	6·t <sub>CYC</sub>		ns	

### BCLK clock output characteristics

(Note) These AC characteristic values are applied only when the high-speed oscillation circuit is used.

(Unless otherwise specified: V<sub>DDE</sub>=2.7V to 3.6V, V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, T<sub>a</sub>=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
BCLK clock output duty	t <sub>CBD</sub>	40	60	%	

### Common characteristics

(Unless otherwise specified: V<sub>DDE</sub>=2.7V to 3.6V, V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, T<sub>a</sub>=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
Address delay time	t <sub>AD</sub>	–	10	ns	1
#CE <sub>x</sub> delay time (1)	t <sub>CE1</sub>	–	10	ns	
#CE <sub>x</sub> delay time (2)	t <sub>CE2</sub>	–	10	ns	
Wait setup time	t <sub>WTS</sub>	18	–	ns	
Wait hold time	t <sub>WTH</sub>	0	–	ns	
Read signal delay time (1)	t <sub>RD1</sub>		10	ns	2
Read data setup time	t <sub>RDS</sub>	18		ns	
Read data hold time	t <sub>RDH</sub>	0		ns	
Write signal delay time (1)	t <sub>WRD1</sub>		10	ns	3
Write data delay time (1)	t <sub>WDD1</sub>		10	ns	
Write data delay time (2)	t <sub>WDD2</sub>	0	10	ns	
Write data hold time	t <sub>WDH</sub>	0		ns	

- \* note 1) This applies to the #BSH and #BSL timings.  
 2) This applies to the #GAAS and #GARD timings.  
 3) This applies to the #GAAS timing.

### SRAM read cycle

(Unless otherwise specified: V<sub>DDE</sub>=2.7V to 3.6V, V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, T<sub>a</sub>=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
Read signal delay time (2)	t <sub>RD2</sub>		10	ns	
Read signal pulse width	t <sub>RDW</sub>	t <sub>CYC</sub> (0.5+WC)-10		ns	
Read address access time (1)	t <sub>ACC1</sub>		t <sub>CYC</sub> (1+WC)-25	ns	
Chip enable access time (1)	t <sub>CEAC1</sub>		t <sub>CYC</sub> (1+WC)-25	ns	
Read signal access time (1)	t <sub>RDAC1</sub>		t <sub>CYC</sub> (0.5+WC)-25	ns	

### SRAM write cycle

(Unless otherwise specified: V<sub>DDE</sub>=2.7V to 3.6V, V<sub>DD</sub>=1.65V to 1.95V, V<sub>SS</sub>=0V, T<sub>a</sub>=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
Write signal delay time (2)	t <sub>WRD2</sub>		10	ns	
Write signal pulse width	t <sub>WRW</sub>	t <sub>CYC</sub> (1+WC)-10		ns	

**Burst ROM read cycle**(Unless otherwise specified:  $V_{DDE}=2.7V$  to  $3.6V$ ,  $V_{DD}=1.65V$  to  $1.95V$ ,  $V_{SS}=0V$ ,  $T_a=-40^{\circ}C$  to  $+85^{\circ}C$ )

Item	Symbol	Min.	Max.	Unit	*
Read address access time (2)	t <sub>ACC2</sub>		t <sub>cy</sub> (1+WC)-25	ns	
Chip enable access time (2)	t <sub>CEAC2</sub>		t <sub>cy</sub> (1+WC)-25	ns	
Read signal access time (2)	t <sub>RDAC2</sub>		t <sub>cy</sub> (0.5+WC)-25	ns	
Burst address access time	t <sub>ACCB</sub>		t <sub>cy</sub> (1+WC)-25	ns	

**External bus master and NMI**(Unless otherwise specified:  $V_{DDE}=2.7V$  to  $3.6V$ ,  $V_{DD}=1.65V$  to  $1.95V$ ,  $V_{SS}=0V$ ,  $T_a=-40^{\circ}C$  to  $+85^{\circ}C$ )

Item	Symbol	Min.	Max.	Unit	*
#BUSREQ signal setup time	t <sub>BRQS</sub>	18		ns	
#BUSREQ signal hold time	t <sub>BRQH</sub>	0		ns	
#BUSACK signal output delay time	t <sub>BAKD</sub>		10	ns	
High-impedance → output delay time	t <sub>ZZE</sub>		10	ns	
Output → high-impedance delay time	t <sub>BZZ</sub>		10	ns	
#NMI pulse width	t <sub>NMIW</sub>	30		ns	

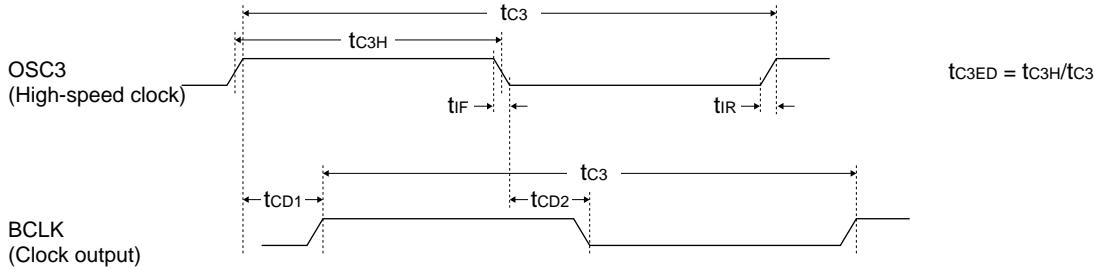
**Input, Output and I/O port**(Unless otherwise specified:  $V_{DDE}=2.7V$  to  $3.6V$ ,  $V_{DD}=1.65V$  to  $1.95V$ ,  $V_{SS}=0V$ ,  $T_a=-40^{\circ}C$  to  $+85^{\circ}C$ )

Item	Symbol	Min.	Max.	Unit	*
Input data setup time	t <sub>INPS</sub>	20		ns	
Input data hold time	t <sub>INPH</sub>	10		ns	
Output data delay time	t <sub>OUTD</sub>		20	ns	
K-port interrupt input pulse width	SLEEP, HALT2 mode	t <sub>KINW</sub>	30	ns	
	Others		2 × t <sub>cy</sub>	ns	

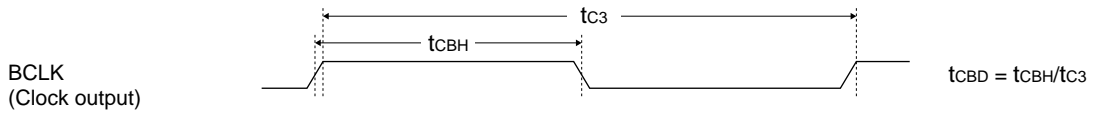
### C33 Block AC Characteristic Timing Charts

#### Clock

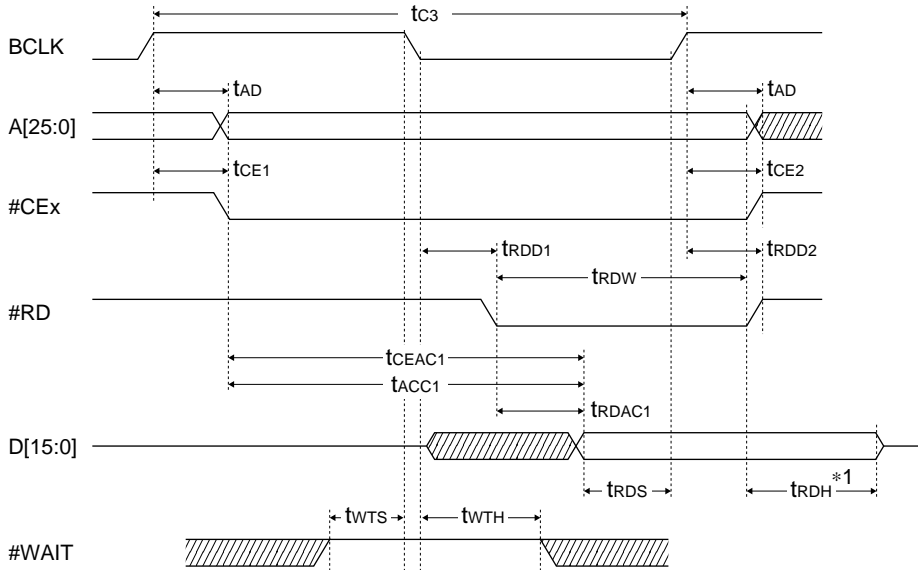
(1) When an external clock is input (in x1 speed mode):



(2) When the high-speed oscillation circuit is used for the operating clock:

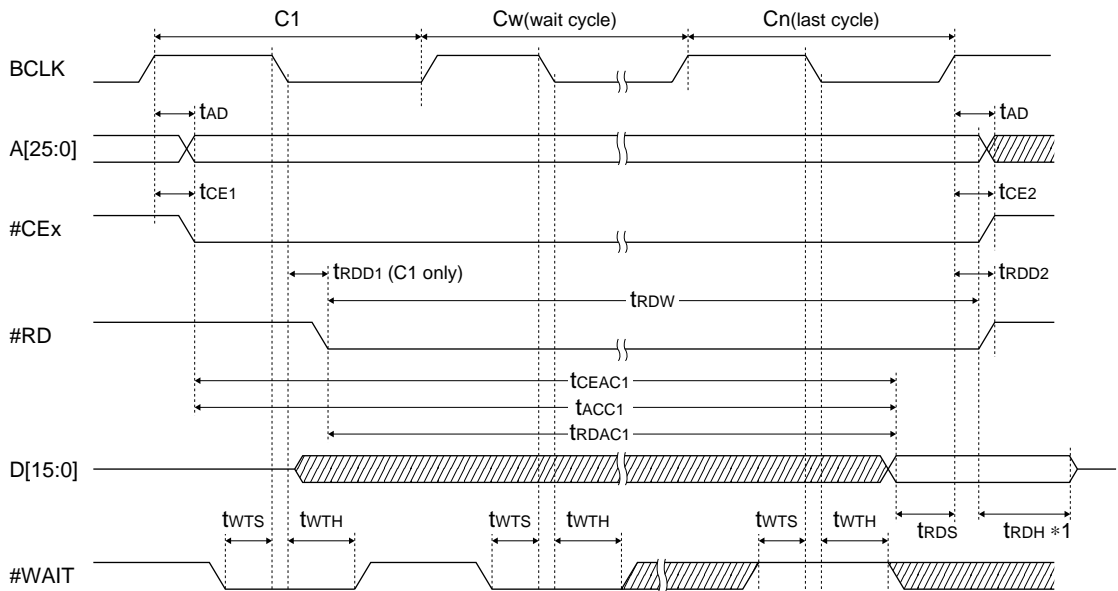


## SRAM read cycle (basic cycle: 1 cycle)



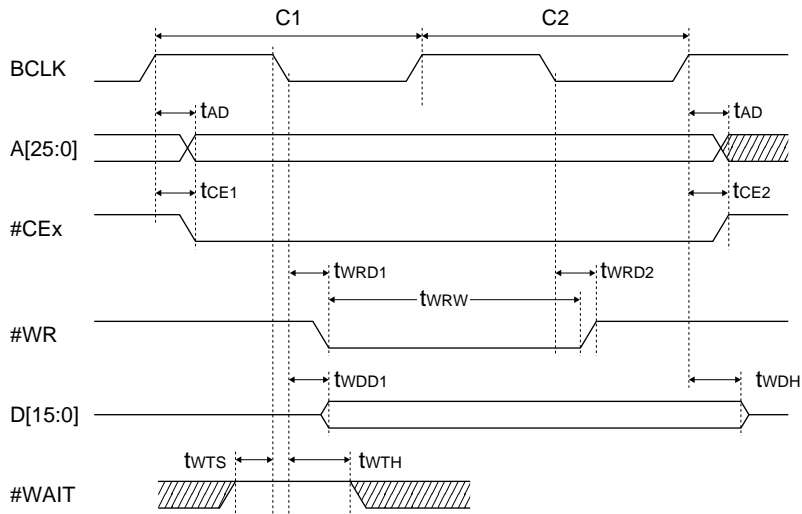
\*1  $t_{RDH}$  is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[25:0] signals.

## SRAM read cycle (when a wait cycle is inserted)

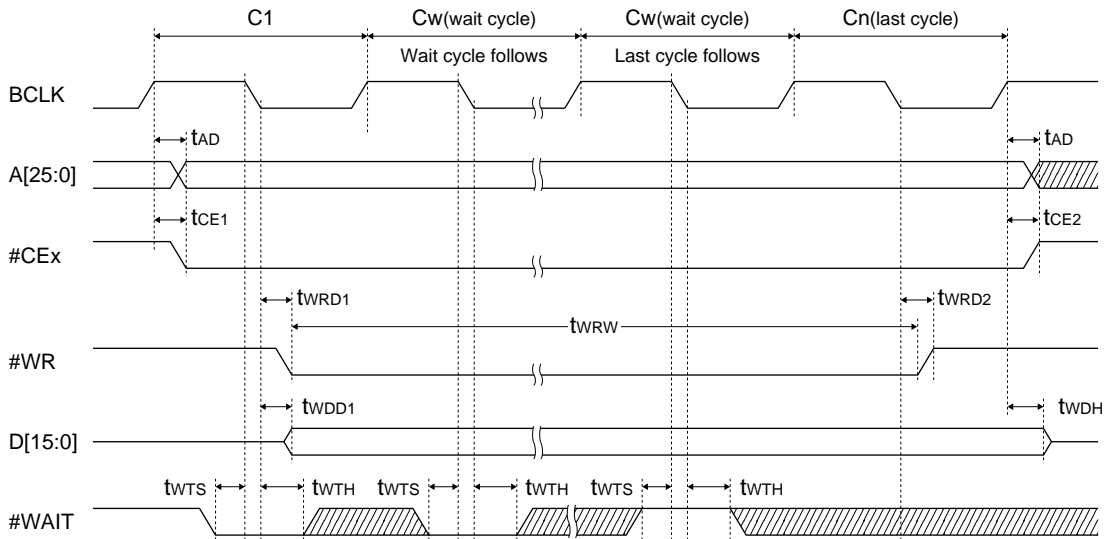


\*1  $t_{RDH}$  is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[25:0] signals.

**SRAM write cycle (basic cycle: 2 cycles)**

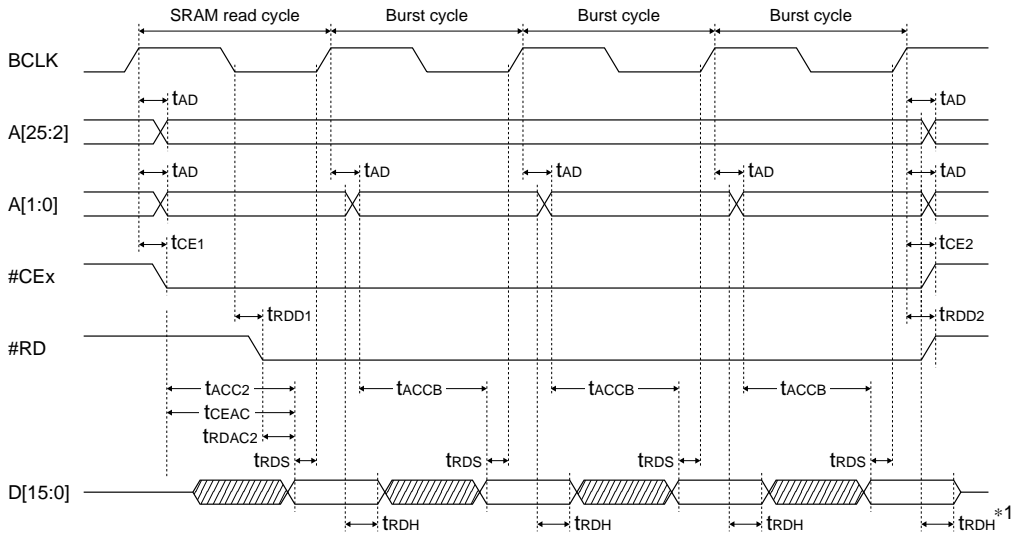


**SRAM write cycle (when wait cycles are inserted)**



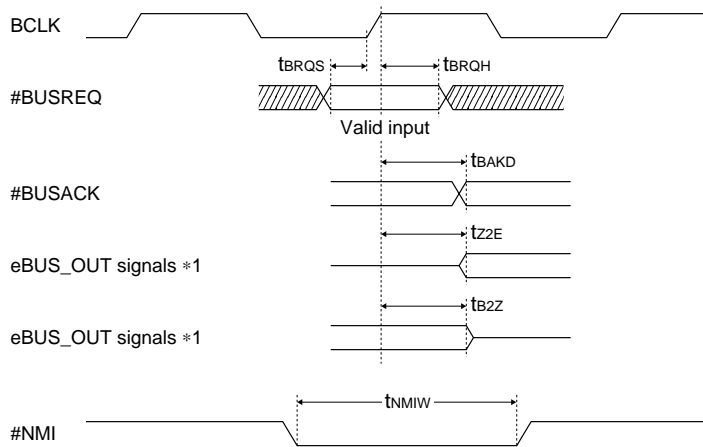


### Burst ROM read cycle



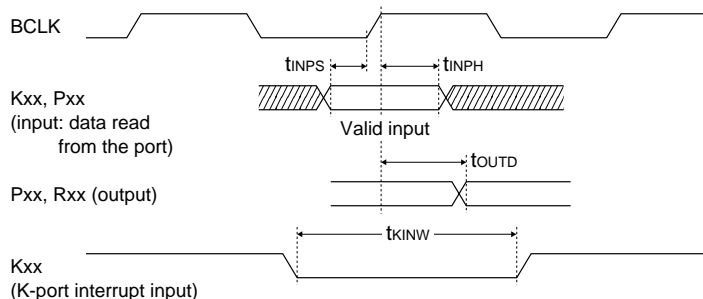
\*1  $t_{RDH}$  is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[25:0] signals.

### #BUSREQ, #BUSACK and #NMI timing



\*1 eBUS\_OUT indicates the following pins:  
A[25:0], #RD, #WRL, #WRH, #HCAS, #LCAS, #CE[17:4], D[15:0]

### Input, output and I/O port timing



## **<Reference> External Device Interface Timings**

---

This section shows setup examples for setting timing conditions of the external system interface as a reference material used when configuring a system with external devices.

Pay attention to the following precautions when using this material.

- The described AC characteristic values of external devices are standard values. They may differ from those of the devices actually used, so the actual setup values (number of cycles) should be determined by referring the manual or specification of the device to be used.
- It is necessary to set the timing values allowing ample margin according to the load capacitance of the bus and signal lines, number of devices to be connected, operating temperature range, I/O levels and other conditions. The number of cycles described in this section is an example and the conditions are not considered.
- The values described in "Time" column of the tables are simply calculated by multiplying the number of cycles by the cycle time. Conditions such as the output delay time of the device, delay due to wiring and load capacitance, and input setup time are not considered.
- The described contents are reference data and cannot be guaranteed to work.

## ROM and Burst ROM

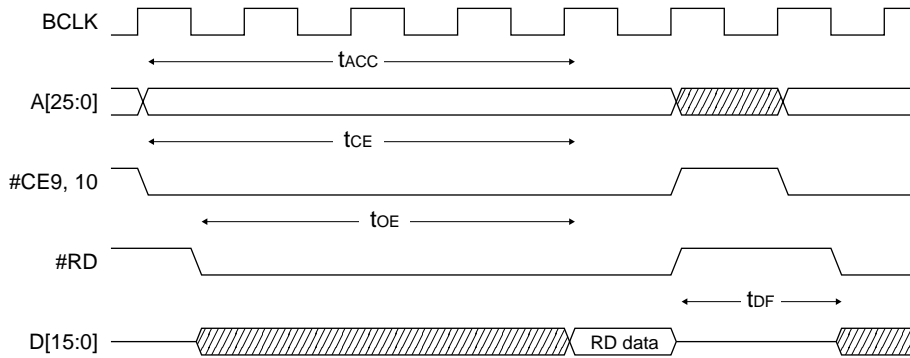
### Burst ROM and mask ROM interface setup examples

Operating frequency	Normal read cycle		Burst read cycle		Output disable delay cycle
	Wait cycle	Read cycle	Wait cycle	Read cycle	
20MHz	2	3	1	2	1.5
25MHz	3	4	1	2	1.5
33MHz	4	5	2	3	1.5

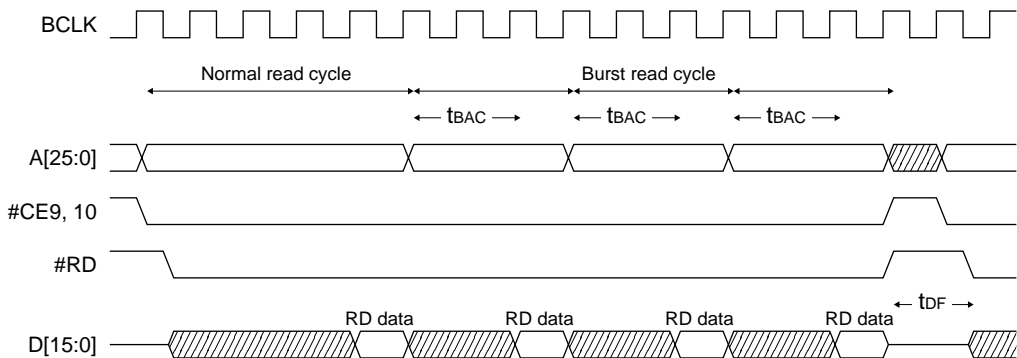
### Burst ROM and mask ROM interface timing

Burst ROM and mask ROM interface				33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time	Cycle	Time	Cycle	Time
Access time	$t_{ACC}$	–	100	5	150	4	160	3	150
#CE output delay time	$t_{CE}$	–	100	5	150	4	160	3	150
#OE output delay time	$t_{OE}$	–	50	4.5	135	3.5	140	2.5	125
Burst access time	$t_{BAC}$	–	50	3	90	2	80	2	100
Output disable delay time	$t_{DF}$	0	40	1.5	45	1.5	60	1.5	75

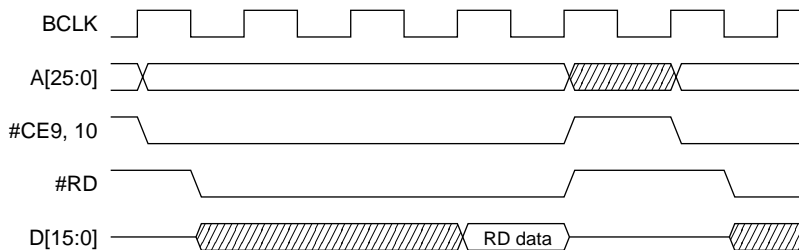
#### ROM: 100ns, CPU: 33MHz, normal read



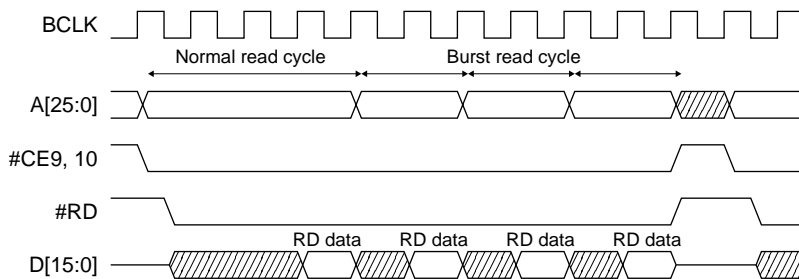
#### ROM: 100ns, CPU: 33MHz, burst read



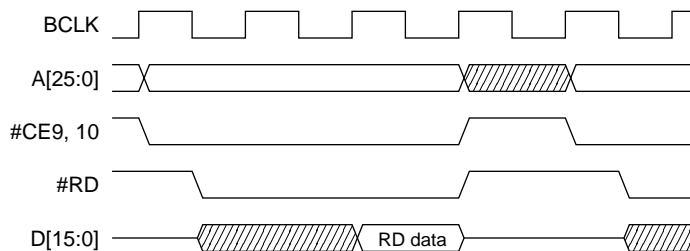
**ROM: 100ns, CPU: 25MHz, normal read**



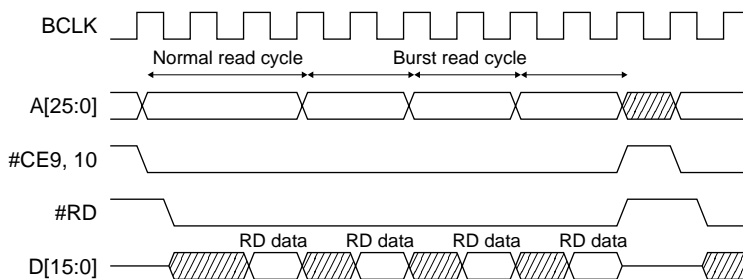
**ROM: 100ns, CPU: 25MHz, burst read**



**ROM: 100ns, CPU: 20MHz, normal read**



**ROM: 100ns, CPU: 20MHz, burst read**

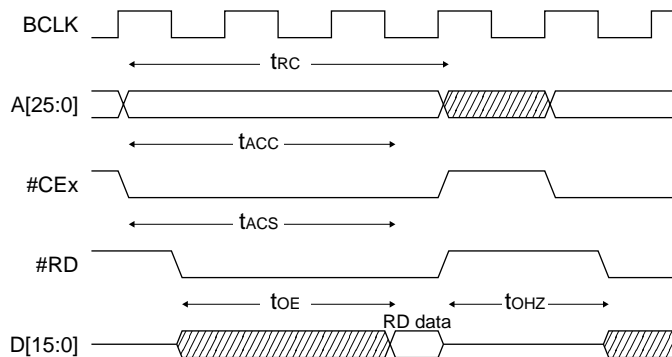
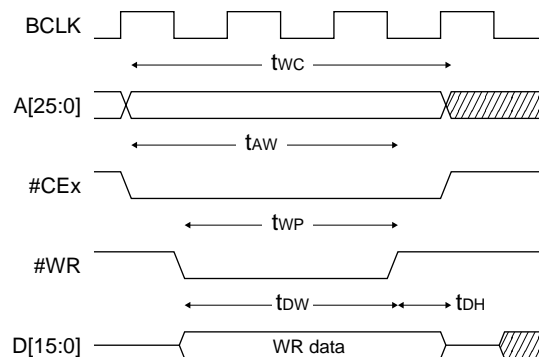


**SRAM (55ns)****SRAM interface setup examples – 55ns**

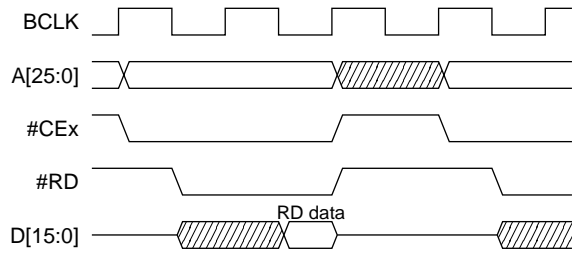
Operating frequency	Read cycle		Write cycle	Output disable delay cycle
	Wait cycle	Read cycle		
20MHz	1	2	2	1.5
25MHz	2	3	3	1.5
33MHz	2	3	3	1.5

**SRAM interface timing – 55ns**

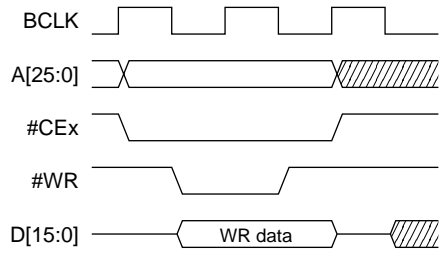
SRAM interface					33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.		Cycle	Time	Cycle	Time	Cycle	Time
<b>&lt;Read cycle&gt;</b>										
Read cycle time	t <sub>RC</sub>	55	–		3	90	3	120	2	100
Address access time	t <sub>ACC</sub>	–	55		3	90	3	120	2	100
#CE access time	t <sub>ACS</sub>	–	55		3	90	3	120	2	100
#OE access time	t <sub>OE</sub>	–	30		2.5	75	2.5	100	1.5	75
Output disable delay time	t <sub>OHZ</sub>	0	30		1.5	45	1.5	60	1.5	75
<b>&lt;Write cycle&gt;</b>										
Write cycle time	t <sub>WC</sub>	55	–		3	90	3	120	2	100
Address enable time	t <sub>AW</sub>	50	–		2.5	75	2.5	100	1.5	75
Write pulse width	t <sub>WP</sub>	45	–		2	60	2	80	1	50
Input data setup time	t <sub>DW</sub>	30	–		2	60	2	80	1	50
Input data hold time	t <sub>DH</sub>	0	–		0.5	15	0.5	20	0.5	25

**SRAM: 55ns, CPU: 33/25MHz, read cycle****SRAM: 55ns, CPU: 33/25MHz, write cycle**

**SRAM: 55ns, CPU: 20MHz, read cycle**



**SRAM: 55ns, CPU: 20MHz, write cycle**

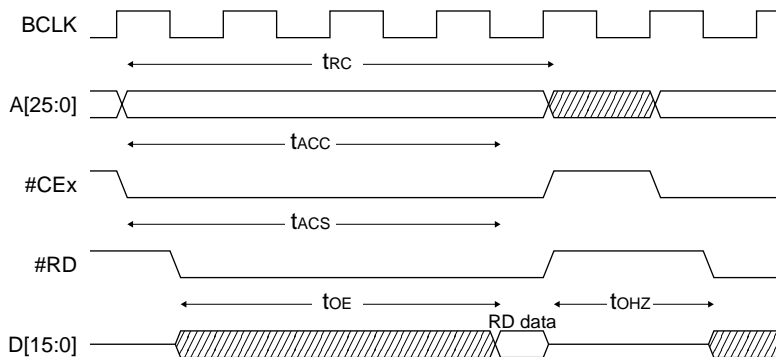
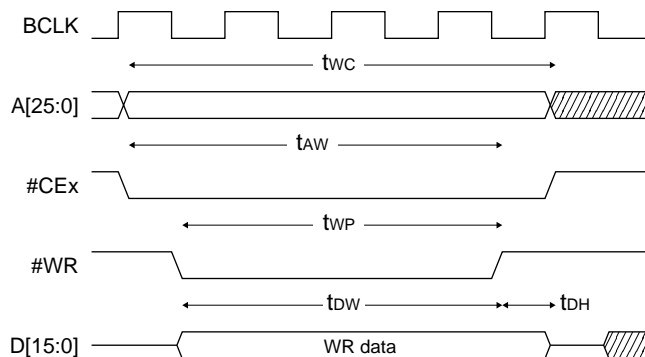


**SRAM (70ns)****SRAM interface setup examples – 70ns**

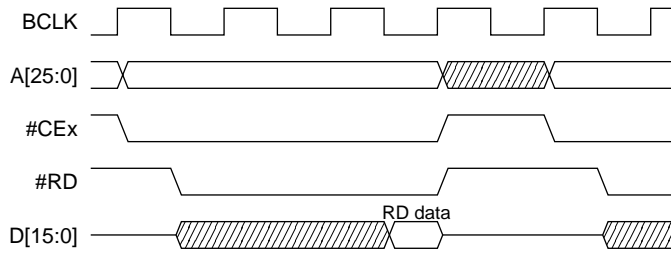
Operating frequency	Read cycle		Write cycle	Output disable delay cycle
	Wait cycle	Read cycle		
20MHz	2	3	3	1.5
25MHz	2	3	3	1.5
33MHz	3	4	4	1.5

**SRAM interface timing – 70ns**

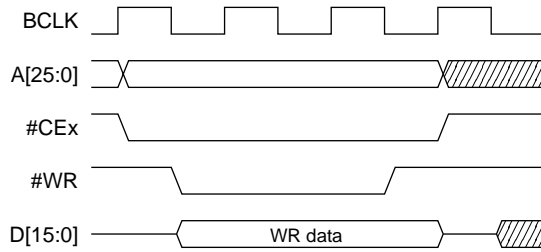
SRAM interface					33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time	Cycle	Time	Cycle	Time	
<b>&lt;Read cycle&gt;</b>										
Read cycle time	t <sub>RC</sub>	70	–	4	120	3	120	3	150	
Address access time	t <sub>ACC</sub>	–	70	4	120	3	120	3	150	
#CE access time	t <sub>ACS</sub>	–	70	4	120	3	120	3	150	
#OE access time	t <sub>OE</sub>	–	40	3.5	105	2.5	100	2.5	125	
Output disable delay time	t <sub>OHZ</sub>	0	30	1.5	45	1.5	60	1.5	75	
<b>&lt;Write cycle&gt;</b>										
Write cycle time	t <sub>WC</sub>	70	–	4	120	3	120	3	150	
Address enable time	t <sub>AW</sub>	60	–	3.5	105	2.5	100	2.5	125	
Write pulse width	t <sub>WP</sub>	55	–	3	90	2	80	2	100	
Input data setup time	t <sub>DW</sub>	30	–	3	90	2	80	2	100	
Input data hold time	t <sub>DH</sub>	0	–	0.5	15	0.5	20	0.5	25	

**SRAM: 70ns, CPU: 33MHz, read cycle****SRAM: 70ns, CPU: 33MHz, write cycle**

**SRAM: 70ns, CPU: 25/20MHz, read cycle**



**SRAM: 70ns, CPU: 25/20MHz, write cycle**





## 8255A

### 8255A interface setup examples

Operating frequency	Read cycle		Write cycle	Output disable delay cycle
	Wait cycle	Read cycle		
20MHz	9 *1	10	10	3.5
25MHz	11	12	12	3.5
33MHz	14	15	15	3.5 *2

### 8255A interface timing

SRAM interface					33MHz		25MHz		20MHz	
Parameter	Symbol	Min.	Max.	Cycle	Time	Cycle	Time	Cycle	Time	
<b>&lt;Read cycle&gt;</b>										
Read cycle time	t <sub>RC</sub>	300	–	15	450	12	480	10	500	
Address access time	t <sub>ACC</sub>	–	250	15	450	12	480	10	500	
#CE access time	t <sub>ACS</sub>	–	250	15	450	12	480	10	500	
#OE access time	t <sub>OE</sub>	–	250	14.5	435	11.5	460	9.5	475	
Output disable delay time	t <sub>OHZ</sub>	10	150	3.5	105	3.5	140	3.5	175	
<b>&lt;Write cycle&gt;</b>										
Write cycle time	t <sub>WC</sub>	430	–	15	450	12	480	10	500	
Address enable time	t <sub>AW</sub>	400	–	14.5	435	11.5	460	9.5	475	
Write pulse width	t <sub>WP</sub>	400	–	14	420	11	440	9	450	
Input data setup time	t <sub>DW</sub>	100	–	14	420	11	440	9	450	
Input data hold time	t <sub>DH</sub>	30	–	0.5	15	0.5	20	0.5	25	

- \*1 The S1C33301 enables up to 7 cycles of wait-cycle insertion. If a number of wait cycles more than 7 cycles needs to be inserted, input the #WAIT signal from external hardware. Note that the interface must be set for SRAM type devices to insert wait cycles using the #WAIT pin. (Refer to Section II-4, "BCU (Bus Control Unit)", for more information.)
- \*2 This setting cannot satisfy the 150 ns of output-disable delay time specification required for the 8255A. When implementing such a low-speed device in the system, the external bus must be separated by inserting a 3-state bus buffer at the output side (when viewed from the CPU) of the external system bus.
- \*3 If the data hold time that can be set is not sufficient for the device, secure it by connecting a bus repeater to the external data bus D[15:0] or by inserting a latch at the output side of the external system interface.

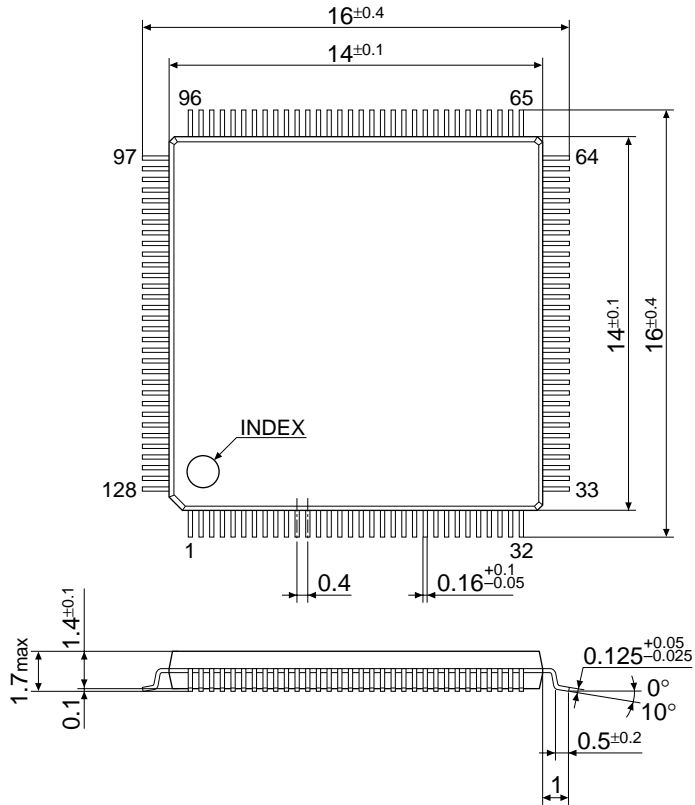
THIS PAGE IS BLANK.

# I-7 Package

## Plastic Package

QFP15-128pin

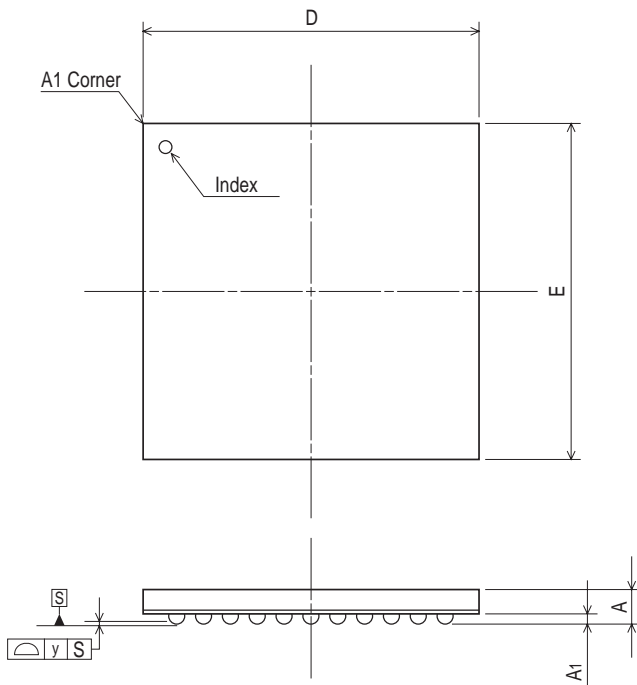
(Unit: mm)



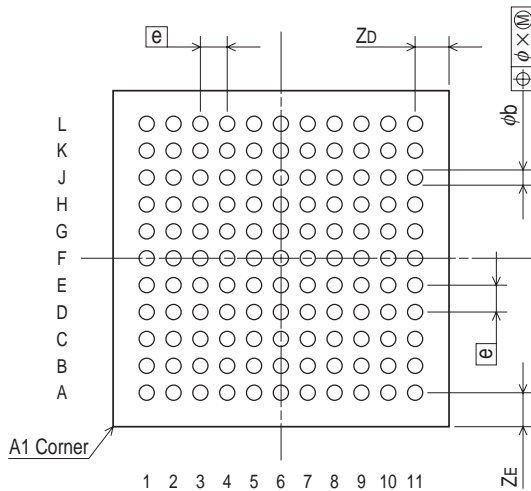
Package

PFBGA-121pin

Top View



Bottom View



Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.8	10.0	10.2
E	9.8	10.0	10.2
A			1.20
A1	0.25	0.30	0.35
e		0.80	
b	0.38	0.43	0.48
X			0.08
Y			0.10
Zb		1.00	
ZE		1.00	

1 = 1mm

1.0	'03/2/12	T. NAKAYAMA	J. TANIGUCHI	H. KIKUCHI	
REV.	DATE	DRAW	CHECKED	APPROVED	CORRECTION ARTICLE
PFBGA10UX121 (P-TFBGA-121-1010-0.8) (PFBGA 10x10 t=1.2 121PIN)					CAD FILE
					DWG No. 4900-0350
<b>SEIKO EPSON CORP.</b>					SCALE Free SHEET 1 of 1

2900-0002-01 (Rev. 1.1)

## Thermal Package Resistance

The chip temperature of an LSI rises according to power consumption. The chip temperature can be calculated from environment temperature ( $T_a$ ), thermal package resistance ( $\theta$ ) and power consumption ( $P_D$ ).

$$\text{Chip temperature } (T_j) = T_a + (P_D \times \theta) \text{ (}^\circ\text{C)}$$

As a guide, normally keep the chip temperature ( $T_j$ ) lower than 100°C.

### Thermal resistance of the QFP15-128pin package

#### 1. Mounted on the board (without air-cooling)

$$\text{Thermal resistance } (\theta_{j-a}) = 33.3^\circ\text{C/W}$$

This thermal resistance is the value under the condition that the measured device is mounted on the board for measurement (dimensions: 114 × 76 × 1.6t mm, FR4/4-layer board) and has no air-cooling.

#### 2. Hanging in the air (without air-cooling)

$$\text{Thermal resistance} = 90 \text{ to } 100^\circ\text{C/W}$$

This thermal resistance is the value under the condition that the measured device is hanging in the air and has no air-cooling.

### Thermal resistance of the PFBGA-128pin (PFBGA10UX121) package

#### 1. Mounted on the board (without air-cooling)

$$\text{Thermal resistance } (\theta_{j-a}) = 30^\circ\text{C/W}$$

This thermal resistance is the value under the condition that the measured device is mounted on the board for measurement (dimensions: 114.5 × 101.5 × 1.6t mm, FR4/4-layer board) and has no air-cooling.

#### 2. Hanging in the air (without air-cooling)

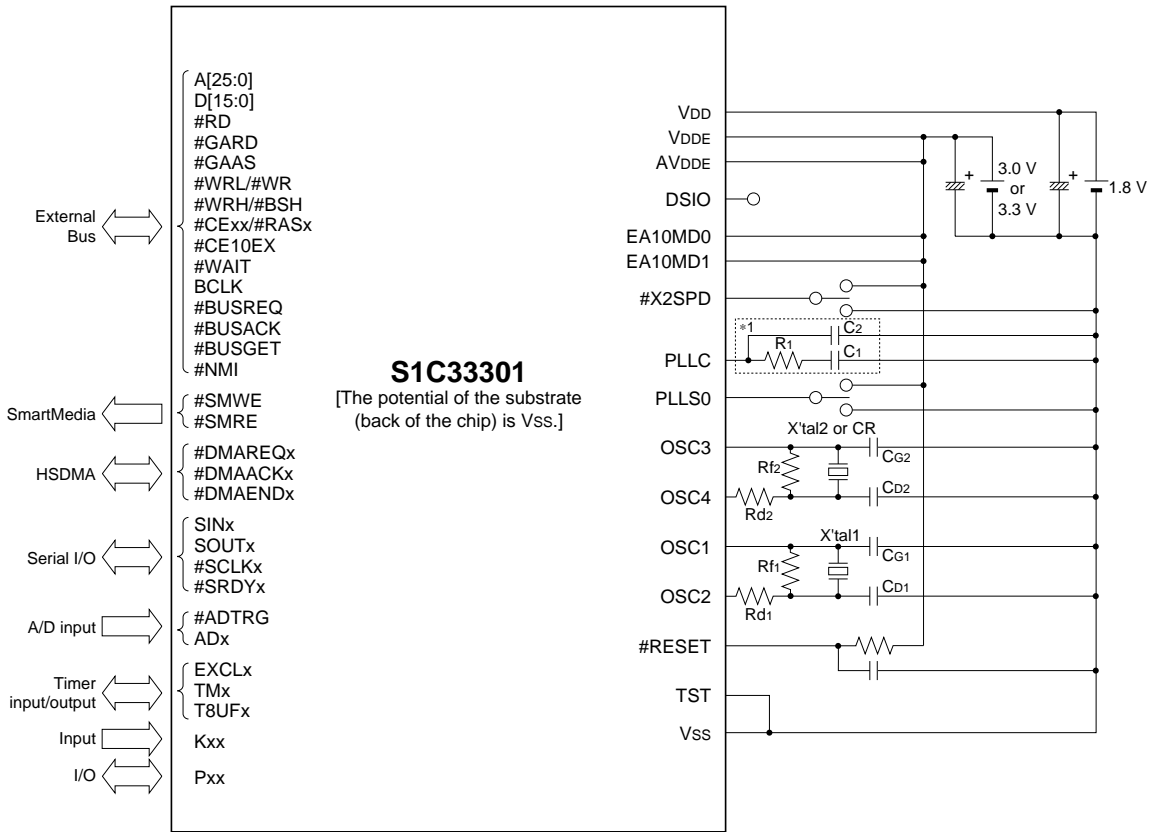
$$\text{Thermal resistance} = 165^\circ\text{C/W}$$

This thermal resistance is the value under the condition that the measured device is hanging in the air and has no air-cooling.

**Note:** Thermal resistance greatly varies according to the mounting condition on the board and air-cooling condition.

THIS PAGE IS BLANK.

# I-8 Basic External Wiring Diagram



Wiring

X'tal1	Crystal oscillator	32.768 kHz
CG1	Gate capacitor	10 pF
CD1	Drain capacitor	10 pF
Rf1	Feedback resistor	10 MΩ
Rd1	Drain resistor	0 Ω
X'tal2	Crystal oscillator	33 MHz (Max.)
CR	Ceramic oscillator	33 MHz (Max.)
CG2	Gate capacitor	10 pF
CD2	Drain capacitor	10 pF
Rf2	Feedback resistor	1 MΩ
Rd2	Drain resistor	0 Ω
R1	Resistor	4.7 kΩ
C1	Capacitor	100 pF
C2	Capacitor	3 pF

\*1: When the PLL is not used, leave the PLLC pin open.

Note: The above table is simply an example, and is not guaranteed to work.

THIS PAGE IS BLANK.



## I-9 Precautions on Mounting

The following shows the precautions when designing the board and mounting the IC.

### Oscillation Circuit

- Oscillation characteristics change depending on conditions such as components used (oscillator,  $R_f$ ,  $R_d$ ,  $C_G$ ,  $C_D$ ) and board pattern. In particular, when a ceramic or crystal oscillator is used, evaluate the components adequately under real operating conditions by mounting them on the board before the external register ( $R_f$ ,  $R_d$ ) and capacitor ( $C_G$ ,  $C_D$ ) values are finally decided.
- Disturbances of the oscillation clock due to noise may cause a malfunction. To prevent this, the following points should be taken into consideration. In particular, the latest devices are more sensitive to noise, as they are more finely processed.

The measures against noise for the OSC2 and PLLC pins, and the components and lines connected to these pins are most essential, and similar measures must also be taken for the OSC1 pins. The measures for the OSC1, OSC2, and PLLC pins are described below.

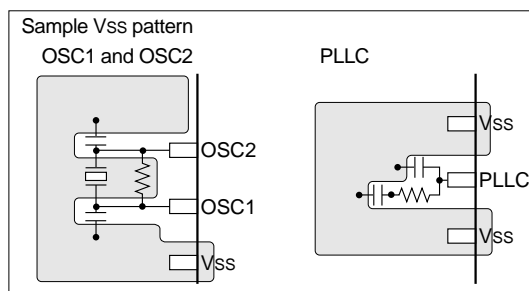
We recommend taking measures similar to those for the high-speed oscillation system, including the OSC3 and OSC4 pins and the components and lines connected to these pins.

- (1) Components that are connected to the OSC1, OSC2, and PLLC pins, such as oscillators, resistors, and capacitors, should be connected in the shortest line.
- (2) Whenever possible, configure digital signal lines with at least three millimeters clearance from the OSC1, OSC2, and PLLC pins and the components and lines connected to these pins. In particular, signals that are switched frequently must not be placed near these pins, components, and lines. The same applies to all layers on the multi-layered board as the distance between the layers is around 0.1 to 0.2 mm. Furthermore, do not configure digital signal lines in parallel with these components and lines when arranging them on the same or another layer of the board. Such an arrangement is strictly prohibited, even with clearance of three millimeters or more. Also, avoid arranging digital signal lines across these components and signal lines.
- (3) Shield the OSC1, OSC2, and PLLC pins and lines connected to those pins as well as the adjacent layers of the board using Vss.

As shown in the figure below, shield the wired layers as much as possible.

Whenever possible, make the whole adjacent layers the ground layers, or ensure there is adequate shielding to a radius of five millimeters around the above pins and lines.

As described in (2), do not configure digital signal lines in parallel with components and lines even if such precautionary measures are taken, and avoid configuring signal lines that are switched frequently across components and lines on other layers.



- (4) Usage of pins near the OSC1, OSC2, and PLLC pins should be limited. On S1C33301, the P15 and P16 pins may generate noise at the OSC2 pin. Whenever possible, do not use the P15 and P16 ports if using the low-speed (OSC1) oscillation circuit. When using these ports for output, assign signals that are rarely changed, and when using them for input, assign signals that are rarely changed or in which rising and falling times exceed 100 ns. To decrease the frequency of the input signal switching, raise the impedance by introducing 1 to 2 k $\Omega$  resistance serially.

- (5) When an external clock is supplied to the OSC1 or OSC3 pin, the clock source should be connected to the OSC1 or OSC3 pin in the shortest line. Furthermore, do not connect anything else to the OSC2 or OSC4 pin.
- (6) After taking the above precautions, check the output clock waveform while operating the actual application program in the actual device.

To do this, measure the output of the BCLK (FOSC1/P60) pins with an oscilloscope.

Check the waveform quality at the OSC3 or PLL (when PLLS0 = "1") output clock by measuring the BCLK output (default output of the pin above). Ensure that the frequencies are as designed and there is no noise.

Furthermore, measure the jitter (the jitter should be below 1 ns when the PLL is used).

Check the waveform quality at the OSC1 clock by measuring the FOSC1 output (the BCLK output will be switched to the FOSC1 output when D9/0x48132 = "1" and D[1:0]/0x30004C = "10"). Scale up the ranges around the rising and falling edges of the clock pulse to ensure that there is no noise, such as clock and spike, in the 100 ns ranges.

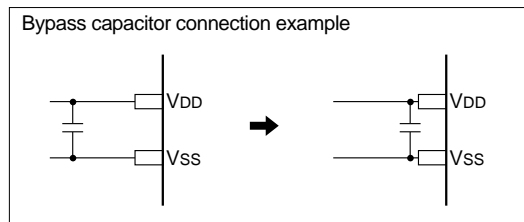
If conditions (1) to (3) are not satisfied, the PLL output may be jittery and the OSC1 output may be noisy. When the PLL output is jittery, the operating frequency will be lowered. When the OSC1 output is noisy, operation of the timer using the OSC1 clock and the CPU core after the system clock is switched to the OSC1 pin will be unstable.

### Reset Circuit

- The power-on reset signal which is input to the #RESET pin changes depending on conditions (power rise time, components used, board pattern, etc.). Decide the time constant of the capacitor and resistor after enough tests have been completed with the application product.
- In order to prevent any occurrences of unnecessary resetting caused by noise during operating, components such as capacitors and resistors should be connected to the #RESET pin in the shortest line.

### Power Supply Circuit

- Sudden power supply variation due to noise may cause malfunction. Consider the following points to prevent this:
  - (1) The power supply should be connected to the VDD, VDDE, VSS and AVDDE pins with patterns as short and large as possible. In particular, the power supply for AVDDE affects A/D conversion precision.
  - (2) When connecting between the VDD and VSS pins with a bypass capacitor, the pins should be connected as short as possible.

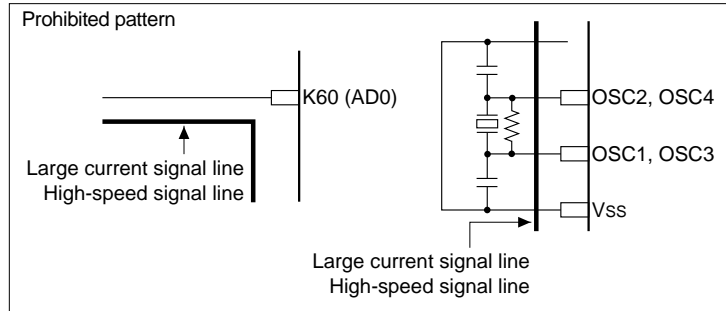


### A/D Converter

- When the A/D converter is not used, the power supply pin AVDDE for the analog system should be connected to VDDE.

## Arrangement of Signal Lines

- In order to prevent generation of electromagnetic induction noise caused by mutual inductance, do not arrange a large current signal line near the circuits that are sensitive to noise such as the oscillation unit and analog input unit.
- When a signal line is parallel with a high-speed line in long distance or intersects a high-speed line, noise may be generated by mutual interference between the signals and it may cause a malfunction.  
Do not arrange a high-speed signal line especially near circuits that are sensitive to noise such as the oscillation unit and analog input unit.



## Reference

Refer to Chapter 4, "The Basic S1C33 Chip Board Circuit", in the "S1C33 Family Application Note" for more detailed precautions on the power supply, oscillation, reset, memory, port, and debug.

## Other

The 0.18  $\mu\text{m}$  fine-pattern process is employed to manufacture this series of products.

Although the product is designed to meet EIAJ and MIL standards regarding basic IC reliability, please pay careful attention to the following points when actually mounting the chip on a board.

Since all OSC and PLLC pins are constructed to use the internal 0.18  $\mu\text{m}$  transistors directly, the pins are susceptible to mechanical damage during the board-mounting process. Moreover, the pins may also be susceptible to electrical damage caused by such disturbances (listed below) whose electrical strength, varying gradually with time, could exceed the absolute maximum rated voltage (2.5 V) of the IC:

- 1) Electromagnetic induction noise from the utility power supply in the reflow process during board-mounting, rework process after board-mounting, or individual characteristic evaluation (experimental confirmation), and
- 2) Electromagnetic induction noise from the tip of a soldering iron

Especially when using a soldering iron, make sure that the IC GND and soldering iron GND are at the same potential before soldering.

THIS PAGE IS BLANK.

S1C33301 Technical Manual  
**II CORE BLOCK**



## II-1 INTRODUCTION

The core block consists of a functional block C33\_CORE including CPU, BCU (Bus Control Unit), ITC (Interrupt Controller), CLG (Clock Generator) and DBG (Debug Unit), an I/O pad block for external interface, and an SBUS (Internal Silicon Integration Bus) for interfacing with on-chip Peripheral Macro Cells.

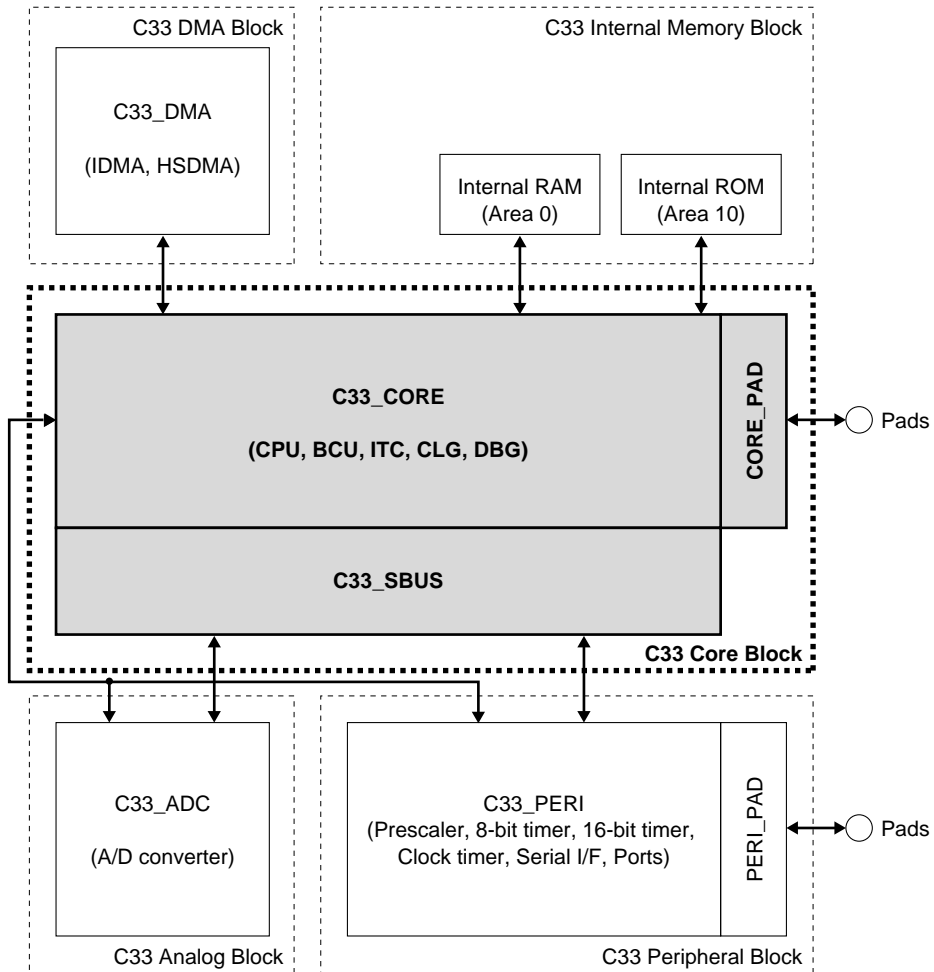


Figure II.1.1 Core Block

THIS PAGE IS BLANK.



## II-2 CPU AND OPERATING MODE

### ***CPU***

---

The C33 Core Block employs the C33 STD 32-bit RISC type CPU as the core CPU. Since it has a built-in multiplier, all instructions (105 instructions) in the C33 STD instruction set including the MAC (multiplication and accumulation) instruction and the multiplication/division instructions are available.

All the internal registers of the C33 STD can be used. The CPU registers and CPU address bus can handle 28-bit addresses.

Refer to the "S1C33000 Core CPU Manual" for details of the C33 STD.

### ***Standby Mode***

---

The CPU supports three standby modes: two HALT modes and a SLEEP mode.

By setting the CPU in the standby mode, power consumption can greatly be reduced.

#### ***HALT Mode***

When the CPU executes the halt instruction, it suspends the program execution and enters the HALT mode.

The CPU supports two types of HALT modes (basic HALT mode and HALT2 mode) and either can be selected using the HLT2OP (D3) / Clock option register (0x40190).

The CPU stops operating in basic HALT mode, so the amount of current consumption can be reduced. The internal peripheral circuits maintain the status (stop/run) before entering HALT mode.

HALT2 mode stops the external bus control functions including DMA and the bus clock as well as the CPU similar to basic HALT mode. Consequently, HALT2 mode realizes more power saving than the basic HALT mode.

The HALT mode is canceled by an initial reset or an interrupt including NMI. This mode is useful for saving power when waiting for an external input or completion of the peripheral circuit operations that do not need to execute the CPU.

The CPU transits to program execution status through trap processing when the HALT mode is canceled by an interrupt and executes the interrupt processing routine. The trap processing of the CPU saves the address of the instruction that follows the executed halt instruction into the stack. Therefore, when the interrupt processing routine is terminated by the reti instruction, the program flow returns to the instruction that follows the halt instruction. Note that the HALT mode cannot be canceled with an interrupt factor except for reset and NMI if the PSR is set into interrupt disabled status.

#### ***SLEEP Mode***

When the CPU executes the slp instruction, it suspends the program execution and enters SLEEP mode.

In SLEEP mode, the CPU and the internal peripheral circuits including the high-speed (OSC3) oscillation circuit stop operating. Thus SLEEP mode can greatly reduce current consumption in comparison to HALT mode. Moreover, the low-speed (OSC1) oscillation circuit and clock timer do not stop operating. The clock function keeps operating in SLEEP mode.

SLEEP mode is canceled by an initial reset or an interrupt (NMI, clock timer interrupt, external interrupt such as a key entry). Note that other interrupts by the internal peripheral circuits that use the OSC3 clock cannot be used for canceling SLEEP mode.

The CPU transits to program execution status through trap processing when the SLEEP mode is canceled by an interrupt and executes the interrupt processing routine. The trap processing of the CPU saves the address of the instruction that follows the executed slp instruction into the stack. Therefore, when the interrupt processing routine is terminated by the reti instruction, the program flow returns to the instruction that follows the slp instruction. Note that SLEEP mode cannot be canceled with an interrupt factor except for reset and NMI if the PSR is set into interrupt disabled status.

## Notes on Standby Mode

### Interrupts

The standby mode can be canceled by an interrupt. Therefore, it is necessary to enable the interrupt to be used for canceling the standby mode before setting the CPU in the standby mode. It is also necessary to set the IE (interrupt enable) and IL (interrupt level) bits in the PSR to a condition that can accept the interrupt. Otherwise, the standby mode cannot be canceled even when an interrupt occurs. Refer to "ITC (Interrupt Controller)", for interrupt settings.

### Oscillation circuit

The high-speed (OSC3) oscillation circuit stops in SLEEP mode and restarts oscillating when SLEEP mode is canceled. If the CPU had operated with the OSC3 clock before entering SLEEP mode, the CPU restarts operating with the OSC3 clock immediately after canceling SLEEP mode. However, the OSC3 oscillation needs appropriate stabilization time. When using the PLL, a lockup time is needed after the OSC3 oscillation is stabilized. To restart the CPU after the oscillation stabilizes, a programmable interval can be inserted between cancellation of SLEEP mode and starting the CPU operation. Refer to "CLG (Clock Generator)", for details.

The oscillation start time of the high-speed (OSC3) oscillation circuit varies according to the components to be used, board pattern and operating environment. The interval must be set to allow enough margin.

### BCU

When the CPU enters the standby mode, the BCU (bus control unit) stops after the current bus cycle has completed. All the chip enable signals are negated.

In basic HALT mode, the BCLK (bus clock) signal can be output and DMA also operates.

In HALT2 or SLEEP mode, the BCLK signal stops and DMA stops.

### Additional

The contents of the CPU registers and input/output port status are retained in the standby mode. Almost all control and data registers of the internal peripheral circuits are also retained, note, however, some registers may be changed at the transition to SLEEP mode. Refer to the section of each peripheral circuit for other precautions.

## Debug Mode

---

The C33 Core Block supports the debug mode.

The debug mode is a CPU function, and realizes single step operation and break functions in the chip itself. Refer to the "S1C33000 Core CPU Manual" for details of the debug mode and the functions.

Area 2 in the memory map can only be accessed in the debug mode.

In the debug mode, the OSC3 clock is used as the CPU operating clock. Therefore, do not stop the high-speed (OSC3) oscillation circuit when using the debugging functions. Furthermore, only the CPU and BCU operate in the debug mode, and other internal peripheral circuits (except the oscillation circuit) stop operating.

## Trap Table

---

Refer to Table II.5.1 in Section II-5, "ITC (Interrupt Controller)", for the trap table in the C33 Core and the "S1C33000 Core CPU Manual" for details of exceptions.

## II-3 INITIAL RESET

### Pins for Initial Reset

Table II.3.1 shows the pins used for initial reset.

Table II.3.1 Pins for Initial Reset

Pin name	I/O	Function
#RESET	I	Initial reset input pin (Low active) Low: Resets the CPU.
#NMI	I	NMI request input pin This pin is also used for selecting a reset method. High: Cold start Low: Hot start

The chip is reset when the #RESET pin goes low and starts operating at the rising edge of the reset signal. The CPU and internal peripheral circuits are initialized while the #RESET pin is low.

### Cold Start and Hot Start

The CPU supports two initial reset methods: cold start and hot start. The #NMI pin is used with the #RESET pin to set this condition.

The differences between cold start and hot start are shown in Table II.3.2.

Table II.3.2 Differences between Cold Start and Hot Start

Setup contents	Cold start	Hot start
Reset condition	#RESET = low & #NMI = high	#RESET = low & #NMI = low
CPU: PC	The vector at the boot address is loaded to the PC.	
CPU: PSR	All the PSR bits are reset to 0.	
CPU: Other registers	Undefined	
CPU: Operating clock	The CPU operates with the OSC3 clock.	
External bus status (0x48120–0x4813F)	Initialized	Status is retained.
Oscillation circuit	Both the OSC1 and OSC3 circuits start oscillating.	
I/O pin status (0x402C0–0x402DF)	Initialized	Status is retained.
Other peripheral circuit	Initialized or undefined	

Since cold start initializes all the internal peripheral circuits as well as the CPU, it is useful as a power-on reset. Hot start initializes the CPU and peripheral circuits, but does not reset the bus control unit and the input, output and I/O port status. It is therefore useful as a reset that maintains the external bus and I/O pin status during operation.

The #NMI pin that specifies the reset method should be set following the timing chart shown in Figure II.3.1.

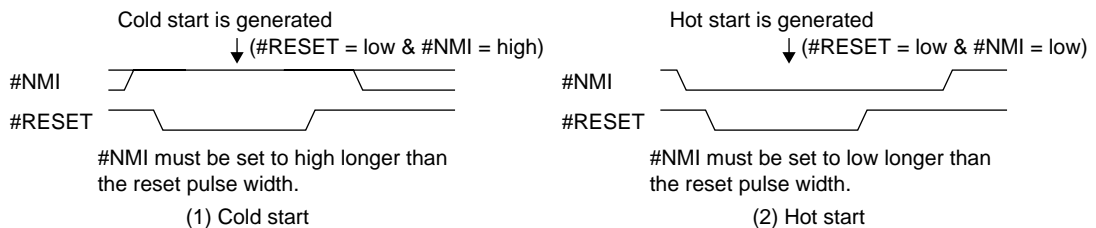


Figure II.3.1 Setup of #RESET and #NMI Pins

## Power-on Reset

Be sure to reset (cold start) the chip after turning on the power to start operating.

Since the #RESET pin is directly connected to an input gate, a power-on reset circuit should be configured outside the chip.

An initial reset (#RESET = low) turns the high-speed (OSC3) oscillation circuit on. The CPU starts operating with the OSC3 clock at the rising edge of the reset signal. The high-speed (OSC3) oscillation circuit needs an oscillation stabilization time. Initial reset must be released after an appropriate oscillation-stabilization time and a PLL lockup time has passed in order to start up the CPU without fault.

**Note:** The OSC3 oscillation start time varies due to the elements used, board pattern and operating environment, therefore allow enough margin for the reset-release time. Refer to "Oscillation Characteristics", in which an example of oscillation start time is provided.

### Power-on sequence

In order to operate the device normally, supply power in accordance with the following timing.

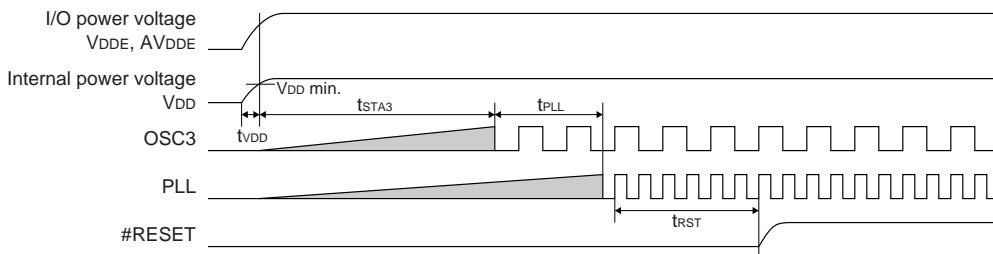


Figure II.3.2 Power-On Sequence

- (1) tvDD: Elapsed time until the power supply stabilizes after power-on  
Supply power in the following sequence (or simultaneously).  
Power-on: Internal core power voltage (VDD level voltages) → I/O power voltage (VDDE, AVDDDE, etc.) → Apply the input signal
- (2) tSTA3: Time at which OSC3 oscillation starts
- (3) tPLL: Time at which PLL locks up
- (4) trST: Minimum reset pulse width  
Time at which the clock supplied to the C33 STD core CPU stabilizes plus at least six clocks; Keep the #RESET signal low.

## Reset Pulse

A low pulse can be input to the #RESET pin for resetting the C33 STD CPU being operated.

The minimum reset pulse width is provided in "Electrical Characteristics". Be sure to input a pulse that has a pulse width longer than the minimum value.

To reset the chip when the high-speed (OSC3) oscillation circuit is in off status, the pulse width must be extended until the clock stabilizes similarly to the power-on reset. Be aware that a short reset pulse may cause an operation error.

## Boot Address

When the #RESET pin goes high after a reset period, the core CPU reads the reset vector (program start address) from the boot address (0x0C00000) and loads the vector to the PC (program counter). Then the CPU starts executing the program from the address.

The trap table in which trap vectors for interrupts and other trap factors are written also begins from the boot address by the default setting. (Refer to the "S1C33000 Core CPU Manual" for details of the trap table.)

The trap table base address can also be changed to a 1KB boundary address using the TTBR register (0x48134 to 0x48137).

## Notes Related to Initial Reset

---

### Core CPU

Since the all registers except for the PC and PSR are indeterminate at initial reset, they should be initialized by a program. In particular, the SP (stack pointer) must be initialized before accessing the stack area. NMI requests are disabled until any value is written to the SP. The initialization is necessary when the CPU is cold-started.

### Internal RAM

The contents of the internal RAM are indeterminate at initial reset. Initialize the area to be used if necessary.

### High-speed (OSC3) oscillation circuit

An initial reset activates the high-speed (OSC3) oscillation circuit and the CPU starts operating with the OSC3 clock after the initial reset is released. In order to prevent a malfunction of the CPU due to an unstabilized clock, the #RESET pin must be maintained at low until the OSC3 oscillation stabilizes when performing a power-on reset or resetting while the high-speed (OSC3) oscillation circuit is stopped.

### Low-speed (OSC1) oscillation circuit

A power-on reset or an initial reset when the low-speed (OSC1) oscillation circuit is off starts the OSC1 oscillation. The low-speed (OSC1) oscillation circuit takes a longer stabilization time (3 sec max. under the standard condition) than the high-speed (OSC3) oscillation circuit. In order to prevent a malfunction due to an unstabilized clock, do not use the OSC1 clock until the stabilization time has passed.

### BCU (Bus Control Unit)

Cold-start initializes the control registers for the BCU (bus control unit). Therefore, it is necessary to set up all the bus conditions.

Hot-start retains the previous bus conditions before an initial reset.

### Input/output ports and input/output pins

Cold start initializes the control and data registers for the input and I/O ports.

Hot start retains the contents of the control registers and input/output pin status before an initial reset. However, when the pins are used for the internal peripheral circuits, it is necessary to set up the control registers of the peripheral circuit because they are initialized by an initial reset.

### Other internal peripheral circuits

The control and data registers of peripheral circuits other than those listed above are initialized with the pre-defined values or become indeterminate regardless of the reset method (cold start or hot start). Therefore, it is necessary to set up the peripheral circuit conditions.

Refer to the I/O maps or explanation of each peripheral circuit section for initial settings of the peripheral circuits.

THIS PAGE IS BLANK.

## II-4 BCU (Bus Control Unit)

The BCU (Bus Control Unit) provides an interface for external devices and on-chip user logic block. The types and sizes of memory and peripheral I/O devices can be set for each area of the memory map and can be controlled directly by the BCU. This unit also supports a direct interface for burst ROM. This chapter describes how to control the external and internal system interface, and how it operates.

**Note:** The control registers of the external system interface shown in this chapter are mapped to the internal 16-bit I/O area. Therefore, the addresses of these control registers are indicated by half-word (16-bit) addresses unless otherwise specified. Note that the control registers can be accessed in bytes, half-words, or words.

### Pin Assignment for External System Interface

#### I/O Pin List

##### External I/O pins

Table II.4.1 lists the pins used for the external system interface.

Table II.4.1 I/O Pin List

Pin name	I/O	Function
A[0](#BSL)	O	Address bus (A0) / Bus strobe (Low-byte)
A[27:1]	O	Address bus (A1–A27)
D[15:0]	I/O	Data bus (D0–D15)
#CE10EX(#CE9&10EX)	O	Area 10/(9&10) external memory chip enable
#CE9(#CE17/#CE17&18)	O	Area 9/17/(17&18) chip enable
#CE8(#RAS1/#CE14/#RAS3)	O	Area 8/14 chip enable / DRAM Row strobe
#CE7(#RAS0/#CE13/#RAS2)	O	Area 7/13 chip enable / DRAM Row strobe
#CE6(#CE7&8)	O	Area 6/(7&8) chip enable
#CE5(#CE15/#CE15&16)	O	Area 5/15/(15&16) chip enable
#CE4(#CE11/#CE11&12)	O	Area 4/11/(11&12) chip enable
#RD	O	Read signal
#WRL(#WR/#WE)	O	Write (Low-byte) / Write / DRAM write
#WRH(#BSH)	O	Write (High-byte) / Bus strobe (High-byte)
BCLK	O	Bus clock output
P35(#BUSACK)	I/O	I/O port / Bus request acknowledge
P34(#BUSREQ/#CE6)	I/O	I/O port / Bus release request / Area 6 chip enable
P31(#BUSGET/#GARD)	I/O	I/O port / Bus status monitor signal output / Area read signal output for GA
P30(#WAIT/#CE4&5)	I/O	I/O port / Wait cycle request / Areas 4&5 chip enable
P21(#DWE/#GAAS)	I/O	I/O port / DRAM write (Low-byte) / Area address strobe output for GA
#X2SPD	I	CPU - BCLK clock ratio 1: CPU clock = Bus clock, 0: CPU clock = Bus clock x 2
EA10MD[1:0]	I	Area 10 boot mode selection 11: External ROM, 10: Internal ROM

- Notes:**
- The input/output pins used for the external bus interface are shared with I/O ports except for some signals. When using the pins as external bus signals, set up the port function select registers according to the system configuration.
  - Some external bus I/O pins may be shared with extended peripheral functions other than the C33 standard block or may not be provided by model. Refer to the "Pin Description" chapter for the pin configuration.

## Combination of System Bus Control Signals

The bus control signal pins that have two or more functions have their functionality determined when an interface method is selected by a program. The BCU supports two interface methods.

Table II.4.2 Interface Selection

Interface method	Control bit
A0 system (default)	SBUSST (D3) / Bus control register (0x4812E) = "0"
#BSL system	SBUSST (D3) / Bus control register (0x4812E) = "1"

SBUSST is initialized to "0" at cold start.

When the IC is hot-started, these bits retain their status before the chip was reset.

Table II.4.3 shows combinations of control signals classified by each interface method.

Table II.4.3 Combinations of Bus Control Signals

A0 system	#BSL system
A0	#BSL (little endian) / #BSH (big endian) *
#WRL	#WR
#WRH	#BSH (little endian) / #BSL (big endian) *
#CE <sub>x</sub>	#CE <sub>x</sub>

\* In the #BSL system, the A0 and #WRH pin functions change according to the endian selected (little endian or big endian).



## Memory Area

### Memory Map

Figure II.4.1 shows the memory map supported by the BCU.

Area	Address		Area	Address	
Area 9 SRAM type Burst ROM type 8 or 16 bits	0x0BFFFFFF 0x08000000	External memory (4MB)	Area 18 SRAM type 8 or 16 bits	0xFFFFFFFF 0xC0000000	External memory (64MB)
Area 8 SRAM type 8 or 16 bits	0x07FFFFFF 0x06000000		Area 17 SRAM type 8 or 16 bits	0xBFFFFFFF 0x80000000	
Area 7 SRAM type 8 or 16 bits	0x05FFFFFF 0x04000000	External memory (2MB)	Area 16 SRAM type 8 or 16 bits	0x7FFFFFFF 0x60000000	External memory (32MB)
Area 6 SRAM type	0x03FFFFFF 0x03800000	External I/O (16-bit device)	Area 15 SRAM type 8 or 16 bits	0x5FFFFFFF 0x40000000	External memory (32MB)
	0x037FFFFFFF 0x03000000	External I/O (8-bit device)		Area 14 SRAM type 8 or 16 bits	
Area 5 SRAM type 8 or 16 bits	0x02FFFFFF 0x02000000	External memory (1MB)	Area 13 SRAM type 8 or 16 bits	0x2FFFFFFF 0x20000000	External memory (16MB)
Area 4 SRAM type 8 or 16 bits	0x01FFFFFF 0x01000000	External memory (1MB)	Area 12 SRAM type 8 or 16 bits	0x1FFFFFFF 0x18000000	External memory (8MB)
Area 3 16 bits Fixed at 1 cycle	0x00FFFFFF 0x00800000	(Reserved)	Area 11 SRAM type 8 or 16 bits	0x17FFFFFF 0x10000000	External memory (8MB)
Area 2 16 bits Fixed at 3 cycles	0x007FFFFF 0x00600000	(Reserved) For CPU core or debug mode	Area 10 SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFFF 0x0C000000	External memory (4MB)
Area 1 8, 16 bits 2 or 4 cycles	0x005FFFFF 0x00500000	(Mirror of internal I/O)			
	0x004FFFFF 0x00400000	Internal I/O			
	0x003FFFFF 0x00300000	(Mirror of internal I/O)			
Area 0 32 bits Fixed at 1 cycle	0x002FFFFF 0x00000000	Internal RAM			

Figure II.4.1 Memory Map

Basically, Areas 0 to 3 are internal memory areas and Areas 4 to 18 are external memory areas.

Area 0 is normally used for a built-in RAM. The built-in memory is mapped from the beginning of the area.

Area 1 is reserved for the I/O memory of the on-chip functional blocks. Address 0x0040000 to address 0x004FFFF are used as the control registers and address 0x0050000 to 0x005FFFF are used as the mirror area.

Area 2 is used in debug mode only and it cannot be accessed in user mode (normal program execution status).

**Note:** Areas 4 to 18 may be internal memory areas in which control registers for model-specific extended peripheral circuits or internal memory devices are allocated. Refer to the "Internal Memory" chapter for the memory map of each model.

### External Memory Map and Chip Enable

By default, the address space is divided into 19 areas (areas 0 to 18) for management purposes. Of these, areas 4 to 10 are open to an external system, each provided with an independent chip-enable pin (#CE[10:4]).

The C33 Core Block is limited to 7 pins for the #CE output due to its package structure. However, the #CE[4:10] output pins can be switched to the high-order area chip enable output pins as shown in Table II.4.4 using software. CEFUNC[1:0] (D[A:9]) / DRAM timing set-up register (0x48130) is used for this switching.

Table II.4.4 Switching of #CE Output

Pin	CEFUNC = "00"	CEFUNC = "01"	CEFUNC = "1x"
#CE4	#CE4	#CE11	#CE11+#CE12
#CE5	#CE5	#CE15	#CE15+#CE16
#CE6	#CE6	#CE6	#CE7+#CE8
#CE7/#RAS0	#CE7/#RAS0	#CE13/#RAS2	#CE13/#RAS2
#CE8/#RAS1	#CE8/#RAS1	#CE14/#RAS3	#CE14/#RAS3
#CE9	#CE9	#CE17	#CE17+#CE18
#CE10EX	#CE10EX	#CE10EX	#CE9+#CE10EX

(Default: CEFUNC = "00")

The high-order areas that are made available for use by writing "01" to CEFUNC can be larger in size than the default low-order areas. For example, when using in default settings, the available space is 4MB in areas 7 and 8. However, if areas 13 and 14 are used, up to 32MB can be used. The same applies to the other areas.

Furthermore, when CEFUNC is set to "10" or "11", five chip enable signals are expanded into two area size.

Figure II.4.2 shows a memory map for an external system.

**Note:** The C33 Core Block features 28-bit internal address processing, note, however, that the number of output pins for the address bus varies by model. The accessible area is limited within the range according to the address bus output pin count.

Area	Address	Area	Address
Area 10 (#CE10) SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFFF 0x0C000000	Area 17 (#CE17) SRAM type 8 or 16 bits	0xBFFFFFFF 0x80000000
Area 9 (#CE9) SRAM type Burst ROM type 8 or 16 bits	0x0BFFFFFFF 0x08000000	Area 15 (#CE15) SRAM type 8 or 16 bits	0x5FFFFFFF 0x40000000
Area 8 (#CE8/#RAS1) SRAM type 8 or 16 bits	0x07FFFFFFF 0x06000000	Area 14 (#CE14/#RAS3) SRAM type 8 or 16 bits	0x3FFFFFFF 0x30000000
Area 7 (#CE7/#RAS0) SRAM type 8 or 16 bits	0x05FFFFFFF 0x04000000	Area 13 (#CE13/#RAS2) SRAM type 8 or 16 bits	0x2FFFFFFF 0x20000000
Area 6 (#CE6) SRAM type	0x03FFFFFFF 0x03800000 0x037FFFFF 0x03000000	Area 11 (#CE11) SRAM type 8 or 16 bits	0x17FFFFFFF 0x10000000
Area 5 (#CE5) SRAM type 8 or 16 bits	0x02FFFFFFF 0x02000000	Area 10 (#CE10) SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFFF 0x0C000000
Area 4 (#CE4) SRAM type 8 or 16 bits	0x01FFFFFFF 0x01000000	Area 6 (#CE6) SRAM type	0x03FFFFFFF 0x03800000 0x037FFFFF 0x03000000

CEFUNC = "00"

CEFUNC = "01"

Area	Address	
Area 17–18 (#CE17+18) SRAM type 8 or 16 bits	0xFFFFFFFF  0x8000000	External memory 7 (128MB)
Areas 15–16 (#CE15+16) SRAM type 8 or 16 bits	0x7FFFFFFF  0x4000000	
Area 14 (#CE14/#RAS3) SRAM type 8 or 16 bits	0x3FFFFFFF  0x3000000	External memory 5 (16MB)
Area 13 (#CE13/#RAS2) SRAM type 8 or 16 bits	0x2FFFFFFF  0x2000000	External memory 4 (16MB)
Areas 11–12 (#CE11+12) SRAM type 8 or 16 bits	0x1FFFFFFF  0x1000000	External memory 3 (16MB)
Areas 9–10 (#CE9+10EX) SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFFF  0x0800000	External memory 2 (8MB)
Areas 7–8 (#CE7+8) SRAM type 8 or 16 bits	0x07FFFFFFF  0x0400000	External memory 1 (4MB)

CEFUNC = "10" or "11"

Figure II.4.2 External System Memory Map

Furthermore, the #CE4+#CE5 and #CE6 signals can be output from the P30 and P34 terminals, respectively. This function expands the accessible area when CEFUNC is set to "01", "10" or "11".

To output the #CE4+#CE5 signal from the P30 terminal:

CFP30 (D0)/P3 function select register (0x402DC) = "1"

IOC30 (D0)/P3 I/O control register (0x402DE) = "1"

To output the #CE6 signal from the P34 terminal:

CFP34 (D4)/P3 function select register (0x402DC) = "1"

IOC34 (D4)/P3 I/O control register (0x402DE) = "1"

The P30 and P34 terminals are set for the general I/O ports at initial reset.

The P30 and P34 terminals are shared with the #WAIT input and the #BUSREQ input, respectively. Therefore, when using the #WAIT and #BUSREQ signals, these terminals cannot be used for #CE4+#CE5 and #CE6 outputs.

## Using Internal Memory on External Memory Area

The BCU allows using of an internal memory in the external memory areas.

The AxxIO bit in the access control register (0x48132) is used to select either internal access or external access. When "1" is written, the internal device will be accessed and when "0" is written, the external device is accessed (external access by default). The bit names and the corresponding areas are as follows:

- A18IO (DF): Areas 17 and 18
- A16IO (DE): Areas 15 and 16
- A14IO (DD): Areas 13 and 14
- A12IO (DC): Areas 11 and 12
- A8IO (DA): Areas 7 and 8
- A6IO (D9): Area 6
- A5IO (D8): Areas 4 and 5

## Exclusive Signals for Areas

Areas can be accessed using the exclusive signals (address strobe and read signals) as well as the common control signals.

To use these exclusive signals, they should be configured using G/A read signal control register (0x48138).

The AxxAS bit is used to enable/disable the address strobe signal, and the AxxRD bit is used to enable/disable the read signal. When "1" is written to the bit, the exclusive signal for the corresponding area(s) is enabled and when "0" is written, it is disabled (disabled by default). The bit names and the corresponding areas are as follows:

- A18AS (DF), A18RD (D7): Areas 17 and 18
- A16AS (DE), A16RD (D6): Areas 15 and 16
- A14AS (DD), A14RD (D5): Areas 13 and 14
- A12AS (DC), A12RD (D4): Areas 11 and 12
- A8AS (DA), A8RD (D2): Areas 7 and 8
- A6AS (D9), A6RD (D1): Area 6
- A5AS (D8), A5RD (D0): Areas 4 and 5

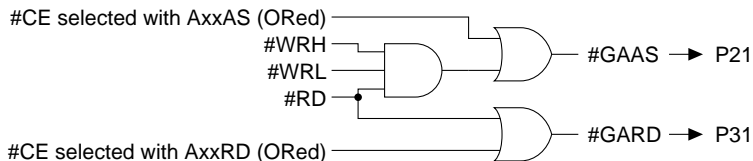


Figure II.4.3 #GAAS and #GARD Signals

The address strobe signal and the read signal are output from the P21 pin and P31 pin, respectively. Therefore, when using these signals, the pin(s) must be configured for exclusive signal output using the port function select register and port function extension register.

To output the exclusive address strobe signal #GAAS:

CFEX2 (D2)/Port function extension register (0x402DF) = "1"

To output the exclusive address strobe signal #GARD:

CFEX3 (D3)/Port function extension register (0x402DF) = "1"

These signals are common used to all the above areas, so when two or more areas are selected to output the exclusive signal, OR condition is applied.

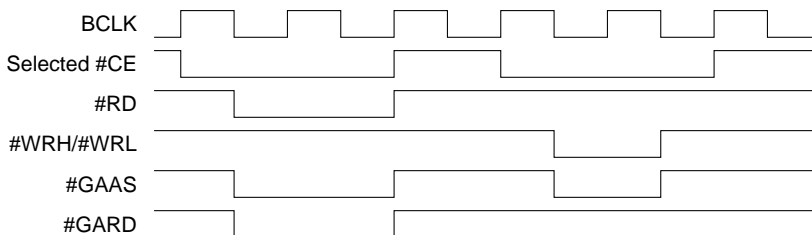


Figure II.4.4 Output Timing of #GAAS and #GARD Signals

## Area 10

Area 10 is an external memory area that includes the boot address (0xC00000). This area supports two boot modes.

**Note:** Refer to the "Internal Memory" chapter for whether ROM exists in Area 10 or not.

### Area 10 boot mode

The boot mode can be configured using the external pins EA10MD[1:0].

Table II.4.5 Area 10 Boot Mode Selection

EA10MD[1:0] pins	Area 10 boot mode
10	Internal ROM boot mode
11	External ROM boot mode

#### Internal ROM boot mode

The CPU boots by the internal ROM mapped to area 10. The internal ROM size should be selected from among eight types (min. 16 KB, max. 2 MB) using the A10IR[2:0] (D[E:C])/Areas 10–9 set-up register (0x48126). This ROM begins with address 0xC00000 and can be read in one cycle. For the remained area within area 10, the external memory will be accessed if it is available.

#### External ROM boot mode

The CPU boots by the external ROM (ROM, Flash, SRAM, etc.). This mode uses the bus condition set by the BCU registers for area 10.

### Setting the internal ROM size

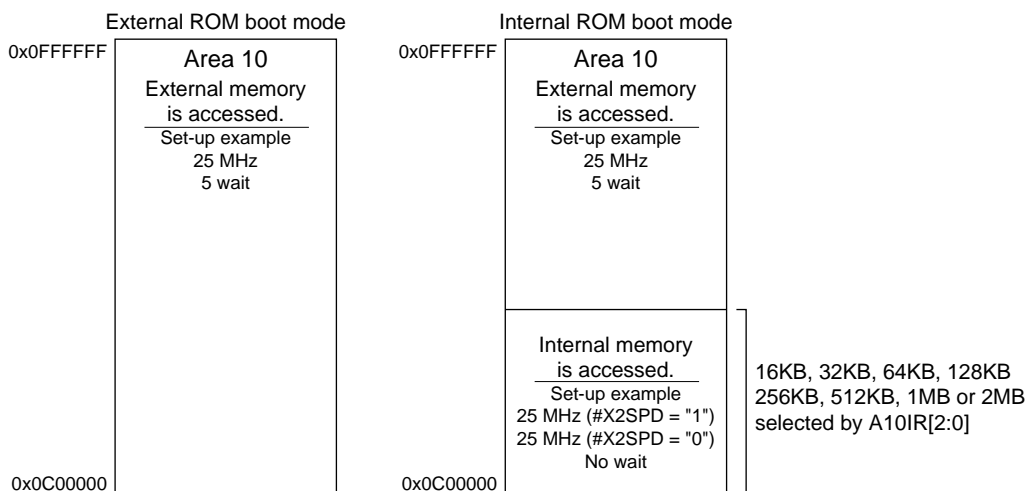
When the internal ROM boot mode is used, the internal ROM size should be set using A10IR[2:0] (D[E:C])/Areas 10–9 set-up register (0x48126).

Table II.4.6 Area 10 Internal ROM Size

A10IR2	A10IR1	A10IR0	ROM size
0	0	0	16 KB
0	0	1	32 KB
0	1	0	64 KB
0	1	1	128 KB
1	0	0	256 KB
1	0	1	512 KB
1	1	0	1 MB
1	1	1	2 MB (default)

### Area 10 memory map

Figure II.4.5 shows the memory map of area 10.



## Setting External Bus Conditions

The type, size, and wait conditions of a device connected to the external bus can be individually set for each area using the control register (0x48120 to 0x48130). The following explains the available setup conditions individually for each area.

The control register used to set bus conditions is initialized at cold start. Therefore, please set up these registers again using software according to the external device configuration and specifications.

When the IC is hot-started, the setup contents and pins retain their previous status before a reset.

## Setting Device Type and Size

Table II.4.7 shows the types of devices that can be connected directly to each area.

Table II.4.7 Device Type

Area	SRAM type	Burst ROM type	Control bit
18–15	○	×	–
14	○	×	–
13	○	×	–
12,11	○	×	–
10	○	○	A10DRA (D8)/Areas 10–9 set-up register(0x48126)
9	○	○	A9DRA (D7)/Areas 10–9 set-up register(0x48126)
8	○	×	–
7	○	×	–
6–4	○	×	–

○: Can be connected    ×: Cannot be connected

When connecting burst ROM, write "1" to each corresponding control bit. These control bits are reset to "0" (SRAM type) at cold start.

The device size can be set to 8 or 16 bits once every two areas except for area 6. Area 6 alone has its first half (0x300000–0x37FFFF) fixed to an 8-bit device and the second half (0x380000–0x3FFFFFF) fixed to a 16-bit device.

Table II.4.8 Device Size Control Bits

Area	Control bit
18, 17	A18SZ(DE)/Areas 18–15 set-up register(0x48120)
16, 15	A16SZ(D6)/Areas 18–15 set-up register(0x48120)
14, 13	A14SZ(D6)/Areas 14–13 set-up register(0x48122)
12, 11	A12SZ(D6)/Areas 12–11 set-up register(0x48124)
10, 9	A10SZ(D6)/Areas 10–9 set-up register(0x48126)
8, 7	A8SZ(D6)/Areas 8–7 set-up register(0x48128)
5, 4	A5SZ(D6)/Areas 6–4 set-up register(0x4812A)

At cold start, each area by default is set to 16 bits.

When using an 8-bit device, write "1" to the control bit.

**Note:** The BCU supports 16-bit burst ROM. Therefore, when connecting burst ROM to area 10 or area 9, do not set the device size to 8 bits (A10SZ = "1").

For differences in bus operation due to the device size and access data size, refer to "Bus Operation of External Memory".

## Setting SRAM Timing Conditions

The areas set for the SRAM allow wait cycles and output disable delay time to be set.

Number of wait cycles: 0 to 7 (incremented in units of one cycle)

Output disable delay time: 0.5, 1.5, 2.5, 3.5 cycles

This selection can be made once every two areas except for area 6.

Table II.4.9 Timing Condition Setting Bits (for SRAM type)

Area	Number of wait cycles	Output disable delay time	Control register
18, 17	A18WT[2:0](D[A:8])	A18DF[1:0](D[D:C])	Areas 18–15 set-up register(0x48120)
16, 15	A16WT[2:0](D[2:0])	A16DF[1:0](D[5:4])	Areas 18–15 set-up register(0x48120)
14, 13	A14WT[2:0](D[2:0])	A14DF[1:0](D[5:4])	Areas 14–13 set-up register(0x48122)
12, 11	A12WT[2:0](D[2:0])	A12DF[1:0](D[5:4])	Areas 12–11 set-up register(0x48124)
10, 9	A10WT[2:0](D[2:0])	A10DF[1:0](D[5:4])	Areas 10–9 set-up register(0x48126)
8, 7	A8WT[2:0](D[2:0])	A8DF[1:0](D[5:4])	Areas 8–7 set-up register(0x48128)
6	A6WT[2:0](D[A:8])	A6DF[1:0](D[D:C])	Areas 6–4 set-up register(0x4812A)
5, 4	A5WT[2:0](D[2:0])	A5DF[1:0](D[5:4])	Areas 6–4 set-up register(0x4812A)

At cold start, the number of wait cycles is set to 7 and the output disable delay time is set to 3.5 cycles. Reset up these parameters as necessary using software according to specifications of the connected device.

At hot start, these parameters retain their previous settings before a reset.

### Wait cycles

When the number of wait cycles is set for an area using the control bit, the BCU extends the bus cycle for a duration equivalent to the wait cycles set when it accesses the area. Set the desired wait cycles according to the bus clock frequency and the external device's access time. Separately from the wait cycles set here, a wait request from an external device can also be accepted using the #WAIT pin. Since the settings of wait cycles using software are made once every two areas, use this external wait request function if you want the wait cycles to be controlled individually in each area or if you need 7 or more wait cycles. The #WAIT pin is shared with the P30 I/O port. For an external wait request to be accepted, write "1" to CFP30 (D0) / P3 function select register (0x402DC [Byte]) and write "1" (default = "0") to SWAITE (D0) / Bus control register (0x4812E) to enable the #WAIT pin.

For timing charts for bus cycles and when wait cycles are inserted, refer to "Bus Cycles in External System Interface".

If the number of wait cycles is set to 0 and no external wait is requested, the basic read cycle (read in byte or half-word) for the SRAM external device consists of one cycle. If wait cycles are set, because these cycles are added, the bus read cycle consists of [number of wait cycles + 1] (providing that there is no external wait). On the other hand, the basic write cycle consists of at least two cycles. This does not change regardless of whether zero or one wait cycle is set. If the number of wait cycles set is 2 or more, the bus cycle is actually extended. In this case, the bus write cycle consists of [number of wait cycles + 1], as in the case of read cycles (providing that there is no external wait).

## Output disable delay time

In cases when a device having a long output disable time is connected, if a read cycle for that device is followed by the next access, contention for the data bus may occur. (Due to the fact the read device's data bus is not placed in the high-impedance state.) The output disable delay time is provided to prevent such data bus contention. This is accomplished by inserting a specified number of output disable delay cycles between a read cycle and the next bus operation. Care is required with the #CE<sub>x</sub> signals, however, since different areas may be asserted consecutively. The output disable delay time affects command signals such as #RD and #WRL/#WRH. Check the specifications of the device to be connected before setting the output disable delay time.

When a bus cycle begins, the #RD or #WRL/#WRH signal is asserted 0.5 cycles after the #CE<sub>x</sub> signal is asserted. This 0.5 cycle period is equivalent to an output disable delay time for the immediately preceding bus access. Therefore, the minimum output disable delay time is 0.5 cycles.

AxxDF[1:0] in the setup register can specify 0 to 3 cycles of output disable delay time in 1 cycle units, so the actual number of cycles to be inserted is 0.5 to 3.5 cycles.

The output disable delay time is inserted only in the following cases:

- when a read cycle from the external device that has had an output disable delay time set is followed by a write cycle performed by the CPU; and
- when a read cycle from the external device that has had an output disable delay time set is followed by a read cycle for a different area (including the internal device).

Conversely, no output disable delay time is inserted in the following conditions:

- immediately after a write cycle, and
- during a successive read from the same external device.

## Setting Timing Conditions of Burst ROM

### Wait cycles

If burst ROM is selected for area 10 or 9, the wait cycles to be inserted in the burst read cycle can be selected in a range from 0 to 3 cycles. A10BW[1:0] (D[A:9]) / Areas 10–9 set-up register (0x48126) is used for this selection. This selection is applied simultaneously to areas 10 and 9, so wait cycles can not be chosen individually for each area. The wait cycles set at cold start is 0.

Even for a burst read, the SRAM settings of wait cycles in the first bus operation are valid. (Refer to A10WT[2:0] in the foregoing section.)

The wait cycles set by A10BW[1:0] are inserted into the burst cycles after the first bus operation.

In addition, when burst ROM is selected, no wait cycles can be inserted into the read cycle via the #WAIT pin. For writing to an area that has had burst ROM selected, an SRAM write cycle is executed. In this case, both the SRAM settings of wait cycles and those input via the #WAIT pin are valid.

### Burst mode

The burst mode can be selected between an eight-consecutive-burst and a four-consecutive-burst mode. RBST8 (DD) / Bus control register (0x4812E) is used for this selection. The eight-consecutive-burst mode is selected by writing "1" to RBST8 and the four-consecutive-burst mode is selected by setting the bit to "0". At cold start, the four-consecutive-burst mode is set by default.



## Bus Operation

### Data Arrangement in Memory

The S1C33 Family of devices handle data in bytes (8 bits), half-words (16 bits), and words (32 bits). When accessing data in memory, it is necessary to specify a boundary address that conforms to the data size involved. Specification of an invalid address causes an address error exception. For instructions (e.g., stack manipulation or branch instructions) that rewrite the SP (stack pointer) or PC (program counter), the specified addresses are forcibly modified to appropriate boundary addresses. Therefore, no address error exception occurs in this type of instruction. For details about the address error exception, refer to the "S1C33000 Core CPU Manual".

Table II.4.10 shows the data arrangement in memory, classified by data type.

Table II.4.10 Data Arrangement in Memory

Data type	Arranged location
Byte data	Byte boundary address (all addresses)
Half-word data	Half-word boundary address (A[0]="0")
Word data	Word boundary address (A[1:0]="00")

The half-word and word data in memory area accessed in little-endian format by default. It can be changed to big-endian format using AxxEC (D[7:0])/Access control register (0x48132). When "1" is written to AxxEC, the corresponding area is accessed in big-endian method. The bit names and the corresponding areas are as follows:

A18EC (D7): Areas 17 and 18

A16EC (D6): Areas 15 and 16

A14EC (D5): Areas 13 and 14

A12EC (D4): Areas 11 and 12

A10EC (D3): Areas 9 and 10 ... Fixed at "0" (little-endian) for booting.

A8EC (D2): Areas 7 and 8

A6EC (D1): Area 6

A5EC (D0): Areas 4 and 5

To increase memory efficiency, try to locate the same type of data at continuous locations on exact boundary addresses in order to minimize invalid areas.

### Bus Operation of External Memory

The external data bus is 16-bits wide. For this reason, more than one bus operation occurs depending on the device size and the data size of the instruction executed, as shown in Table II.4.11.

Table II.4.11 Number of Bus Operation Cycles

Data size to be accessed	Device size	Number of bus operation cycles	Remarks
32 bits	16 bits	2	
16 bits	16 bits	1	
8 bits	16 bits	1	In little-endian method, the low-order byte is accessed when the LSB of the address (A[0]) is "0" or the #BSL signal is L. The high-order byte is accessed when the LSB of the address (A[0]) is "1" or the #BSH signal is H. In big-endian method, the high-order byte is accessed when the LSB of the address (A[0]) is "0" or the #BSL signal is L. The low-order byte is accessed when the LSB of the address (A[0]) is "1" or the #BSH signal is H.
32 bits	8 bits	4	In little-endian method, the 8-bit device must be connected to the low-order 8 bits of the data bus. In big-endian method, the 8-bit device must be connected to the high-order 8 bits of the data bus.
16 bits	8 bits	2	In little-endian method, the 8-bit device must be connected to the low-order 8 bits of the data bus. In big-endian method, the 8-bit device must be connected to the high-order 8 bits of the data bus.
8 bits	8 bits	1	In little-endian method, the 8-bit device must be connected to the low-order 8 bits of the data bus. In big-endian method, the 8-bit device must be connected to the high-order 8 bits of the data bus.

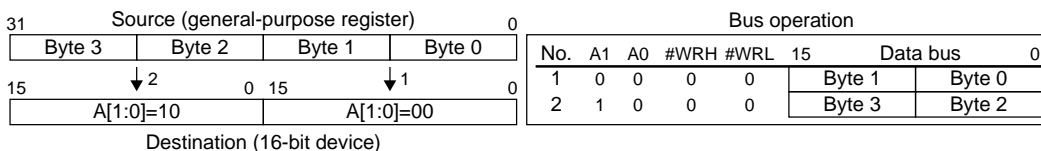
These bus operations are shown in the figure below, taking the example of the A0 method. With the BSL method, the following adjustments should be made when reading the figure.

- (1) For data reads, the operation is as shown in the figure below.
- (2) For little-endian data writes, read A0 as #BSC, and #WRH as #BSH.
- (3) For big-endian data writes, read A0 as #BSL, and #WRL as #BSH.

## II-4 CORE BLOCK: BCU (Bus Control Unit)

For information on memory connection, see Figure II.4.19.

Little-endian



Big-endian

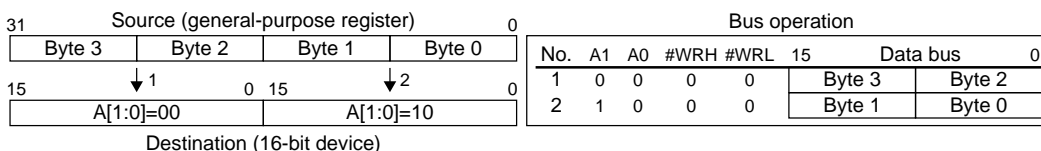
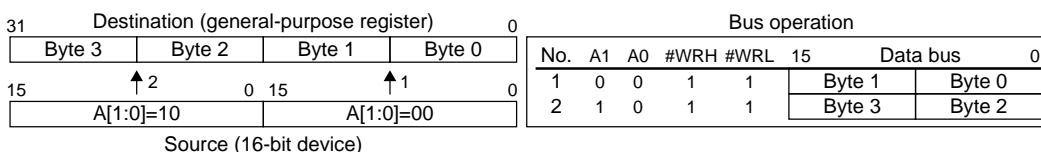


Figure II.4.6 Word Data Writing to a 16-bit Device

Little-endian



Big-endian

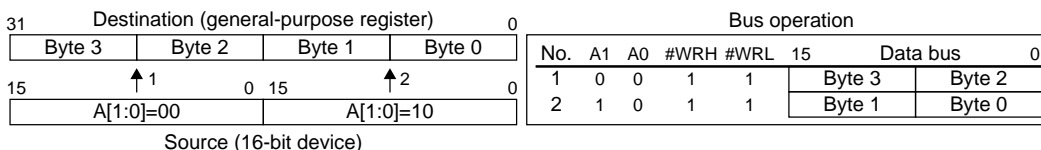
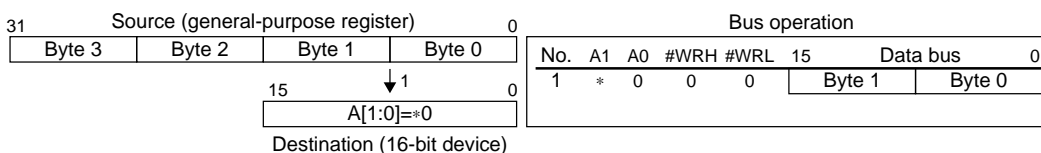


Figure II.4.7 Word Data Reading from a 16-bit Device

Little-endian



Big-endian

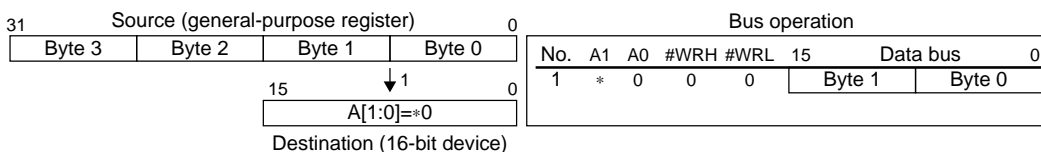
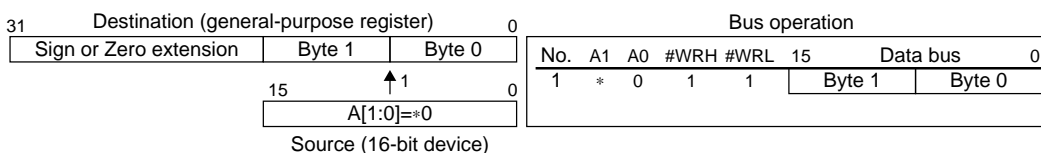


Figure II.4.8 Half-word Data Writing to a 16-bit Device

Little-endian



Big-endian

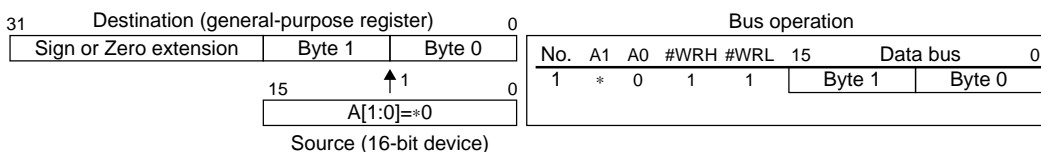
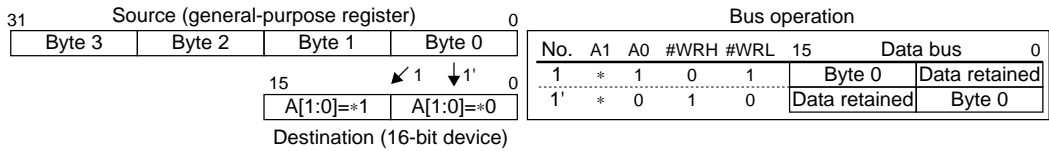


Figure II.4.9 Half-word Data Reading from a 16-bit Device

Little-endian



Big-endian

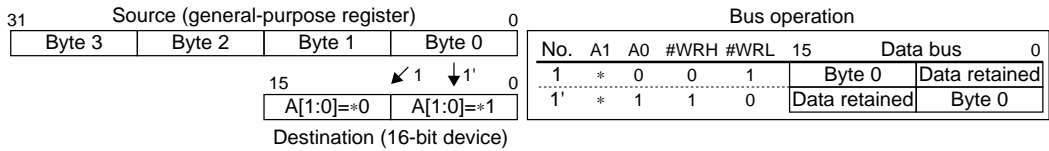
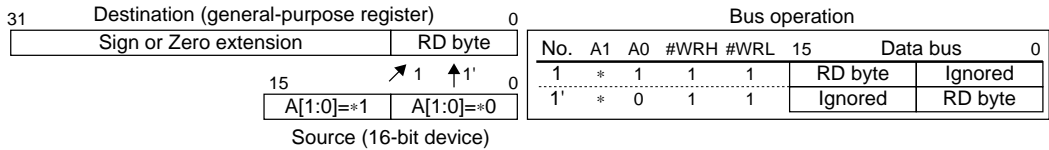


Figure II.4.10 Byte Data Writing to a 16-bit Device

Little-endian



Big-endian

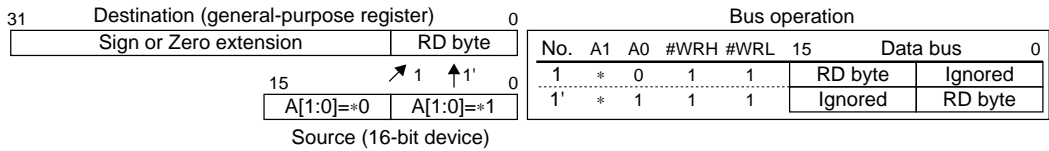
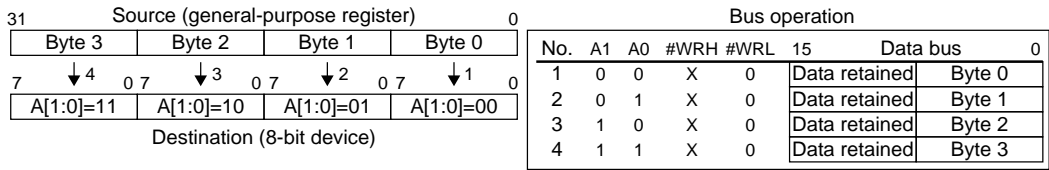


Figure II.4.11 Byte Data Reading from a 16-bit Device

Little-endian



(X: Not connected/Unused)

Big-endian

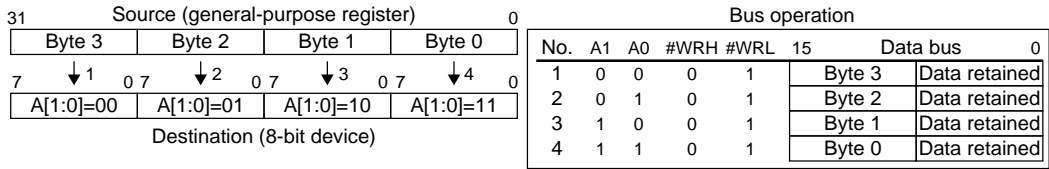
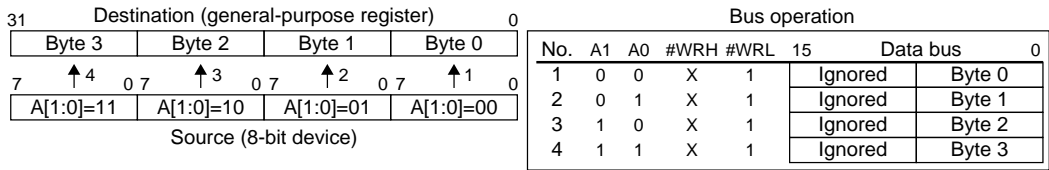


Figure II.4.12 Word Data Writing to an 8-bit Device

Little-endian



(X: Not connected/Unused)

Big-endian

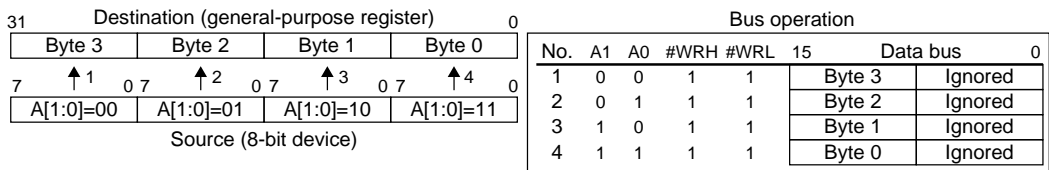
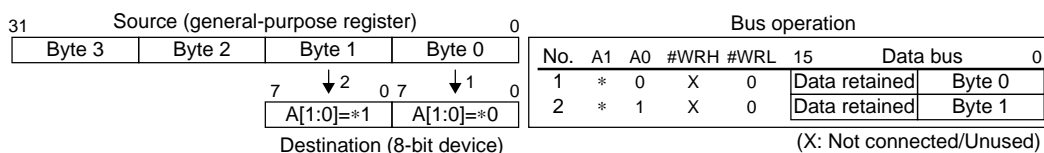


Figure II.4.13 Word Data Reading from an 8-bit Device

## II-4 CORE BLOCK: BCU (Bus Control Unit)

Little-endian



Big-endian

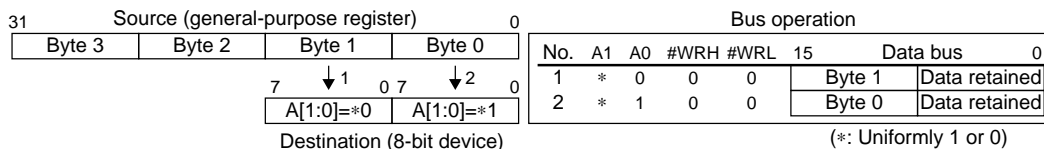
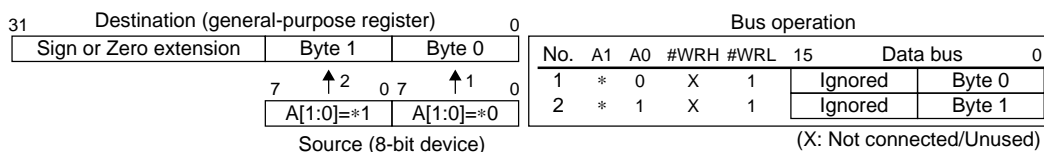


Figure II.4.14 Half-word Data Writing to an 8-bit Device

Little-endian



Big-endian

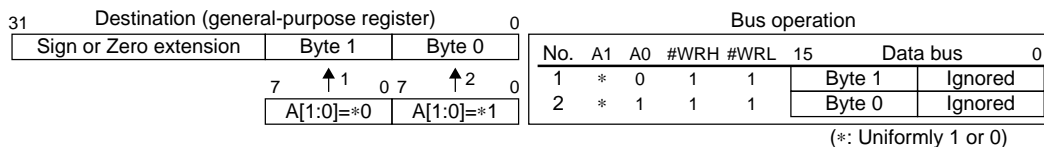
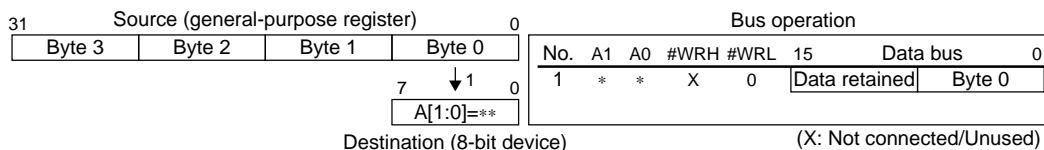


Figure II.4.15 Half-word Data Reading from an 8-bit Device

Little-endian



Big-endian

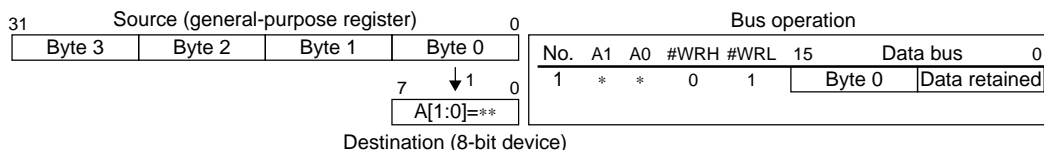
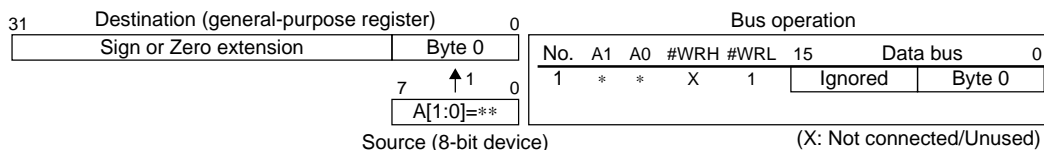


Figure II.4.16 Byte Data Writing to an 8-bit Device

Little-endian



Big-endian

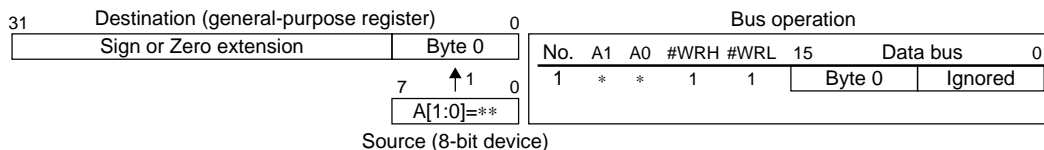
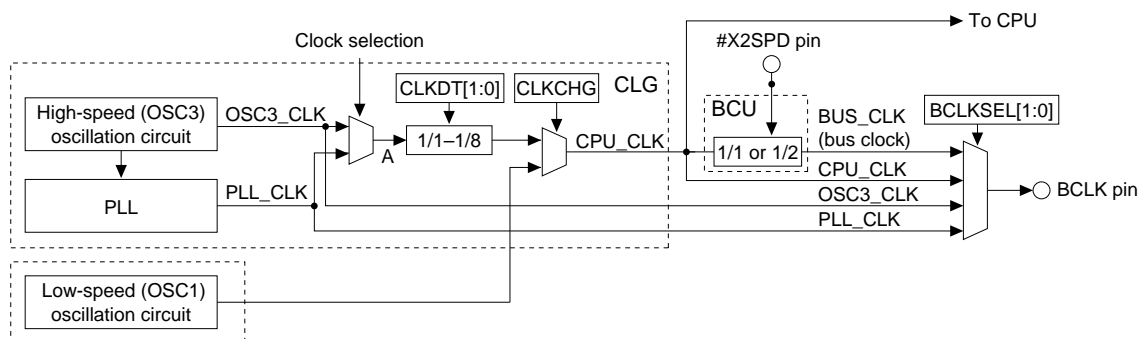


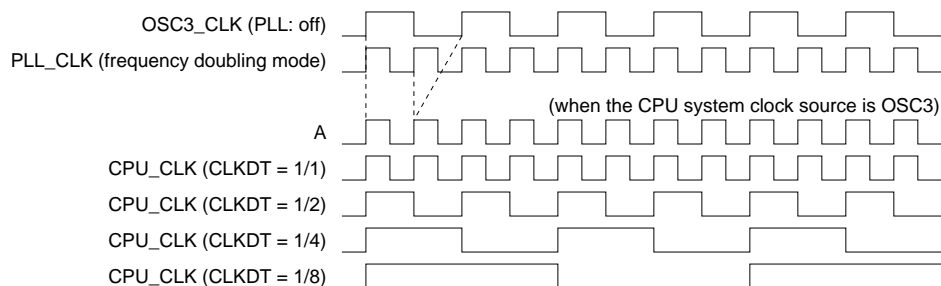
Figure II.4.17 Byte Data Reading from an 8-bit Device

## Bus Clock

The bus clock is generated by the BCU using the CPU system clock output from the clock generator. Figure II.4.18 shows the clock system.



### PLL\_CLK and CPU\_CLK



### BUS\_CLK

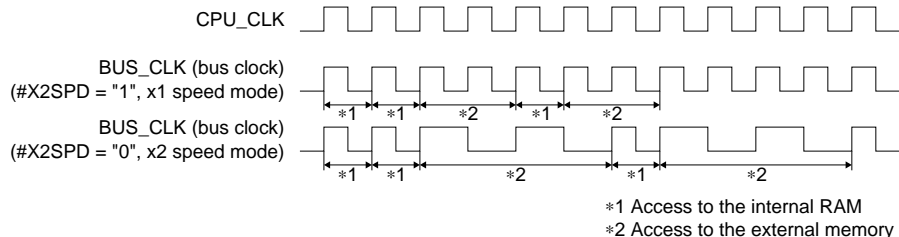


Figure II.4.18 Clock System

Since the bus clock is generated from the CPU system clock (CPU\_CLK), the following settings affect the bus clock:

1. Selection of an oscillation circuit (OSC3 or OSC1)
2. PLL configuration
3. CPU clock division ratio for power saving (1/8, 1/4, 1/2, or 1/1 of OSC3\_CLK or PLL\_CLK)

Items 2 and 3 apply when the high-speed (OSC3) oscillation circuit is selected as the CPU clock source. For details about the settings of the system clock, refer to "CLG (Clock Generator)".

Bus clock operation during standby is as follows:

Basic HALT mode: the BCU and bus clock continue operating.

HALT2 mode: the BCU and bus clock are stopped.

SLEEP mode: the BCU and bus clock are stopped.

**Note:** The configuration of the oscillation circuits and the PLL frequency multiplication mode provided and its setup method are different from model to model. Refer to the chapters for the oscillation circuits, such as "CLG (Clock Generator)".

### Bus Speed Mode

The CPU - bus clock ratio can be set using the #X2SPD pin as follows:

When #X2SPD = "1", x1 speed mode (CPU - bus clock ratio is 1 : 1) is set. The bus clock and the CPU system clock will be the same.

When #X2SPD = "0", x2 speed mode (CPU - bus clock ratio is 2 : 1) is set. In x2 speed mode, the bus clock will be dynamically varied according to the memory to be accessed.

- When an external memory area is accessed, the bus clock frequency becomes half of the CPU system clock.
- When the internal RAM/ROM area is accessed, the bus clock frequency becomes equal to the CPU system clock.

The x1 speed mode can be set up for each area even if the #X2SPD pin has been set to "0" by setting AxxBS (D[7:0]) / bus speed setting register (0x4813E).

In x1 speed mode, area 1 (internal I/O area) is accessed in 2 cycles of the CPU system clock.

In x2 speed mode, area 1 is accessed in 2 cycles by setting A1X1MD (D3) / BCLK select register (0x4813A) to "1" (default value = "0"). A1X1MD must be set to "1" when using the chip in x2 speed mode.

**Note:** The bus speed including the internal bus is limited, up to 40 MHz. When running the CPU with a 40 MHz or more operating clock, set the #X2SPD pin to "0", AxxBS to "0", and A1X1MD to "1".

### Bus Clock Output

The bus clock is also output from the BCLK pin to an external device. The BCLK output clock can be selected from among four types using BCLKSEL[1:0] (D[1:0]) / BCLK select register (0x4813A).

Table II.4.12 Selection of BCLK Output Clock

BCLKSEL1	BCLKSEL0	Output clock
1	1	PLL_CLK (PLL output clock)
1	0	OSC3_CLK (OSC3 oscillation clock)
0	1	BUS_CLK (Bus clock)
0	0	CPU_CLK (CPU operating clock)

## Bus Cycles in External System Interface

The following shows a sample SRAM connection the basic bus cycles.

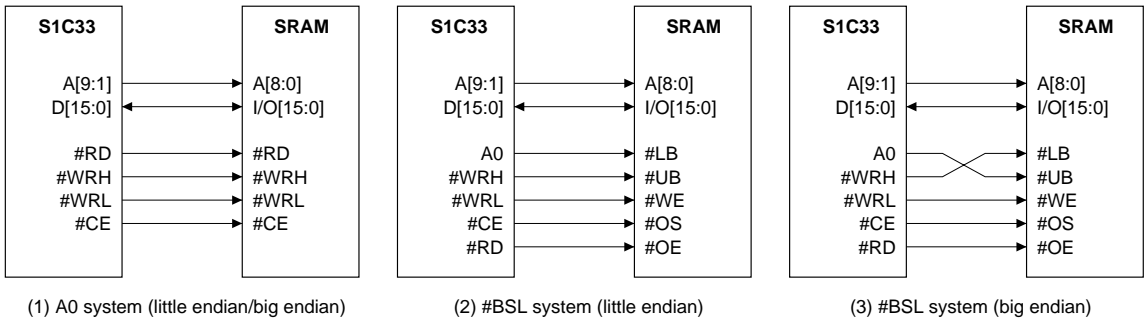


Figure II.4.19 Sample DRAM Connection

## SRAM Read Cycles

### Basic read cycle with no wait specified

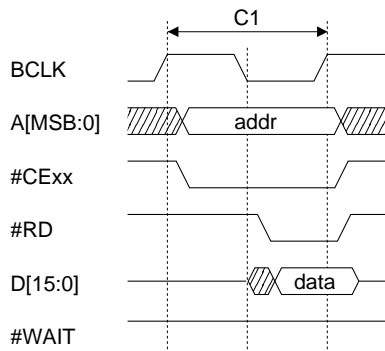


Figure II.4.20 Read Cycle with No Wait Specified

### Inserting wait cycles into a read cycle

The BCU supports the following two methods to insert wait cycles into a read cycle:

1. Specification using the BCU's AxxWT[2:0] control bits (0 to 7 cycles can be specified)
2. External wait request input to the #WAIT pin (can be used in combination with method 1)

#### 1. Wait insertion using the BCU's AxxWT[2:0] control bits

Example: 1-wait cycle is inserted (AxxWT[2:0] = "0x1"); no external wait request using the #WAIT pin

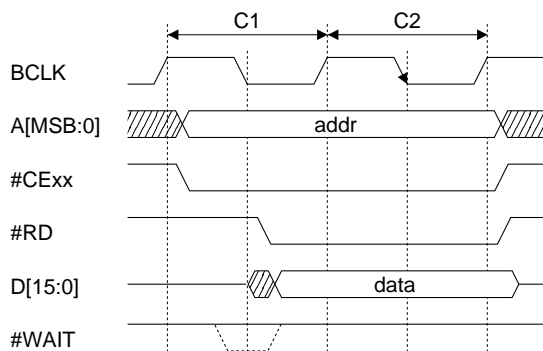


Figure II.4.21 Read Cycle with Wait (specified using AxxWT[2:0])

One wait cycle (C2) specified using AxxWT[2:0] is inserted after the basic read cycle (C1) shown in Figure II.4.20.

2. Wait request via the #WAIT pin

Example: An external wait request is input to the #WAIT pin to insert a wait into a read cycle (Figure II.4.21) in which a 1-wait cycle is inserted using AxxWT[2:0].

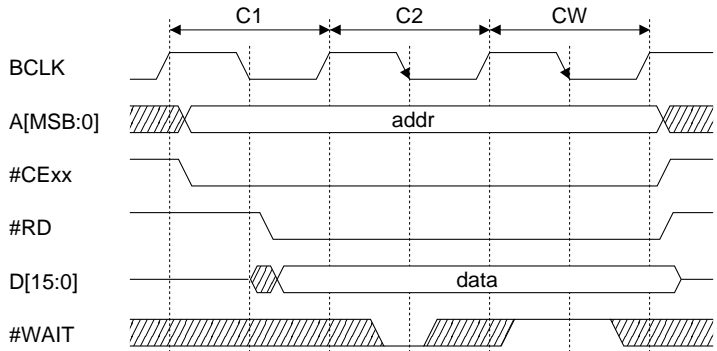


Figure II.4.22 Read Cycle with #WAIT Control

During a read cycle, the wait cycle insertion is controlled by a valid wait request honored by which the #WAIT signal is sampled as shown in the timings below.

- Falling edge of BCLK (bus clock) during the last wait cycle that is inserted using AxxWT[2:0] (or falling edge of BCLK during the C1 cycle if no wait cycle is inserted) \*
- Falling edge of BCLK (bus clock) in each wait cycle that is inserted using the #WAIT pin

When #WAIT = low is sampled at the timing above, 1 wait cycle will be inserted subsequent to the current cycle. When a high level is sampled, no wait cycle will be inserted.

\* Although the #WAIT signal is actually sampled every falling edge of BCLK during a read cycle, the sampling results other than above are not effective since the wait cycles inserted using AxxWT[2:0] have precedence over the #WAIT signal.

In both Figures II.4.21 and II.4.22, the #WAIT status sampled at the falling edge of BCLK during C2 cycle is effective because 1-wait cycle (C1) is inserted using AxxWT[2:0]. The #WAIT status at the falling edge of BCLK during C1 cycle does not affect wait cycle insertion.

In the case of Figure II.4.21, the read cycle is terminated at the end of the C2 cycle since #WAIT = high is sampled. (It assumes that a read-hold cycle and output disable delay time are not specified.)

In the case of Figure II.4.22, #WAIT = low is sampled, so a wait cycle (CW) due to the #WAIT signal is inserted subsequent to the C2 cycle. In this example, 1-wait cycle only is inserted because #WAIT = high is sampled during the inserted wait (CW) cycle. If #WAIT = low is sampled during the CW cycle, one more wait cycle is added.

**Note:** Insertion of wait cycles via the #WAIT pin is possible only when the device type in the bus conditions is configured as SRAM, and the SWAITE (D0) / Bus control register (0x4812E) is set to enable wait requests.

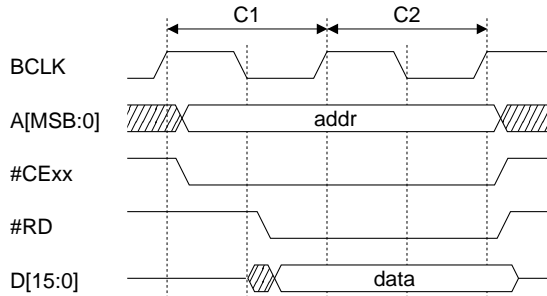


### Adding a read-hold cycle

In read cycles of this BCU, the rise of the #RD signal, negating the chip enable (#CE<sub>xx</sub>) signal and changing the address signals occur simultaneously at the same clock edge. No hold time is inserted to the chip enable and address signals. Therefore connecting a peripheral circuit, which changes its internal state by reading, to the bus may cause a malfunction. The read cycle hold time control register (0x4813C) is provided to avoid this problem. 1 cycle of hold time can be inserted to the #CE<sub>xx</sub> and A[MSB:0] signals after the #RD signal rises by setting the A<sub>xx</sub>RH bit corresponding to each area.

When the output disable delay time is set up and the insertion is enabled, an output disable cycle is inserted after the read hold cycle.

When no read-hold cycle is inserted (A<sub>xx</sub>RH = "0", number of wait cycles A<sub>xx</sub>WT[2:0] = "0x1")



When read-hold cycle is inserted (A<sub>xx</sub>RH = "1", number of wait cycles A<sub>xx</sub>WT[2:0] = "0x1")

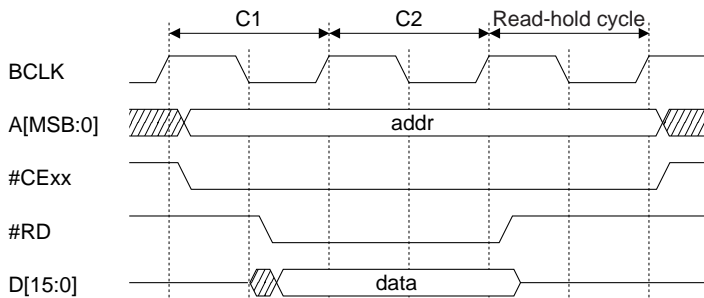
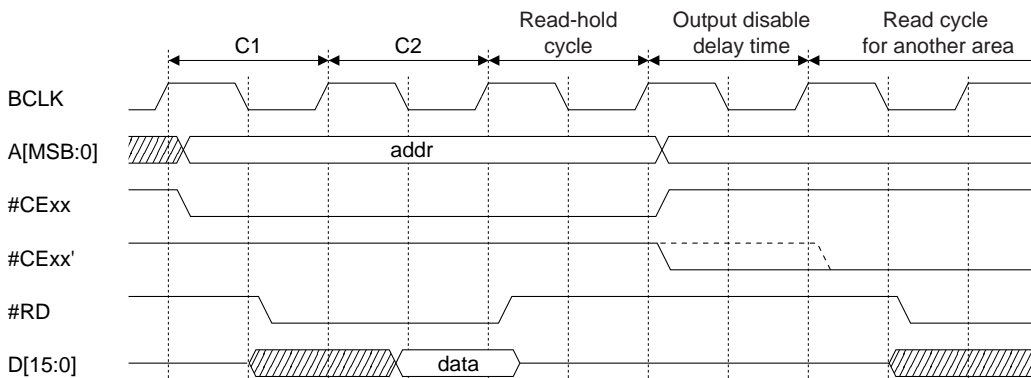


Figure II.4.23 Inserting a Read-Hold Cycle

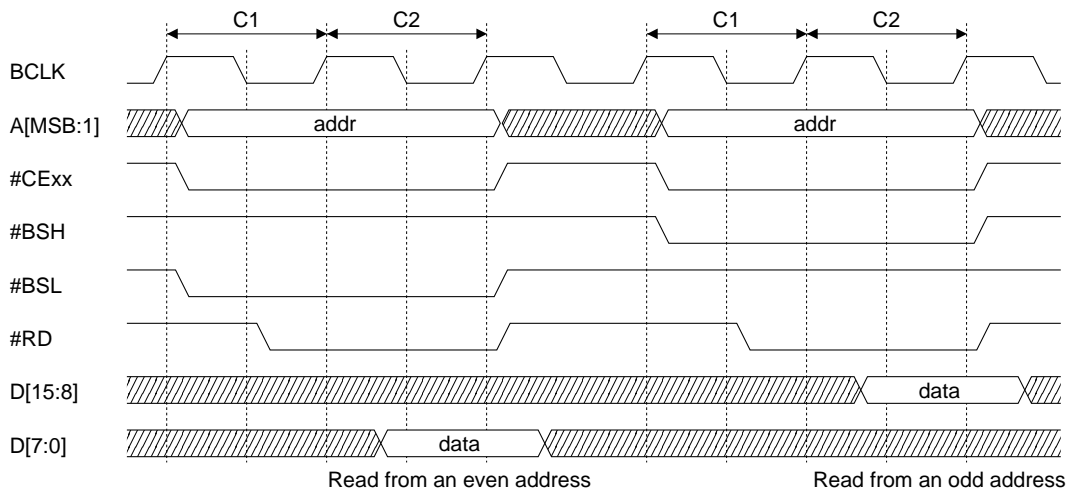


(example when output disable delay time is set to 1.5)

Figure II.4.24 Inserting a Read-Hold Cycle and an Output Disable Delay Time

**Note:** When an output disable delay time is inserted, the specified time of delay occurs until the #RD and #WRL signals are asserted. Note, however, that the chip enable signal may be asserted during the output disable delay time as well as it may be delayed.

Byte data reading from a 16-bit device in BSL method



(example when the number of wait cycles AxxWT[2:0] = "0x1")  
 Figure II.4.25 Byte Data Reading from a 16-bit Device in BSL Method

The #BSL signal is asserted for reading from an even address and the #BSH signal is asserted for an odd address.

## SRAM Write Cycles

### Basic write cycle in A0 method without a wait specification (1 wait)

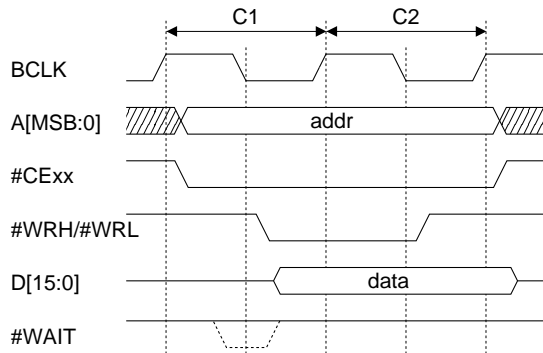


Figure II.4.26 Basic Write Cycle without a Wait Specification (A0 method)

### Basic write cycle in BSL method without a wait specification (1 wait)

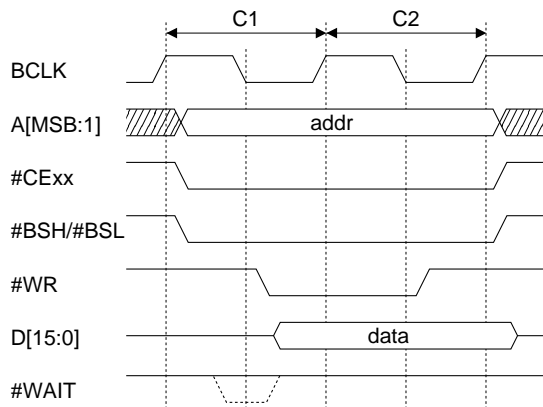


Figure II.4.27 Basic Write Cycle without a Wait Specification (BSL method)

**Note:** Always 1-wait cycle (C2) is inserted to write cycles even if no wait insertion is specified.

### Inserting wait cycles into a write cycle

The BCU supports the following two methods to insert wait cycles into a write cycle:

1. Specification using the BCU's AxxWT[2:0] control bits (2 to 7 cycles can be specified)  
 Always 1-wait cycle (C2) is inserted to write cycles, so specifying AxxWT[2:0] = "0x0" or "0x1" is not effective.
2. External wait request input to the #WAIT pin (can be used in combination with method 1)

#### 1. Wait insertion using the BCU's AxxWT[2:0] control bits

Example: 2-wait cycle is specified (AxxWT[2:0] = "0x2"); no external wait request using the #WAIT pin; A0 method

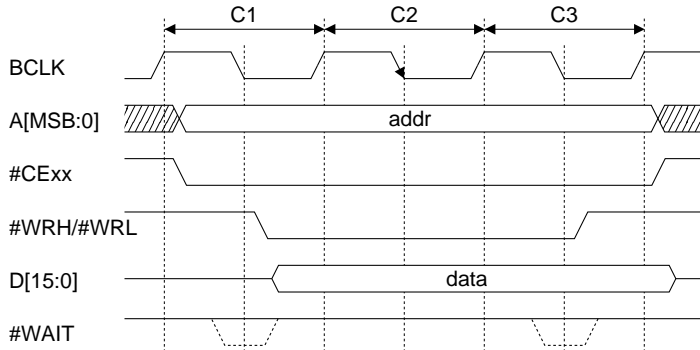


Figure II.4.28 Write Cycle in which 1 Wait Cycle is Added (specified using AxxWT[2:0])

Two wait cycles (C2 and C3) specified using AxxWT[2:0] are inserted after the basic write cycle (C1) shown in Figure II.4.26. (The C2 cycle in the basic write cycle is included in the number of cycles specified.)

#### 2. Wait request via the #WAIT pin

Example: An external wait request is input to the #WAIT pin to insert wait into a write cycle (Figure II.4.28) in which 2 wait cycles are inserted using AxxWT[2:0].

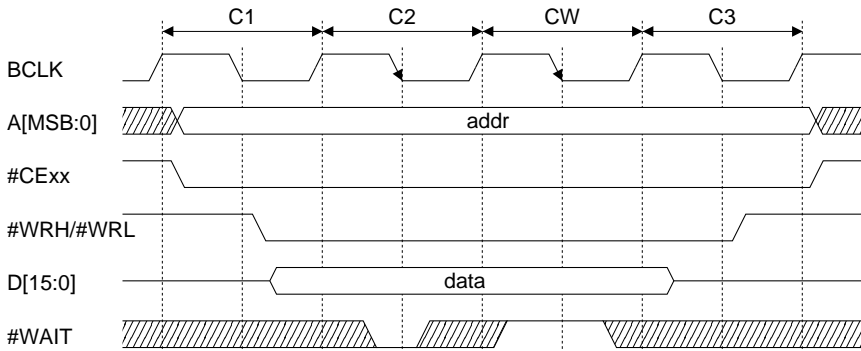


Figure II.4.29 Write Cycle with #WAIT Control

During a write cycle, the wait cycle insertion is controlled by a valid wait request honored by which the #WAIT signal is sampled as shown in the timings below.

- Falling edge of BCLK (bus clock) during the Second to last bus cycle including the wait cycles inserted using AxxWT[2:0] (or falling edge of BCLK during the C1 cycle if no wait cycle is specified)
- Falling edge of BCLK (bus clock) in each wait cycle that is inserted using the #WAIT pin

When #WAIT = low is sampled at the timing above, 1 wait cycle will be inserted subsequent to the current cycle. When a high level is sampled, no wait cycle will be added.

In both Figures II.4.28 and II.4.29, the #WAIT status sampled at the falling edge of BCLK during C2 cycle is effective because 2 wait cycles (C2 and C3) are inserted using AxxWT[2:0]. The #WAIT status at the falling edge of BCLK during C1 and C3 cycles do not affect wait cycle insertion.

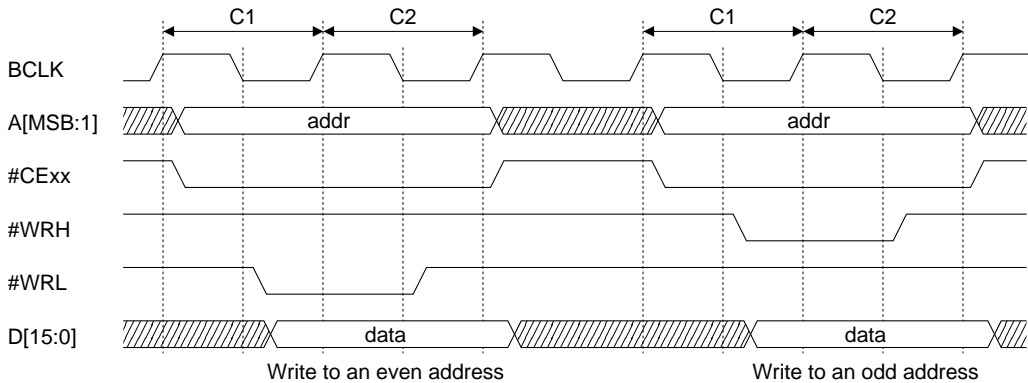
In the case of Figure II.4.28, the write cycle is terminated at the end of the C3 cycle since #WAIT = high is sampled.

In the case of Figure II.4.29, #WAIT = low is sampled, so a wait cycle (CW) due to the #WAIT signal is inserted subsequent to the C2 cycle. In this example, 1-wait cycle only is inserted because #WAIT = high is sampled during the inserted wait (CW) cycle. If #WAIT = low is sampled during the CW cycle, one more wait cycle is added.

**Note:** Insertion of wait cycles via the #WAIT pin is possible only when the device type in the bus conditions is configured as SRAM, and the SWAITE (D0) / Bus control register (0x4812E) is set to enable wait requests.

### Byte data writing to a 16-bit device

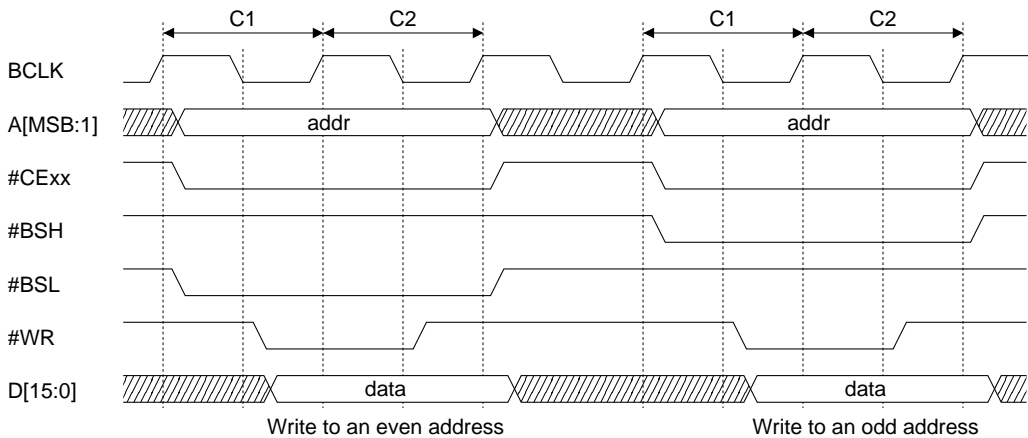
#### 1. A0 method (little endian)



(A0 method, little endian, no wait specified)  
Figure II.4.30 Byte Data Writing to a 16-bit Device

The #WRL signal is asserted for writing to an even address and the #WRH signal is asserted for an odd address.

#### 2. BSL method (little endian)



(BSL method, little endian, no wait specified)  
Figure II.4.31 Byte Data Writing to a 16-bit Device

The #BSL signal is asserted for writing to an even address and the #BSH signal is asserted for an odd address.

## Burst ROM Read Cycles

### Burst read cycle

Example: When 4-consecutive-burst and 2-wait cycles are set during the first access



Figure II.4.32 Burst Read Cycle

A burst read cycle occurs when area 10 or 9 is set for burst ROM and one of those areas is accessed for the following reasons:

#### 1) Instruction fetch

The burst read cycle is executed as long as a instruction fetch from contiguous addresses continues until  $A[2:1] = "11"$  (for 4-consecutive bursts); or  $A[3:1] = "111"$  (for 8-consecutive bursts)

#### 2) Word (32-bit) data read out

**Note:** A 16-bit output is supported for the burst ROM. Set the device size to 16 bits.

### Wait cycles during burst read

In the first bus operation, 0 to 7 wait cycles can be inserted using the wait control bits  $A10WT[2:0]$  ( $D[2:0]$ ) / Areas 10–9 set-up register (0x48126) in the same way as for ordinary SRAM. For the wait cycles to be inserted in the burst cycle that follows, use a dedicated wait control bits,  $A10BW[1:0]$ , which is only used for reading bursts. The wait cycles can be set in the range from 0 to 3 using these bits.

Note that no wait cycle via the #WAIT pin can be inserted into the burst-read cycle.

### Write cycle to burst ROM area

If area 10 or 9 is set for burst ROM, a SRAM write cycle is executed when a write to that area is attempted. In this case, wait cycles via the #WAIT pin can be inserted.

## Releasing External Bus

The external bus is normally controlled by the CPU, but the BCU is designed to release control of the bus ownership to an external device. This function is enabled by writing "1" to SEMAS (D2) / Bus control register (0x4812E) (disabled by default). The #BUSREQ (P34) and #BUSACK (P35) pins are used for control of the bus ownership. To direct the P34 and P35 pins for input/output of the #BUSREQ and #BUSACK signals, write "1" to CFP34 (D4) and CFP35 (D5) / P3 function select register (0x402DC [Byte]).

### Sequence in which control of the bus is released

This sequence is described below.

1. The external bus master device requesting control of the bus ownership lowers the #BUSREQ pin.
2. The CPU keeps monitoring the status of the #BUSREQ pin, so that when this pin is lower, the CPU terminates the bus cycle being executed and places the signals listed below in high-impedance state one cycle later:  
A[MSB:0], D[15:0], #RD, #WRL, #WRH, #HCAS, #LCAS, #CE<sub>xx</sub>  
Then the CPU lowers the #BUSACK pin to inform the external device that control of the bus ownership has been released.
3. One cycle later, the external bus starts its own bus cycle. The external bus master must hold the #BUSREQ pin low until the bus cycle is completed.
4. After completing the necessary bus cycles, the external bus master places the bus in high-impedance state and releases the #BUSREQ pin back high.
5. After confirming that the #BUSREQ pin is raised again, the CPU raises the #BUSACK pin one cycle later and resumes the processing that has been suspended.

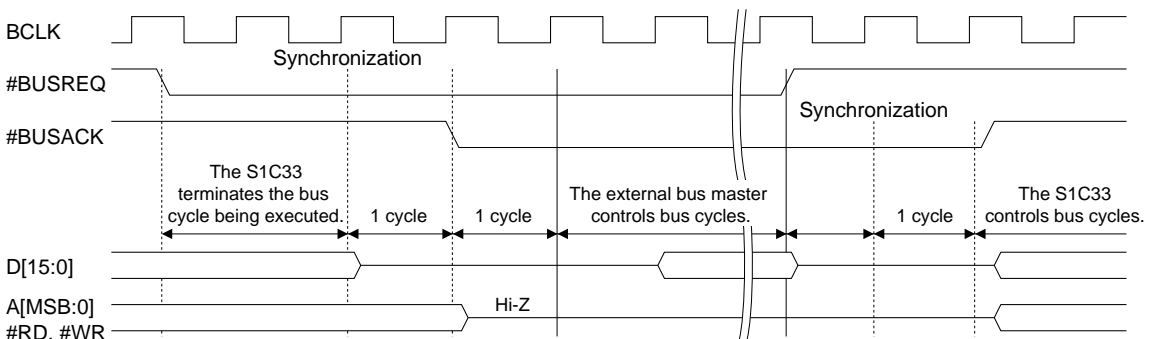


Figure II.4.33 External Bus Release Timing

If control of the bus ownership is requested during a DMA transfer by the internal DMA controller, the DMA transfer under way is suspended at a break in data to accept the request for bus ownership control. The DMA transfer that has been kept pending is restarted when the CPU gains control of the bus ownership.

## Power-down Control by External Device

In addition to requesting the releasing of bus ownership control described above, it is possible to place the CPU in a HALT state by using the #BUSREQ signal. This allows the CPU to be stopped during bus operation by an external bus master in order to conserve power.

This function is enabled by writing "1" to SEPD (D1) / Bus control register (0x4812E).

If SEPD = "1", the CPU and the BCU stop operating when the #BUSREQ pin is lowered, thus entering a HALT state. This HALT state is not cleared by an interrupt from the internal peripheral circuits and remains set until the #BUSREQ pin is released back high. Unlike in the case of ordinary releasing of the bus by #BUSREQ, the address bus and bus control signals are not placed in high-impedance state.

## I/O Memory of BCU

Table II.4.13 shows the control bits of the BCU. These I/O memories are mapped into the area (0x48000 and following addresses) used for the internal 16-bit peripheral circuits. However, these I/O memories can be accessed in bytes or words, as well as in half-words.

For the control bits of the external system interface pins assigned to the I/O ports, refer to the "Input/output Ports" chapter.

The I/O memory for BCU contains DRAM control registers/bits. Leave these registers/bits unchanged (as default settings).

Table II.4.13 Control Bits of External System Interface

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Areas 18–15 set-up register	0048120 (HW)	DF	–	reserved	–	–	–	0 when being read.	
		DE	A18SZ	Areas 18–17 device size selection	1 8 bits   0 16 bits	0	R/W		
		DD	A18DF1	Areas 18–17 output disable delay time	A18DF[1:0]   Number of cycles		1	R/W	
		DC	A18DF0		1   1	3.5			
					1   0	2.5			
					0   1	1.5			
					0   0	0.5			
		DB	–	reserved	–	–	–	–	0 when being read.
		DA	A18WT2	Areas 18–17 wait control	A18WT[2:0]   Wait cycles		1	R/W	
		D9	A18WT1		1   1   1	7			
		D8	A18WT0		1   1   0	6			
					1   0   1	5			
					1   0   0	4			
					0   1   1	3			
					0   1   0	2			
				0   0   1	1				
		0   0   0	0						
D7	–	reserved	–	–	–	–	0 when being read.		
D6	A16SZ	Areas 16–15 device size selection	1 8 bits   0 16 bits	0	R/W				
D5	A16DF1	Areas 16–15 output disable delay time	A16DF[1:0]   Number of cycles		1	R/W			
D4	A16DF0		1   1	3.5					
			1   0	2.5					
			0   1	1.5					
			0   0	0.5					
D3	–	reserved	–	–	–	–	0 when being read.		
D2	A16WT2	Areas 16–15 wait control	A16WT[2:0]   Wait cycles		1	R/W			
D1	A16WT1		1   1   1	7					
D0	A16WT0		1   1   0	6					
			1   0   1	5					
			1   0   0	4					
			0   1   1	3					
			0   1   0	2					
		0   0   1	1						
		0   0   0	0						
Areas 14–13 set-up register	0048122 (HW)	DF–9	–	reserved	–	–	–	0 when being read.	
		D8	A14DRA	Area 14 DRAM selection	1 Used   0 Not used	0	R/W		
		D7	A13DRA	Area 13 DRAM selection	1 Used   0 Not used	0	R/W		
		D6	A14SZ	Areas 14–13 device size selection	1 8 bits   0 16 bits	0	R/W		
		D5	A14DF1	Areas 14–13 output disable delay time	A14DF[1:0]   Number of cycles		1	R/W	
		D4	A14DF0		1   1	3.5			
					1   0	2.5			
					0   1	1.5			
					0   0	0.5			
		D3	–	reserved	–	–	–	–	0 when being read.
D2	A14WT2	Areas 14–13 wait control	A14WT[2:0]   Wait cycles		1	R/W			
D1	A14WT1		1   1   1	7					
D0	A14WT0		1   1   0	6					
			1   0   1	5					
			1   0   0	4					
			0   1   1	3					
			0   1   0	2					
		0   0   1	1						
		0   0   0	0						



Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Areas 12–11 set-up register	0048124 (HW)	DF-7	–	reserved	–	–	–	0 when being read.		
		D6	A12SZ	Areas 12–11 device size selection	1   8 bits   0   16 bits	0	R/W			
		D5	A12DF1	Areas 12–11	output disable delay time	A12DF[1:0]   Number of cycles	1	R/W		
		D4	A12DF0	1   1   3.5						
		1   0   2.5								
		0   1   1.5								
		0   0   0.5								
		D3	–	reserved	–	–	–	0 when being read.		
		D2	A12WT2	Areas 12–11 wait control	A12WT[2:0]   Wait cycles	1	R/W	1		
		D1	A12WT1	1   1   1   7						
		D0	A12WT0	1   1   0   6						
		1   0   1   5								
		1   0   0   4								
0   1   1   3										
0   1   0   2										
0   0   1   1										
0   0   0   0										
Areas 10–9 set-up register	0048126 (HW)	DF	–	reserved	–	–	–	0 when being read.		
		DE	A10IR2	Area 10 internal ROM size selection	A10IR[2:0]   ROM size	1	R/W			
		DD	A10IR1						1   1   1   2MB	
		DC	A10IR0						1   1   0   1MB	
		1   0   1   512KB								
		1   0   0   256KB								
		0   1   1   128KB								
		0   1   0   64KB								
		0   0   1   32KB								
		0   0   0   16KB								
		DB	–	reserved	–	–	–	0 when being read.		
		DA	A10BW1	Areas 10–9 burst ROM burst read cycle wait control	A10BW[1:0]   Wait cycles	0	R/W	0		
		D9	A10BW0							1   1   3
		1   0   2								
		0   1   1								
		D8	A10DRA	Area 10 burst ROM selection	1   Used   0   Not used	0	R/W			
		D7	A9DRA	Area 9 burst ROM selection	1   Used   0   Not used	0	R/W			
		D6	A10SZ	Areas 10–9 device size selection	1   8 bits   0   16 bits	0	R/W			
		D5	A10DF1	Areas 10–9	A10DF[1:0]   Number of cycles	1	R/W	1		
		D4	A10DF0	output disable delay time						1   1   3.5
1   0   2.5										
0   1   1.5										
0   0   0.5										
D3	–	reserved	–	–	–	0 when being read.				
D2	A10WT2	Areas 10–9 wait control	A10WT[2:0]   Wait cycles	1	R/W	1				
D1	A10WT1	1   1   1   7								
D0	A10WT0	1   1   0   6								
1   0   1   5										
1   0   0   4										
0   1   1   3										
0   1   0   2										
0   0   1   1										
0   0   0   0										
Areas 8–7 set-up register	0048128 (HW)	DF-9	–	reserved	–	–	–	0 when being read.		
		D8	A8DRA	Area 8 DRAM selection	1   Used   0   Not used	0	R/W			
		D7	A7DRA	Area 7 DRAM selection	1   Used   0   Not used	0	R/W			
		D6	A8SZ	Areas 8–7 device size selection	1   8 bits   0   16 bits	0	R/W			
		D5	A8DF1	Areas 8–7	A8DF[1:0]   Number of cycles	1	R/W	1		
		D4	A8DF0	output disable delay time						1   1   3.5
		1   0   2.5								
		0   1   1.5								
0   0   0.5										
D3	–	reserved	–	–	–	0 when being read.				
D2	A8WT2	Areas 8–7 wait control	A8WT[2:0]   Wait cycles	1	R/W	1				
D1	A8WT1	1   1   1   7								
D0	A8WT0	1   1   0   6								
1   0   1   5										
1   0   0   4										
0   1   1   3										
0   1   0   2										
0   0   1   1										
0   0   0   0										

II-4 CORE BLOCK: BCU (Bus Control Unit)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Areas 6-4 set-up register	004812A (HW)	DF-E	-	reserved		-	-	0 when being read.		
		DD	A6DF1	Area 6	A6DF[1:0] Number of cycles	1	R/W			
		DC	A6DF0	output disable delay time					1 1	3.5
									1 0	2.5
									0 1	1.5
									0 0	0.5
		DB	-	reserved		-	-	0 when being read.		
		DA	A6WT2	Area 6 wait control	A6WT[2:0] Wait cycles	1	R/W			
		D9	A6WT1						1 1 1	7
		D8	A6WT0						1 1 0	6
									1 0 1	5
									1 0 0	4
									0 1 1	3
									0 1 0	2
			0 0 1	1						
D7	-	reserved		-	-	0 when being read.				
D6	A5SZ	Areas 5-4 device size selection	1	8 bits	0	16 bits	0	R/W		
D5	A5DF1	Areas 5-4	A5DF[1:0] Number of cycles	1	R/W					
D4	A5DF0	output disable delay time					1 1	3.5		
							1 0	2.5		
							0 1	1.5		
D3	-	reserved		-	-	0 when being read.				
D2	A5WT2	Areas 5-4 wait control	A5WT[2:0] Wait cycles	1	R/W					
D1	A5WT1						1 1 1	7		
D0	A5WT0						1 1 0	6		
							1 0 1	5		
							1 0 0	4		
							0 1 1	3		
							0 1 0	2		
							0 0 1	1		
			0 0 0	0						
Bus control register	004812E (HW)	DF	RBCLK	BCLK output control	1	Fixed at H	0	Enabled	0	R/W
		DE	-	reserved		-	-	0	-	Writing 1 not allowed.
		DD	RBST8	Burst ROM burst mode selection	1	8-successive	0	4-successive	0	R/W
		DC	REDO	DRAM page mode selection	1	EDO	0	Fast page	0	R/W
		DB	RCA1	Column address size selection	RCA[1:0] Size	0	R/W			
		DA	RCA0						1 1	11
									1 0	10
									0 1	9
					0 0	8				
		D9	RPC2	Refresh enable	1	Enabled	0	Disabled	0	R/W
		D8	RPC1	Refresh method selection	1	Self-refresh	0	CBR-refresh	0	R/W
		D7	RPC0	Refresh RPC delay setup	1	2.0	0	1.0	0	R/W
		D6	RRA1	Refresh RAS pulse width selection	RRA[1:0] Number of cycles	0	R/W			
		D5	RRA0						1 1	5
		1 0	4							
		0 1	3							
			0 0	2						
D4	-	reserved		-	-	0	-	Writing 1 not allowed.		
D3	SBUSST	External interface method selection	1	#BSL	0	A0	0	R/W		
D2	SEMAS	External bus master setup	1	Existing	0	Nonexistent	0	R/W		
D1	SEPD	External power-down control	1	Enabled	0	Disabled	0	R/W		
D0	SWAITE	#WAIT enable	1	Enabled	0	Disabled	0	R/W		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
DRAM timing set-up register	0048130 (HW)	DF-C	-	reserved		-	-	-	0 when being read.	
		DB	A3EEN	Area 3 emulation	1   Internal ROM	0   Emulation	1	R/W		
		DA	CEFUNC1	#CE pin function selection	CEFUNC[1:0]   #CE output		0	R/W		
		D9	CEFUNC0		1   x	#CE7/8..#CE17/18	0			
					0   1	#CE6..#CE17				
						0   0	#CE4..#CE10			
		D8	CRAS	Successive RAS mode setup	1   Successive	0   Normal	0	R/W		
		D7	RPRC1	DRAM RAS precharge cycles selection	RPRC[1:0]   Number of cycles		0	R/W		
		D6	RPRC0		1   1	4	0			
					1   0	3				
					0   1	2				
						0   0	1			
		D5	-	reserved		-	-	-	-	0 when being read.
		D4	CASC1	DRAM CAS cycles selection	CASC[1:0]   Number of cycles		0	R/W		
D3	CASC0	1   1	4		0					
		1   0	3							
		0   1	2							
				0   0	1					
D2	-	reserved		-	-	-	-	0 when being read.		
D1	RASC1	DRAM RAS cycles selection	RASC[1:0]   Number of cycles		0	R/W				
D0	RASC0		1   1	4	0					
			1   0	3						
			0   1	2						
				0   0	1					
Access control register	0048132 (HW)	DF	A18IO	Area 18, 17 internal/external access	1   Internal access	0   External access	0	R/W		
		DE	A16IO	Area 16, 15 internal/external access			0	R/W		
		DD	A14IO	Area 14, 13 internal/external access			0	R/W		
		DC	A12IO	Area 12, 11 internal/external access			0	R/W		
		DB	-	reserved		-	0	-	0 when being read.	
		DA	A8IO	Area 8, 7 internal/external access	1   Internal access	0   External access	0	R/W		
		D9	A6IO	Area 6 internal/external access			0	R/W		
		D8	A5IO	Area 5, 4 internal/external access			0	R/W		
		D7	A18EC	Area 18, 17 endian control	1   Big endian	0   Little endian	0	R/W		
		D6	A16EC	Area 16, 15 endian control			0	R/W		
		D5	A14EC	Area 14, 13 endian control			0	R/W		
		D4	A12EC	Area 12, 11 endian control			0	R/W		
		D3	A10EC	Area 10, 9 endian control			0	R/W		
		D2	A8EC	Area 8, 7 endian control			0	R/W		
D1	A6EC	Area 6 endian control			0	R/W				
D0	A5EC	Area 5, 4 endian control			0	R/W				
G/A read signal control register	0048138 (HW)	DF	A18AS	Area 18, 17 address strobe signal	1   Enabled	0   Disabled	0	R/W		
		DE	A16AS	Area 16, 15 address strobe signal			0	R/W		
		DD	A14AS	Area 14, 13 address strobe signal			0	R/W		
		DC	A12AS	Area 12, 11 address strobe signal			0	R/W		
		DB	-	reserved		-	0	-	0 when being read.	
		DA	A8AS	Area 8, 7 address strobe signal	1   Enabled	0   Disabled	0	R/W		
		D9	A6AS	Area 6 address strobe signal			0	R/W		
		D8	A5AS	Area 5, 4 address strobe signal			0	R/W		
		D7	A18RD	Area 18, 17 read signal	1   Enabled	0   Disabled	0	R/W		
		D6	A16RD	Area 16, 15 read signal			0	R/W		
		D5	A14RD	Area 14, 13 read signal			0	R/W		
		D4	A12RD	Area 12, 11 read signal			0	R/W		
		D3	-	reserved		-	0	-	0 when being read.	
		D2	A8RD	Area 8, 7 read signal	1   Enabled	0   Disabled	0	R/W		
D1	A6RD	Area 6 read signal			0	R/W				
D0	A5RD	Area 5, 4 read signal			0	R/W				
BCLK select register	004813A (B)	D7-4	-	reserved		-	0	-	0 when being read.	
		D3	A1X1MD	Area 1 access-speed	1   2 cycles	0   4 cycles	0	R/W	x2 speed mode only (Setting "0" is not allowed.)	
		D2	-	reserved		-	0	-	0 when being read.	
		D1	BCLKSEL1	BCLK output clock selection	BCLKSEL[1:0]   BCLK		0	R/W		
D0	BCLKSEL0	1   1	PLL_CLK		0					
			1   0	OSC3_CLK						
			0   1	BUS_CLK						
			0   0	CPU_CLK						

## II-4 CORE BLOCK: BCU (Bus Control Unit)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Read cycle hold time control register	004813C (B)	D7	<b>A18RH</b>	Areas 18–17 read hold cycle	1	Inserted	0	Not inserted	0	R/W	
		D6	<b>A16RH</b>	Areas 16–15 read hold cycle					0	R/W	
		D5	<b>A14RH</b>	Areas 14–13 read hold cycle					0	R/W	
		D4	<b>A12RH</b>	Areas 12–11 read hold cycle					0	R/W	
		D3	<b>A10RH</b>	Areas 10–9 read hold cycle					0	R/W	
		D2	<b>A8RH</b>	Areas 8–7 read hold cycle					0	R/W	
		D1	<b>A6RH</b>	Area 6 read hold cycle					0	R/W	
		D0	<b>A5RH</b>	Areas 5–4 read hold cycle					0	R/W	
Bus speed setting register	004813E (B)	D7	<b>A18BS</b>	Areas 18–17 bus speed selection	1	x1 speed	0	#X2SPD	0	R/W	This register's setting is meaningful when #X2SPD = 0 (2 × speed mode)
		D6	<b>A16BS</b>	Areas 16–15 bus speed selection					0	R/W	
		D5	<b>A14BS</b>	Areas 14–13 bus speed selection					0	R/W	
		D4	<b>A12BS</b>	Areas 12–11 bus speed selection					0	R/W	
		D3	<b>A10BS</b>	Areas 10–9 bus speed selection					0	R/W	
		D2	<b>A8BS</b>	Areas 8–7 bus speed selection					0	R/W	
		D1	<b>A6BS</b>	Area 6 bus speed selection					0	R/W	
		D0	<b>A5BS</b>	Areas 5–4 bus speed selection					0	R/W	

**A18SZ:** Areas 18–17 device size selection (DE) / Areas 18–15 set-up register (0x48120)

**A16SZ:** Areas 16–15 device size selection (D6) / Areas 18–15 set-up register (0x48120)

**A14SZ:** Areas 14–13 device size selection (D6) / Areas 14–13 set-up register (0x48122)

**A12SZ:** Areas 12–11 device size selection (D6) / Areas 12–11 set-up register (0x48124)

**A10SZ:** Areas 10–9 device size selection (D6) / Areas 10–9 set-up register (0x48126)

**A8SZ:** Areas 8–7 device size selection (D6) / Areas 8–7 set-up register (0x48128)

**A5SZ:** Areas 5–4 device size selection (D6) / Areas 6–4 set-up register (0x4812A)

Select the size of the device connected to each area.

Write "1": 8 bits

Write "0": 16 bits

Read: Valid

A device size can be selected for every two areas. An 8-bit size is selected by writing "1" to AxxSZ and a 16-bit size is selected by writing "0" to AxxSZ. Area 6 has its first half (0x300000 through 0x37FFFF) fixed to an 8-bit device and the last half (0x380000 through 0x3FFFFFF) fixed to a 16-bit device.

At cold start, these bits are set to "0" (16 bits). At hot start, these bits retain their status before being initialized.

**A18DF1–A18DF0:** Areas 18–17 output disable delay time (D[D:C]) / Areas 18–15 set-up register (0x48120)

**A16DF1–A16DF0:** Areas 16–15 output disable delay time (D[5:4]) / Areas 18–15 set-up register (0x48120)

**A14DF1–A14DF0:** Areas 14–13 output disable delay time (D[5:4]) / Areas 14–13 set-up register (0x48122)

**A12DF1–A12DF0:** Areas 12–11 output disable delay time (D[5:4]) / Areas 12–11 set-up register (0x48124)

**A10DF1–A10DF0:** Areas 10–9 output disable delay time (D[5:4]) / Areas 10–9 set-up register (0x48126)

**A8DF1–A8DF0:** Areas 8–7 output disable delay time (D[5:4]) / Areas 8–7 set-up register (0x48128)

**A6DF1–A6DF0:** Area 6 output disable delay time (D[D:C]) / Areas 6–4 set-up register (0x4812A)

**A5DF1–A5DF0:** Areas 5–4 output disable delay time (D[5:4]) / Areas 6–4 set-up register (0x4812A)

Set the output-disable delay time.

Table II.4.14 Output Disable Delay Time

AxxDF1	AxxDF0	Delay time
1	1	3.5 cycles
1	0	2.5 cycles
0	1	1.5 cycles
0	0	0.5 cycles

When using a device that has a long output-disable time, set a delay time to ensure that no contention for the data bus occurs during the bus operation immediately after a device is read.

At cold start, these bits are set to "11" (3.5 cycles). At hot start, the bits retain their status before being initialized.

**A18WT2–A18WT0:** Areas 18–17 wait control (D[A:8]) / Areas 18–15 set-up register (0x48120)

**A16WT2–A16WT0:** Areas 16–15 wait control (D[2:0]) / Areas 18–15 set-up register (0x48120)

**A14WT2–A14WT0:** Areas 14–13 wait control (D[2:0]) / Areas 14–13 set-up register (0x48122)

**A12WT2–A12WT0:** Areas 12–11 wait control (D[2:0]) / Areas 12–11 set-up register (0x48124)

**A10WT2–A10WT0:** Areas 10–9 wait control (D[2:0]) / Areas 10–9 set-up register (0x48126)

**A8WT2–A8WT0:** Areas 8–7 wait control (D[2:0]) / Areas 8–7 set-up register (0x48128)

**A6WT2–A6WT0:** Area 6 wait control (D[A:8]) / Areas 6–4 set-up register (0x4812A)

**A5WT2–A5WT0:** Areas 5–4 wait control (D[2:0]) / Areas 6–4 set-up register (0x4812A)

Set the number of wait cycles to be inserted when accessing an SRAM device.

The values 0 through 7 written to the control bits equal the number of wait cycles inserted.

Note that the write cycle consists of a minimum of two cycles, so that a writing 0 or 1 is invalid.

When an SRAM device is connected, wait cycles derived via the #WAIT pin can also be inserted. In this case too, the wait cycles set by AxxWT are valid. The burst read cycle of a burst ROM (except for the first access) does not have any wait cycle inserted. The first read cycle of a burst ROM and the write cycle to the burst ROM area have wait cycles inserted that are set by AxxWT. Wait cycles derived from the #WAIT pin also can be inserted in the cycle for writing to the burst ROM area.

At cold start, these bits are set to "111" (7 cycles). At hot start, the bits retain their status before being initialized.

**A14DRA:** Area 14 DRAM selection (D8) / Areas 14–13 set-up register (0x48122)

**A13DRA:** Area 13 DRAM selection (D7) / Areas 14–13 set-up register (0x48122)

**A8DRA:** Area 8 DRAM selection (D8) / Areas 8–7 set-up register (0x48128)

**A7DRA:** Area 7 DRAM selection (D7) / Areas 8–7 set-up register (0x48128)

Leave these bits unchanged ("0").

**A10IR2–A10IR0:** Area 10 internal ROM size selection (D[E:C]) / Areas 10–9 set-up register (0x48126)

Select an area 10 internal memory size.

Table II.4.15 Area 10 Internal ROM Size

A10IR2	A10IR1	A10IR0	ROM size
0	0	0	16 KB
0	0	1	32 KB
0	1	0	64 KB
0	1	1	128 KB
1	0	0	256 KB
1	0	1	512 KB
1	1	0	1 MB
1	1	1	2 MB (default)

At cold start, A10IR is set to "111" (2 MB). At hot start, A10IR retains its status before being initialized.

**A10BW1–A10BW0:** Burst read cycle wait control (D[A:9]) / Areas 10–9 set-up register (0x48126)

Set the number of wait cycles inserted during a burst read.

The values 0 to 3 written to the bits constitute the number of wait cycles inserted. The contents set here are applied to both areas 10 and 9. The wait cycles set by AxxWT are inserted in the first read cycle of burst ROM and in the burst ROM write cycle. For the burst ROM write cycle, the wait cycles set via the #WAIT pin can also be used.

At cold start, A10BW is set to "0" (no wait cycle). At hot start, A10BW retains its status before being initialized.

**A10DRA:** Area 10 burst ROM selection (D8) / Areas 10–9 set-up register (0x48126)

**A9DRA:** Area 9 burst ROM selection (D7) / Areas 10–9 set-up register (0x48126)

Set areas 10 and 9 for use of burst ROM.

Write "1": Burst ROM is used

Write "0": Burst ROM is not used

Read: Valid

When using burst ROM, write "1" to the control bit. The ordinary SRAM interface is selected by writing "0" to the bit. Area 9 can only be used when the CEFUNC = "00".

At cold start, these bits are set to "0" (burst ROM not used). At hot start, the bits retain their status before being initialized.

## II-4 CORE BLOCK: BCU (Bus Control Unit)

---

### **RBCLK:** BCLK output control (DF) / Bus control register (0x4812E)

---

Control the bus clock BCLK to enable or disable external output.

Write "1": Fixed at high level  
Write "0": Output enabled  
Read: Valid

To stop outputting the bus clock from the BCLK pin, write "1" to RBCLK. When the clock output is stopped, the BCLK pin is fixed at high level. The bus clock output from the BCLK pin is enabled by writing "0" to RBCLK. The bus clock output from the BCLK pin also is stopped in the HALT2 and the SLEEP modes. At cold start, the RBCLK is set to "0" (output enabled). At hot start, RBCLK retains its status before being initialized.

---

### **RBST8:** Burst mode selection (DD) / Bus control register (0x4812E)

---

Set the operation mode during a burst read.

Write "1": 8-successive-burst mode  
Write "0": 4-successive-burst mode  
Read: Valid

The 8-successive-burst mode is selected by writing "1" to RBST8 and the 4-successive-burst mode is selected by writing "0" to RBST8. This setting is valid when areas 10 and 9 are set for burst ROM, and the setting is applied to both areas simultaneously. At cold start, RBST8 is set to "0" (4-successive-burst mode). At hot start, RBST8 retains its status before being initialized.

---

### **REDO:** Page mode selection (DC) / Bus control register (0x4812E)

---

Leave this bit unchanged ("0").

---

### **RCA1–RCA0:** Column address size selection (D[B:A]) / Bus control register (0x4812E)

---

Leave these bits unchanged ("0").

---

### **RPC2:** Refresh enable (D9) / Bus control register (0x4812E)

---

Leave this bit unchanged ("0").

---

### **RPC1:** Refresh method selection (D8) / Bus control register (0x4812E)

---

Leave this bit unchanged ("0").

---

### **RPC0:** Refresh RPC delay (D7) / Bus control register (0x4812E)

---

Leave this bit unchanged ("0").

---

### **RRA1–RRA0:** Refresh RAS pulse width selection (D[6:5]) / Bus control register (0x4812E)

---

Leave these bits unchanged ("0").

---

### **SBUSST:** External interface method select register (D3) / Bus control register (0x4812E)

---

Select the interface method of an SRAM device.

Write "1": #BSL system  
Write "0": A0 system  
Read: Valid

When using the #BSL system, write "1" to SBUSST. The contents set here are applied to all areas that are set for the SRAM type. At cold start, SBUSST is set to "0" (A0 system). At hot start, SBUSST retains its status before being initialized.

**SEMAS: External bus master setup (D2) / Bus control register (0x4812E)**

Specify whether an external bus master exists.

Write "1": Existing  
 Write "0": Nonexistent  
 Read: Valid

A request for bus ownership control via the #BUSREQ pin is made acceptable by writing "1" to SEMAS. If the system does not have any external bus master, fix this register at "0".

At cold start, SEMAS is set to "0" (nonexistent). At hot start, SEMAS retains its status before being initialized.

**SEPD: External power-down control (D1) / Bus control register (0x4812E)**

Enable or disable the CPU's power-down control by an external bus master.

Write "1": Enabled  
 Write "0": Disabled  
 Read: Valid

Power-down control via an external pin (#BUSREQ) is enabled by writing "1" to SEPD. If the #BUSREQ pin is lowered when external power-down control is thus enabled, the CPU is placed in a HALT state, allowing for reduction in power consumption.

At cold start, SEPD is set to "0" (disabled). At hot start, SEPD retains its status before being initialized.

**SWAITE: #WAIT enable (D0) / Bus control register (0x4812E)**

Enable or disable wait cycle control via the #WAIT pin.

Write "1": Enabled  
 Write "0": Disabled  
 Read: Valid

A wait request from an SRAM device is made acceptable by writing "1" to SWAITE. The wait request signal input from the #WAIT pin is sampled at each falling edge of the bus clock when executing an SRAM read/write cycle.

Wait cycles are inserted until the wait request signal is sampled and detected as high (inactive).

Wait control for 0 to 7 cycles can be accomplished by AxxWT without using the #WAIT pin. However, since the setting via AxxWT is applied to every two areas, the number of wait cycles may be controlled individually in each area or more than 7 wait cycles may be set. In such a case, use an external wait request via the #WAIT pin.

Wait requests from the #WAIT pin are ignored when SWAITE = "0".

The contents set here are applied to all areas that are set for SRAM, and are also effective for write cycles in the areas that are set for burst ROM.

At cold start, SWAITE is set to "0" (disabled). At hot start, SWAITE retains its status before being initialized.

**A3EEN: Area 3 emulation (DB) / DRAM timing set-up register (0x48130)**

Leave this bit unchanged ("0").

**CEFUNC1–CEFUNC0: #CE pin function selection (D[A:9]) / DRAM timing set-up register (0x48130)**

Change the #CE pin-assigned area.

Table II.4.16 #CE Output Assignment

Pin	CEFUNC = "00"	CEFUNC = "01"	CEFUNC = "1x"
#CE4	#CE4	#CE11	#CE11+#CE12
#CE5	#CE5	#CE15	#CE15+#CE16
#CE6	#CE6	#CE6	#CE7+#CE8
#CE7/#RAS0	#CE7/#RAS0	#CE13/#RAS2	#CE13/#RAS2
#CE8/#RAS1	#CE8/#RAS1	#CE14/#RAS3	#CE14/#RAS3
#CE9	#CE9	#CE17	#CE17+#CE18
#CE10EX	#CE10EX	#CE10EX	#CE9+#CE10EX

(Default: CEFUNC = "00")

The high-order areas that are made available for use by writing "01" to CEFUNC can be larger in size than the default low-order areas. For example, when using in default settings, the available space is 4 MB in areas 7 and 8.

However, if areas 13 and 14 are used, up to 32 MB can be used. The same applies to the other areas.

Furthermore, when CEFUNC is set to "10" or "11", four chip enable signal is expanded into two area size.

At cold start, CEFUNC is set to "00". At hot start, CEFUNC retains its status before being initialized.

## II-4 CORE BLOCK: BCU (Bus Control Unit)

---

**CRAS:** Successive RAS mode (D8) / DRAM timing set-up register (0x48130)

Leave this bit unchanged ("0").

---

**RPRC1–RPRC0:** Number of RAS precharge cycles (D[7:6]) / DRAM timing set-up register (0x48130)

Leave these bits unchanged ("0").

---

**CASC1–CASC0:** Number of CAS cycles (D[4:3]) / DRAM timing set-up register (0x48130)

Leave these bits unchanged ("0").

---

**RASC1–RASC0:** Number of RAS cycles (D[1:0]) / DRAM timing set-up register (0x48130)

Leave these bits unchanged ("0").

---

**A18IO:** Areas 18–17 internal/external access selection (DF) / Access control register (0x48132)

**A16IO:** Areas 16–15 internal/external access selection (DE) / Access control register (0x48132)

**A14IO:** Areas 14–13 internal/external access selection (DD) / Access control register (0x48132)

**A12IO:** Areas 12–11 internal/external access selection (DC) / Access control register (0x48132)

**A8IO:** Areas 8–7 internal/external access selection (DA) / Access control register (0x48132)

**A6IO:** Area 6 internal/external access selection (D9) / Access control register (0x48132)

**A5IO:** Areas 5–4 internal/external access selection (D8) / Access control register (0x48132)

Select either internal access or external access for each area.

Write "1": Internal access

Write "0": External access

Read: Valid

When AxxIO is set to "1", the internal device that mapped to the corresponding area is accessed. When AxxIO is set to "0", the external device is accessed.

At cold start, these bits are set to "0" (external access). At hot start, these bits retain their status before being initialized.

---

**A18EC:** Areas 18–17 little/big endian method selection (D7) / Access control register (0x48132)

**A16EC:** Areas 16–15 little/big endian method selection (D6) / Access control register (0x48132)

**A14EC:** Areas 14–13 little/big endian method selection (D5) / Access control register (0x48132)

**A12EC:** Areas 12–11 little/big endian method selection (D4) / Access control register (0x48132)

**A10EC:** Areas 10–9 little/big endian method selection (D3) / Access control register (0x48132)

**A8EC:** Areas 8–7 little/big endian method selection (D2) / Access control register (0x48132)

**A6EC:** Area 6 little/big endian method selection (D1) / Access control register (0x48132)

**A5EC:** Areas 5–4 little/big endian method selection (D0) / Access control register (0x48132)

Select either little endian or big-endian method for accessing each area.

Write "1": Big-endian

Write "0": Little-endian

Read: Valid

When AxxEC is set to "1", the corresponding area is accessed in big-endian method. When AxxEC is set to "0", the area is accessed in little-endian method. When using area 10 as the boot area, fix A10EC at "0" (little-endian).

At cold start, these bits are set to "0" (little-endian). At hot start, these bits retain their status before being initialized.



---

**A18AS:** Areas 18–17 address strobe signal (DF) / G/A read signal control register (0x48138)

**A16AS:** Areas 16–15 address strobe signal (DE) / G/A read signal control register (0x48138)

**A14AS:** Areas 14–13 address strobe signal (DD) / G/A read signal control register (0x48138)

**A12AS:** Areas 12–11 address strobe signal (DC) / G/A read signal control register (0x48138)

**A8AS:** Areas 8–7 address strobe signal (DA) / G/A read signal control register (0x48138)

**A6AS:** Area 6 address strobe signal (D9) / G/A read signal control register (0x48138)

**A5AS:** Areas 5–4 address strobe signal (D8) / G/A read signal control register (0x48138)

---

Enable/disable the exclusive address strobe signal output.

Write "1": Enabled

Write "0": Disabled

Read: Valid

If AxxAS is set to "1", the exclusive address strobe signal is output from #GAAS (P21) pin when the corresponding area is accessed. If AxxAS is set to "0", the signal output is disabled.

At cold start, these bits are set to "0" (disabled). At hot start, these bits retain their status before being initialized.

---

**A18RD:** Areas 18–17 read signal (D7) / G/A read signal control register (0x48138)

**A16RD:** Areas 16–15 read signal (D6) / G/A read signal control register (0x48138)

**A14RD:** Areas 14–13 read signal (D5) / G/A read signal control register (0x48138)

**A12RD:** Areas 12–11 read signal (D4) / G/A read signal control register (0x48138)

**A8RD:** Areas 8–7 read signal (D2) / G/A read signal control register (0x48138)

**A6RD:** Area 6 read signal (D1) / G/A read signal control register (0x48138)

**A5RD:** Areas 5–4 read signal (D0) / G/A read signal control register (0x48138)

---

Enable/disable the exclusive read signal output.

Write "1": Enabled

Write "0": Disabled

Read: Valid

If AxxRD is set to "1", the exclusive read signal is output from #GARD (P31) pin when the corresponding area is read. If AxxRD is set to "0", the signal output is disabled.

At cold start, these bits are set to "0" (disabled). At hot start, these bits retain their status before being initialized.

---

**BCLKSEL1–BCLKSELO:** BCLK output clock selection (D[1:0]) / BCLK select register (0x4813A)

---

Select a clock to be output from the BCLK pin.

Table II.4.17 Selection of BCLK Output Clock

BCLKSEL1	BCLKSELO	Output clock
1	1	PLL_CLK (PLL output clock)
1	0	OSC3_CLK (OSC3 oscillation clock)
0	1	BUS_CLK (Bus clock)
0	0	CPU_CLK (CPU operating clock)

**PLL\_CLK:** PLL output clock. This clock is stable and kept as output except in the following cases:

1. When the PLL is off by setting the PLLS0 pin.
  2. When the OSC3 (high-speed) oscillation is stopped by executing the SLP instruction.
  3. When the OSC3 (high-speed) oscillation is stopped using the CLG register.
- Note that the PLL\_CLK clock is out of phase with the CPU operating clock.

**OSC3\_CLK:** OSC3 (high-speed) oscillation circuit output clock. This clock is stable and kept as output except in the following cases:

1. When the OSC3 (high-speed) oscillation is stopped by executing the SLP instruction.
  2. When the OSC3 (high-speed) oscillation is stopped using the CLG register.
- Note that the OSC3\_CLK clock is out of phase with the CPU operating clock.

## II-4 CORE BLOCK: BCU (Bus Control Unit)

**BUS\_CLK:** Bus clock in the bus controller. This clock varies according to the bus cycle speed. Furthermore, the clock frequency changes dynamically in x2 speed mode as follows:

1. When the internal RAM/ROM is accessed, x2 clock (e.g., 50 MHz same as the CPU operating clock) is output.
2. When an external device is accessed via the external bus, x1 clock (e.g., 25 MHz) is output. This dynamic change (e.g., between 50 MHz and 25 MHz) does not affect the external memory access timing, such as position relationship between the rising or falling edge of the 25 MHz clock and the falling edge of the #WR signal. (It is the same as that in the x1 speed mode with 25 MHz clock.)

**CPU\_CLK:** The CPU operating clock. The clock frequency is as follows:

1. Equals to the PLL output clock frequency when the PLL is on.
2. Equals to the OSC3 (high-speed) oscillation circuit output clock frequency when the PLL is off.
3. However, it equals to the divided frequency when the CLG is set to generate the CPU operating clock by dividing the source clock.
4. When the CPU stops by the HALT or SLP instruction, this clock is also stopped.

This clock is almost in phase with the bus clock.

At initial reset, BCLKSEL is set to "00" (CPU\_CLK).

---

### **A1X1MD:** Area 1 access speed (D3) / BCLK select register (0x4813A)

---

Select a number of access cycles for area 1 in x2 speed mode.

Write "1": 2 cycles

Write "0": 4 cycles (Setting "0" is not allowed in x2 speed mode)

Read: Valid

When x2 speed mode is set (#X2SPD pin = "0") and A1X1MD = "1", area 1 is read/written in 2 cycles of the CPU system clock. A1X1MD must be set to "1" when using the chip in x2 speed mode.

When x1 speed mode is set (#X2SPD pin = "1"), area 1 is always accessed in 2 cycles regardless of the A1X1MD value.

At cold start, A1X1MD is set to "0" (4 cycles). At hot start, A1X1MD retains its status before being initialized.

---

### **A18RH:** Areas 18–17 read hold cycle insertion (D7) / Read cycle hold time control register (0x4813C)

**A16RH:** Areas 16–15 read hold cycle insertion (D6) / Read cycle hold time control register (0x4813C)

**A14RH:** Areas 14–13 read hold cycle insertion (D5) / Read cycle hold time control register (0x4813C)

**A12RH:** Areas 12–11 read hold cycle insertion (D4) / Read cycle hold time control register (0x4813C)

**A10RH:** Areas 10–9 read hold cycle insertion (D3) / Read cycle hold time control register (0x4813C)

**A8RH:** Areas 8–7 read hold cycle insertion (D2) / Read cycle hold time control register (0x4813C)

**A6RH:** Area 6 read hold cycle insertion (D1) / Read cycle hold time control register (0x4813C)

**A5RH:** Areas 5–4 read hold cycle insertion (D0) / Read cycle hold time control register (0x4813C)

---

Select whether or not a read hold cycle is inserted in each area when accessed.

Write "1": Read hold cycle inserted

Write "0": No read hold cycle inserted

Read: Valid

When AxxRH is set to "1", the read cycle access time of the corresponding area is extended by 1 cycle. However, the length of the #RD pulse and the timing do not change; only #CE and address signals are lengthened by 1 cycle. When read hold cycle insertion is specified, a read hold cycle is always inserted after a read cycle to the specified area (see Figure II.4.23). Also, if an output disable cycle is set up, after the read hold cycle, an output disable cycle is also inserted.

At cold start, AxxRH is set to "0" (No read hold cycle inserted). At hot start, AxxRH retains its status before being initialized.

---

**A18BS:** Areas 18–17 bus speed setting (D7) / Bus speed set-up register (0x4813E)  
**A16BS:** Areas 16–15 bus speed setting (D6) / Bus speed set-up register (0x4813E)  
**A14BS:** Areas 14–13 bus speed setting (D5) / Bus speed set-up register (0x4813E)  
**A12BS:** Areas 12–11 bus speed setting (D4) / Bus speed set-up register (0x4813E)  
**A10BS:** Areas 10–9 bus speed setting (D3) / Bus speed set-up register (0x4813E)  
**A8BS:** Areas 8–7 bus speed setting (D2) / Bus speed set-up register (0x4813E)  
**A6BS:** Area 6 bus speed setting (D1) / Bus speed set-up register (0x4813E)  
**A5BS:** Areas 5–4 bus speed setting (D0) / Bus speed set-up register (0x4813E)

---

Sets the bus speed mode for the corresponding areas.

Write "1": x1 speed mode

Write "0": #X2SPD pin setting

Read: Valid

When #X2SPD = "0" (x2 speed mode), it is possible to switch x2 speed mode to x1 speed mode for each area. Since this setting is independent of the access control register upper byte, which switches each area between internal and external, each area can be set to x1 or x2 speed and to internal or external arbitrarily.

At cold start, AxxBS is set to "0" (#X2SPD pin setting). At hot start, AxxBS retains its status before being initialized.

**Note:** The bus speed including the internal bus is limited, up to 40 MHz. When running the CPU with a 40 MHz or more operating clock, set the #X2SPD pin to "0", AxxBS to "0", and A1X1MD to "1".

## ***Programming Notes***

---

- (1) A1X1MD (D3/0x4813A) must be set to "1" when using the chip in x2 speed mode (#X2SPD pin = "0").
- (2) The bus speed including the internal bus is limited, up to 40 MHz. When running the CPU with a 40 MHz or more operating clock, set the #X2SPD pin to "0", AxxBS (0x4813E) to "0", and A1X1MD (D3/0x4813A) to "1".

## II-5 ITC (Interrupt Controller)

The C33 Core Block contains an interrupt controller, making it possible to control all interrupts generated by the internal peripheral circuits. This section explains the functions of this interrupt controller centering around the method for controlling maskable interrupts. For details about the various factors and conditions under which interrupts are generated, refer to the description of each peripheral circuit in this manual.

### Outline of Interrupt Functions

#### Maskable Interrupts

The ITC can handle 50 kinds of maskable interrupts. Table II.5.1 shows the trap table in the C33 core.

Table II.5.1 Trap Table

Vector number (Hex address)	Exception/interrupt name (peripheral circuit)	Exception/interrupt factor	IDMA Ch.	Priority
0(Base)	Reset	Low input to the reset pin	–	High ↑
1–3	reserved	–	–	
4(Base+10)	Zero division	Division instruction	–	
5	reserved	–	–	
6(Base+18)	Address error exception	Memory access instruction	–	
0x0 or 0x60000	Debugging exception	brk instruction, etc.	–	
8(Base+1C)	NMI	Low input to the NMI pin	–	
9–11	reserved	–	–	
12(Base+30)	Software exception 0	int instruction	–	
13(Base+34)	Software exception 1	int instruction	–	
14(Base+38)	Software exception 2	int instruction	–	
15(Base+3C)	Software exception 3	int instruction	–	
16(Base+40)	Port input interrupt 0	Edge (rising or falling) or level (High or Low)	1	
17(Base+44)	Port input interrupt 1	Edge (rising or falling) or level (High or Low)	2	
18(Base+48)	Port input interrupt 2	Edge (rising or falling) or level (High or Low)	3	
19(Base+4C)	Port input interrupt 3	Edge (rising or falling) or level (High or Low)	4	
20(Base+50)	Key input interrupt 0	Rising or falling edge	–	
21(Base+54)	Key input interrupt 1	Rising or falling edge	–	
22(Base+58)	High-speed DMA Ch.0	High-speed DMA Ch.0, end of transfer	5	
23(Base+5C)	High-speed DMA Ch.1	High-speed DMA Ch.1, end of transfer	6	
24(Base+60)	High-speed DMA Ch.2	High-speed DMA Ch.2, end of transfer	–	
25(Base+64)	High-speed DMA Ch.3	High-speed DMA Ch.3, end of transfer	–	
26(Base+68)	IDMA	Intelligent DMA, end of transfer	–	
27–29	reserved	–	–	
30(Base+78)	16-bit programmable timer 0	Timer 0 comparison B	7	
31(Base+7C)		Timer 0 comparison A	8	
32–33	reserved	–	–	
34(Base+88)	16-bit programmable timer 1	Timer 1 comparison B	9	
35(Base+8C)		Timer 1 comparison A	10	
36–37	reserved	–	–	
38(Base+98)	16-bit programmable timer 2	Timer 2 comparison B	11	
39(Base+9C)		Timer 2 comparison A	12	
40–41	reserved	–	–	
42(Base+A8)	16-bit programmable timer 3	Timer 3 comparison B	13	
43(Base+AC)		Timer 3 comparison A	14	
44–45	reserved	–	–	
46(Base+B8)	16-bit programmable timer 4	Timer 4 comparison B	15	
47(Base+BC)		Timer 4 comparison A	16	
48–49	reserved	–	–	
50(Base+C8)	16-bit programmable timer 5	Timer 5 comparison B	17	
51(Base+CC)		Timer 5 comparison A	18	
52(Base+D0)	8-bit programmable timer 0–3	Timer 0 underflow	19	
53(Base+D4)		Timer 1 underflow	20	
54(Base+D8)		Timer 2 underflow	21	
55(Base+DC)		Timer 3 underflow	22	
56(Base+E0)	Serial interface Ch.0	Receive error	–	
57(Base+E4)		Receive buffer full	23	
58(Base+E8)		Transmit buffer empty	24	
59	reserved	–	–	Low ↓

## II-5 CORE BLOCK: ITC (Interrupt Controller)

Vector number (Hex address)	Exception/interrupt name (peripheral circuit)	Exception/interrupt factor	IDMA Ch.	Priority
60(Base+F0)	Serial interface Ch.1	Receive error	—	High ↑
61(Base+F4)		Receive buffer full	25	
62(Base+F8)		Transmit buffer empty	26	
63	reserved	—	—	
64(Base+100)	A/D converter	A/D converter, end of conversion	27	
65(Base+104)	Clock timer	Falling edge of 32 Hz, 8 Hz, 2 Hz or 1 Hz signal 1-minute, 1-hour or specified time count up	—	
66–67	reserved	—	—	
68(Base+110)	Port input interrupt 4	Edge (rising or falling) or level (High or Low)	28	
69(Base+114)	Port input interrupt 5	Edge (rising or falling) or level (High or Low)	29	
70(Base+118)	Port input interrupt 6	Edge (rising or falling) or level (High or Low)	30	
71(Base+11C)	Port input interrupt 7	Edge (rising or falling) or level (High or Low)	31	
72(Base+120)	8-bit programmable timer 4/5	Timer 4 underflow	32	
73(Base+124)		Timer 5 underflow	33	
74–75	reserved	—	—	
76(Base+130)	Serial interface Ch.2	Receive error	—	
77(Base+134)		Receive buffer full	34	
78(Base+138)		Transmit buffer empty	35	
79	reserved	—	—	
80(Base+140)	Serial interface Ch.3	Receive error	—	
81(Base+144)		Receive buffer full	36	
82(Base+148)		Transmit buffer empty	37	
83–111	reserved	—	—	
112(Base+1C0)	Serial interface with FIFO Ch.0	Receive error	—	
113(Base+1C4)		Receive buffer full	56	↓ Low
114(Base+1C8)		Transmit buffer empty	57	

### Contents of table

"Vector number (Address)" indicates the trap table's vector number. The numerals in parentheses show an offset (in bytes) from the starting address (Base) of the trap table. The starting address (Base) of the trap table by default is the boot address, 0xC00000 set at an initial reset. This address can be changed using the TTBR register (0x48134 to 0x48137).

"Exception/interrupt name (peripheral circuit)" indicates that interrupt levels can be programmed for each peripheral circuit written.

"Exception/interrupt factor" indicates the factor of the interrupt occurring in each interrupt system.

"IDMA Ch." indicates that an interrupt factor which has a numeric value in this column can start up the intelligent DMA (IDMA) to transfer data when an interrupt factor occurs. The numeric value indicates the IDMA's channel number. Interrupt factors that do not have a numeric value here cannot start up the IDMA.

"Priority" indicates the priority of interrupts in cases when all interrupt systems are set to the same interrupt level. If two or more interrupt factors occur simultaneously, interrupt requests are accepted in order of highest priority. Interrupt priority varies depending on the interrupt levels set in each interrupt system. However, the priorities of interrupt factors in the same interrupt system are fixed in the order that they are written here.

### Maskable interrupt generating conditions

A maskable interrupt to the CPU occurs when all of the conditions described below are met.

- The interrupt enable register for the interrupt factor that has occurred is set to "1".
- The IE (Interrupt Enable) bit of the Processor Status Register (PSR) in the CPU is set to "1".
- The interrupt factor that has occurred has a higher priority level than the value that is set in the PSR's Interrupt Level (IL). (The interrupt levels can be set using the interrupt priority register in each interrupt system.)
- No other trap factor having higher priority, such as NMI, has occurred.
- The interrupt factor does not invoke IDMA (the IDMA request bit is set to "0").

When an interrupt factor occurs, the corresponding interrupt factor flag is set to "1" and the flag remains set until it is reset in the software program. Therefore, in no cases can the generated interrupt factor be inadvertently cleared even if the above conditions are not met when the interrupt factor has occurred. The interrupt will occur when the above conditions are met.

However, when the interrupt factor invokes IDMA, the interrupt factor is reset if the following condition is met.

- The IDMA transfer counter is not "0".
- Interrupts are disabled in the IDMA control information even if the transfer counter is "0".

If two or more maskable interrupt factors occur simultaneously, the interrupt factor that has the highest priority is allowed to signal an interrupt request to the CPU. The other interrupts with lower priorities are kept pending until the above conditions are met.

The PSR and interrupt control register will be detailed later.

For details about interrupt factor generating conditions, refer to the description of each peripheral circuit in this manual.

II

## ***Interrupt Factors and Intelligent DMA***

Several interrupt factors can be set so that they can invoke IDMA startup. When one of these interrupt factors occurs, IDMA is started up before an interrupt request to the CPU. The interrupt request to the CPU is generated after IDMA is completed. (The interrupt request can be disabled by a program.)

IDMA is always started up regardless of how the PSR is set. For details, refer to "IDMA Invocation".

ITC

## ***Nonmaskable Interrupt (NMI)***

The nonmaskable interrupt (NMI) can be generated by pulling the #NMI pin low or using the internal watchdog timer. The vector number of NMI is 7, with the vector address set to the trap table's starting address + 28 bytes.

This interrupt is prioritized over other interrupts and is unconditionally accepted by the CPU.

However, since this interrupt may operate erratically if it occurs before the stack pointer (SP) is set up, it is masked in hardware until a write to the SP is completed after an initial reset.

## ***Interrupt Processing by the CPU***

The CPU keeps sampling interrupt requests every cycle. When the CPU accepts an interrupt request, it enters trap processing after completing execution of the instruction that was being executed.

The following lists the contents executed in trap processing.

- (1) The PSR and the current program counter (PC) value are saved to the stack.
- (2) The IE bit of the PSR is reset to "0" (following maskable interrupts are disabled).
- (3) The IL of the PSR is set to the priority level of the accepted interrupt (NMI does not have its interrupt level changed).
- (4) The vector of the generated interrupt factor is loaded into the PC, thus executing the interrupt processing routine.

Thus, once an interrupt is accepted, all maskable interrupts that may follow are disabled in (2). Multiple interrupts can also be handled by setting the IE bit to "1" in the interrupt processing routine. In this case, since the IL has been changed in (3), only an interrupt that has a higher priority than that of the currently processed interrupt is accepted. When the interrupt processing routine is terminated by the `reti` instruction, the PSR is restored to its previous status before the interrupt has occurred. The program restarts processing after branching to the instruction next to the one that was being executed when the interrupt occurred.

## ***Clearing Standby Mode by Interrupts***

The standby modes (HALT and SLEEP) are cleared by an NMI or a maskable interrupt.

All maskable interrupts can be used to clear HALT mode. However, if the bus clock has stopped in HALT2 mode, a DMA interrupt cannot be used.

In SLEEP mode, since the high-speed (OSC3) oscillation circuit is deactivated, interrupts from the peripheral circuits that operate with the OSC3 clock cannot be used.

Interrupts that can be used to clear basic HALT mode: NMI and all maskable interrupts

Interrupts that can be used to clear HALT2 mode: NMI and all maskable interrupts (except DMA interrupts)

Interrupts that can be used to clear SLEEP mode: NMI, input port interrupts, and clock timer interrupts

Clearing of the standby modes is accomplished by an interrupt request to the CPU. Therefore, this requires that the PSR be set in such a way that the requested interrupt will be accepted, and that the interrupt enable register for the interrupt factor be set to accept the interrupt.

When standby mode is cleared and the CPU has accepted the interrupt, it returns to the instruction next to the halt or slp instruction after executing the interrupt processing routine.

**Note:** If the interrupt factor used to restart from the standby mode has been set to invoke the IDMA, the IDMA is started up by that interrupt.

In the case of SLEEP mode, the high-speed (OSC3) oscillation circuit also starts operating.

If an interrupt to be generated upon completion of IDMA is disabled at the setting of the IDMA side, no interrupt request is signaled to the CPU. Therefore, the CPU remains idle until the next interrupt request is generated.



## Trap Table

---

The C33 Core Block allows the base (starting) address of the trap table to be set by the TTBR register.

TTBR0 (D[9:0]) / TTBR low-order register (0x48134): Trap table base address [9:0] (fixed at "0")

TTBR1 (D[F:A]) / TTBR low-order register (0x48134): Trap table base address [15:10]

TTBR2 (D[B:0]) / TTBR high-order register (0x48136): Trap table base address [27:16]

TTBR3 (D[F:C]) / TTBR high-order register (0x48136): Trap table base address [31:28] (fixed at "0")

After an initial reset, the TTBR register is set to 0x0C00000.

Therefore, even when the trap table position is changed, it is necessary that at least the reset vector be written to the above address.

TTBR0 and TTBR3 are read-only bits which are fixed at "0". Therefore, the trap table starting address always begins with a 1KB boundary address.

The TTBR register is normally write-protected to prevent them from being inadvertently rewritten. To remove this write protection function, another register, TBRP (D[7:0]) / TTBR write-protect register (0x4812D [byte]), is provided. A write to the TTBR register is enabled by writing "0x59" to TBRP and is disabled back again by a write to the most significant byte of the TTBR register (0x48137). Consequently, a write to the TTBR register needs to begin with the low-order half-word first. However, since an occurrence of NMI or the like between writes of the low-order and high-order half-words would cause a malfunction, it is recommended that the register be written in words.

II

ITC

## Control of Maskable Interrupts

### Structure of the Interrupt Controller

The interrupt controller is configured as shown in Figure II.5.1.

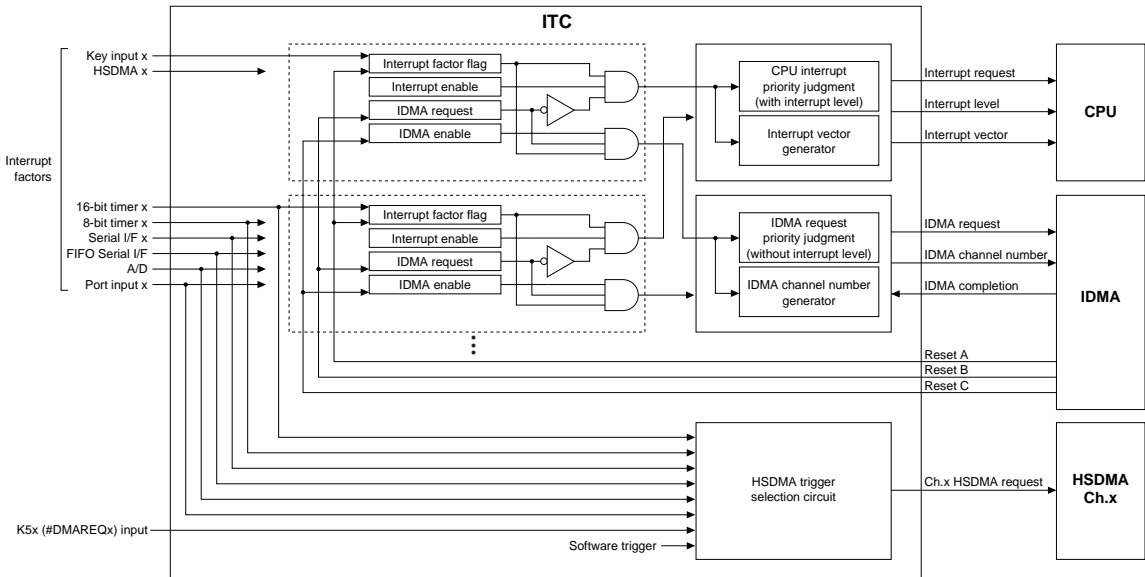


Figure II.5.1 Configuration of Interrupt Controller

The following sections explain the functions of the registers used to control interrupts.

### Processor Status Register (PSR)

The PSR is a special register incorporated in the core CPU and contains control bits to enable or disable an interrupt request to the CPU.

#### Interrupt Enable (IE) bit: PSR[4]

This bit is used to enable or disable an interrupt request to the CPU. When this bit is set to "1", the CPU is enabled to accept a maskable interrupt request. When this bit is reset to "0", no maskable interrupt request is accepted by the CPU.

When the CPU accepts an interrupt request (or some other trap occurs), it saves the PSR to the stack and resets the IE bit to "0". Consequently, no maskable interrupt request occurring thereafter will be accepted unless the IE bit is set to "1" in software program or the interrupt (trap) processing routine is terminated by the reti instruction.

The IE bit is initialized to "0" (interrupts disabled) by an initial reset.

#### Interrupt Level (IL): PSR[11:8]

The IL bits disable the interrupts whose priorities are below the set interrupt level. For example, if the interrupt level set in the IL is 3, the interrupts whose priorities are set below 3 in the interrupt priority register (described later) are not accepted by the CPU even if the IE bit is set to "1". The IL and the interrupt priority register together allow you to control the interrupt priorities in each interrupt system. For details about the interrupt levels, refer to "Interrupt Priority Register and Interrupt Levels".

When the CPU accepts a maskable interrupt request, it saves the PSR to the stack and sets the IL to the accepted interrupt's priority level. Therefore, even when the IE bit is set to "1" in the interrupt processing routine, no interrupts whose priority levels are equal or below that of the interrupt currently being processed are accepted unless the IL is rewritten.

The IL is restored to its previous status when the interrupt processing routine is terminated by the reti instruction.

The IL is rewritten for only maskable interrupts and not for any other traps (except a reset).

The IL is set to level 0 (that is, all interrupts above level 1 are enabled) by an initial reset.

**Note:** As the C33 STD Core CPU function, the IL allows interrupt levels to be set in the range of 0 to 15. However, since the interrupt priority register in the ITC consists of three bits, interrupt levels in each interrupt system can only be set for up to 8.

## ***Interrupt Factor Flag and Interrupt Enable Register***

An interrupt factor flag and an interrupt enable register are provided for each maskable interrupt factor.

### **Interrupt factor flag**

The interrupt factor flag is set to "1" when the corresponding interrupt factor occurs. Reading the flag enables you to determine what caused an interrupt, making it unnecessary to resort to the CPU's trap processing. The interrupt factor flag is reset by writing data in software. Note that the method by which this flag is reset can be selected from the software application using either of the two methods described below. This selection is accomplished using RSTONLY (D0) / Flag set/reset method select register (0x4029F).

- **Reset-only method (default)**

This method is selected (RSTONLY = "1") when initially reset.

With this method, the interrupt factor flag is reset by writing "1". Although multiple interrupt factor flags are located at the same address of the interrupt control register, the interrupt factor flags for which "0" has been written can be neither set nor reset. Therefore, this method ensures that only a specific factor flag is reset. However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that an interrupt factor flag that has been set to "1" is reset by writing.

In this method, no interrupt factor flag can be set in the software application.

- **Read/write method**

This method is selected by writing "0" to RSTONLY.

When this method is used, interrupt factor flags can be read and written as for other registers. Therefore, the flag is reset by writing "0" and set by writing "1". In this case, all factor flags for which "0" has been written are reset. Even in a read-modify-write operation, an interrupt factor can occur between the read and the write, so be careful when using this method.

Since interrupt factor flags are not initialized by an initial reset, be sure to reset them before enabling interrupts.

**Note:** Even when a maskable interrupt request is accepted by the CPU and control branches off to the interrupt processing routine, the interrupt factor flag is not reset. Consequently, if control is returned from the interrupt processing routine by the reti instruction without resetting the interrupt factor flag in a program, the same interrupt factor occurs again.

For details about interrupt factor generating conditions, refer to the description of each peripheral circuit in this manual.

### Interrupt enable register

This register controls the output of an interrupt request to the CPU. Only when the interrupt enable bit of this register is set to "1" can an interrupt request to the CPU be enabled by an occurrence of the corresponding interrupt factor. If the bit is set to "0", no interrupt request is made to the CPU even when the corresponding interrupt factor occurs.

Interrupt enable bits can be read and written as for other registers. Therefore, the interrupt enable bit is reset by writing "0" and set by writing "1". By reading this register, its setup status can be checked at any time.

Settings of the interrupt enable register do not affect the operation of interrupt factor flags, so when an interrupt factor occurs the interrupt factor flag is set to "1" even if the corresponding interrupt enable bit is set to "0".

When initially reset, the interrupt enable register is set to "0" (interrupts are disabled).

In cases when IDMA is started up by occurrence of an interrupt factor or when clearing standby mode (HALT or SLEEP mode) too, the corresponding interrupt enable bit must be set to "1".

The interrupt controller outputs an interrupt request to the CPU when the following conditions are met:

- An interrupt factor has occurred and the interrupt factor flag is set to "1".
- The bit of the interrupt enable register for the interrupt factor that has occurred is set to "1" (interrupt enable).
- The bit of the IDMA request register for the interrupt factor that has occurred is set to "0" (interrupt request).

If two or more interrupt factors occur simultaneously, the interrupt factor that has the highest priority is allowed to signal an interrupt request to the CPU. (See the following section.)

When these conditions are met, the interrupt controller outputs an interrupt request signal to the CPU along with the setup content (interrupt level) of the interrupt priority register for the generated interrupt system and its vector number.

These signals remain asserted until the interrupt factor flag is reset to "0" or the corresponding bit of the interrupt enable register is set to "0" (interrupts are disabled) or until some other interrupt factor of higher priority occurs.

They are not cleared if the CPU simply accepts the interrupt request.

## Interrupt Priority Register and Interrupt Levels

The interrupt priority register is a 3-bit register provided for each interrupt system. It allows the interrupt levels of a given interrupt system to be set in the range of 0 to 7. The default priorities shown in Table II.5.1 can be modified according to system requirements by this setting.

The value set in this register is used by the interrupt controller and the CPU as described below.

### Roles of the interrupt priority register in the interrupt controller

If two or more interrupt factors that have been enabled by the interrupt enable register occur simultaneously, the interrupt factor in the interrupt system whose interrupt priority register contains the greatest value is allowed by the interrupt controller to signal an interrupt request to the CPU.

If an interrupt factor occurs in two or more interrupt systems having the same value, the interrupt priority is resolved according to the default priorities in Table II.5.1. Interrupt factors in the same interrupt system also have their priorities resolved according to the order in Table II.5.1.

Other interrupt factors are kept pending until all interrupts of higher priority are accepted by the CPU.

When outputting an interrupt request signal to the CPU, the interrupt controller outputs the content of the interrupt priority register to the CPU along with it.

If another interrupt factor of higher priority occurs during outputting an interrupt request signal, the interrupt controller changes the vector number and interrupt level to those of the new interrupt factor before they are output to the CPU. The first interrupt request is left pending.

### Roles of the interrupt priority register in CPU processing

The CPU compares the content of the interrupt priority register received from the interrupt controller with the interrupt level that is set in the IL of the PSR to determine whether or not to accept the interrupt request.

IE bit = "1" &  $IL < \text{interrupt priority register}$ : the interrupt request is accepted

IE bit = "1" &  $IL \geq \text{interrupt priority register}$ : the interrupt request is rejected

Before interrupts can be controlled by an interrupt level, the interrupt disabling level must be written to the IL. For example, if the value written to the IL is 3, only the interrupts whose interrupt levels written in the interrupt priority register are 4 or more will be accepted.

When an interrupt is accepted, the interrupt level that is set in its interrupt priority register is written to the IL.

As a result, the interrupt requests below that interrupt level can no longer be accepted.

If the interrupt priority register for an interrupt is set to "0", the interrupt is disabled. However, invoking IDMA by means of an interrupt factor works fine.

- Notes:**
- As the C33 STD Core CPU function, the IL allows interrupt levels to be set in the range of 0 to 15. However, since the interrupt priority register in the C33 Core Block consists of three bits, interrupt levels in each interrupt system can only be set for up to 8.
  - Multiple interrupts can also be handled by rewriting the interrupt level to the IL in the interrupt processing routine. However, if the interrupt level of the IL is set below the current level and the IE is set to enable interrupts before resetting the interrupt factor flag after an interrupt has occurred, the same interrupt may occur again.

## **IDMA Invocation**

---

The interrupt factors for which IDMA channel numbers are written in Table II.5.1 have the function to invoke the intelligent DMA (IDMA).

### **IDMA request register**

The IDMA request register is used to specify the interrupt factor that invoke an IDMA transfer. If an IDMA request bit is set to "1", the IDMA request will be generated when the corresponding interrupt factor occurs. When the IDMA request bit is set to "0", the corresponding interrupt factor does not invoke IDMA and a normal interrupt processing will be performed. The IDMA request register is set to "0" by an initial reset. The method by which this register is set can be selected from the software application using either of the two methods described below. This selection is accomplished using IDMAONLY (D1) / Flag set/reset method select register (0x4029F).

- **Set-only method (default)**

This method is selected (IDMAONLY = "1") when initially reset.

With this method, an IDMA request bit is set by writing "1". Although multiple IDMA request bits are located in the IDMA request register, the IDMA request bits for which "0" has been written can be neither set nor reset. Therefore, this method ensures that only a specific IDMA request bit is set.

However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that an IDMA request bit that has been set to "1" is not reset by writing.

- **Read/write method**

This method is selected by writing "0" to IDMAONLY.

When this method is used, IDMA request bits can be read and written as for other registers. Therefore, the IDMA request bit is reset by writing "0" and set by writing "1". In this case, all IDMA request bits for which "0" has been written are reset. Even in a read-modify-write operation, an IDMA request bit can be reset by the hardware between the read and the write, so be careful when using this method.

### **IDMA enable register**

To perform IDMA transfer using an interrupt factor, the corresponding bit of the IDMA enable register must be set to "1". If this bit is set to "0", the interrupt factor cannot invoke the IDMA channel. The IDMA enable register is set to "0" by an initial reset.

The IDMA enable register allows selection of a set method (set-only method or Read/write method) similar to the IDMA request register. This selection is accomplished using DENONLY (D2) / Flag set/reset method select register (0x4029F). See the above explanation for the set method.

### **Invoking IDMA**

Before IDMA can be invoked by the occurrence of an interrupt factor, the corresponding bits of the IDMA request and IDMA enable registers must be set to "1". Then when an interrupt factor occurs, the interrupt request to the CPU is made pending and the corresponding IDMA channel is invoked. The DMA transfer is performed according to the control information of that IDMA channel. The interrupt level set by the interrupt priority register of the ITC does not affect the IDMA invocation. The IDMA request can be accepted even if the interrupt level of the CPU is higher than the set value of the interrupt priority register. However, when generating the interrupt request to the CPU after the IDMA transfer is completed, the interrupt is controlled using the interrupt level set by the interrupt priority register.

An IDMA invocation request is accepted even when the interrupt enable register and PSR of the CPU is set to disable interrupts. It is also necessary that the control information for the IDMA channel has been set.

## Interrupt after IDMA transfer

### To generate an interrupt after completion of IDMA transfer:

The interrupt request that has been kept pending can be generated after completion of the DMA transfer.

In this case, the interrupt must be enabled by the IDMA control information (DINTEN = "1") in addition to the interrupt controller and the PSR register settings.

However, if the transfer counter set for the selected IDMA channel does not reach the terminal count of 0 after the number of transfers set have been performed, the interrupt factor flag is reset and no interrupt request is generated. The transfer counter is decremented by 1 for each transfer performed.

If the transfer counter is decremented to 0 when DINTEN is set to "1", the interrupt factor flag is not reset and the IDMA request bit is cleared to "0". An interrupt request is generated if other interrupt conditions are met.

The IDMA request bit must be set up again in order for IDMA to be invoked when an interrupt factor occurs next time as well. To ensure that no unwanted IDMA request occurs, this setup must be performed after resetting the interrupt factor flag.

Figure II.5.2 shows the hardware sequence when DINTEN is set to "1".

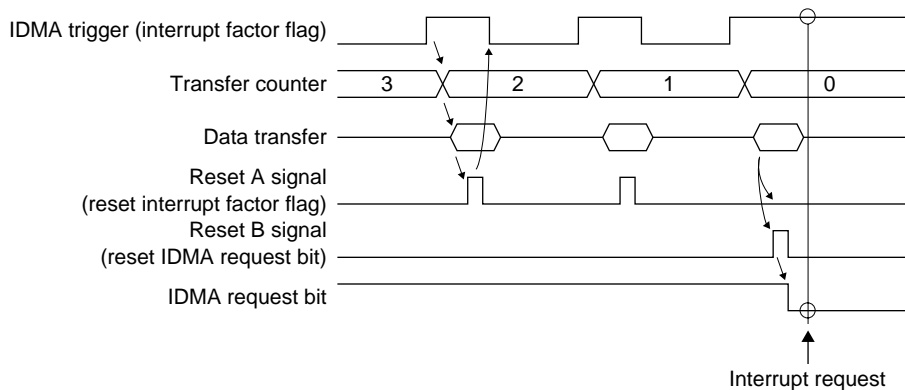


Figure II.5.2 Sequence when DINTEN = "1"

### To disable an interrupt after completion of IDMA transfer:

If an interrupt has been disabled in the IDMA control information (DINTEN = "0"), the interrupt is not generated since the interrupt factor flag is reset when the transfer counter becomes 0.

In this case, the IDMA request bit remains set to "1" without being cleared. However, the IDMA enable bit is cleared, so the following IDMA request by the same interrupt factor will be disabled.

Figure II.5.3 shows the hardware sequence when DINTEN is set to "0".

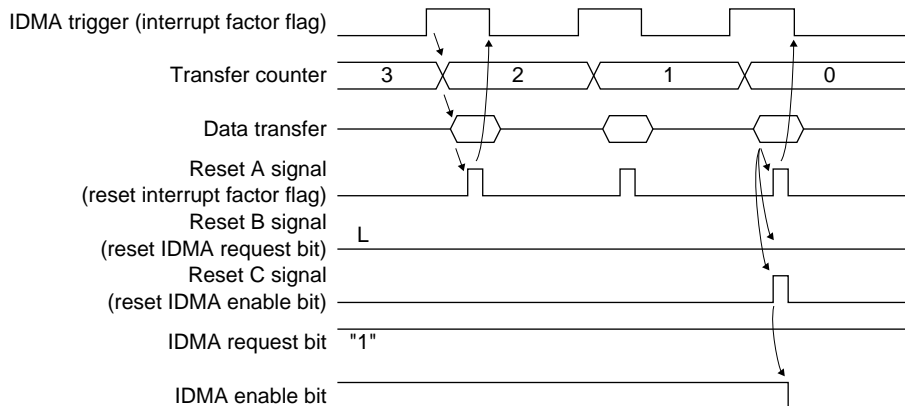


Figure II.5.3 Sequence when DINTEN = "0"

For details on IDMA, refer to "IDMA (Intelligent DMA)".

## HSDMA Invocation

Some interrupt factors can invoke high-speed DMAs (HSDMA).

### HSDMA trigger set-up register

The DMA block contains four channel of HSDMA circuit. Each channel allows selection of an interrupt factor as the trigger. The HSDMA trigger set-up registers are used for this selection.

HSDMA Ch.0: HSD0S[3:0] (D[3:0])/HSDMA Ch.0/1 trigger set-up register (0x40298)

HSDMA Ch.1: HSD1S[3:0] (D[7:4])/HSDMA Ch.0/1 trigger set-up register (0x40298)

HSDMA Ch.2: HSD2S[3:0] (D[3:0])/HSDMA Ch.2/3 trigger set-up register (0x40299)

HSDMA Ch.3: HSD3S[3:0] (D[7:4])/HSDMA Ch.2/3 trigger set-up register (0x40299)

Table II.5.2 shows the setting value and the corresponding trigger factor.

Table II.5.2 HSDMA Trigger Factor

Value	Ch.0 trigger factor	Ch.1 trigger factor	Ch.2 trigger factor	Ch.3 trigger factor
0000	Software trigger	Software trigger	Software trigger	Software trigger
0001	K50 port input (falling edge)	K51 port input (falling edge)	K53 port input (falling edge)	K54 port input (falling edge)
0010	K50 port input (rising edge)	K51 port input (rising edge)	K53 port input (rising edge)	K54 port input (rising edge)
0011	Port 0 input	Port 1 input	Port 2 input	Port 3 input
0100	Port 4 input	Port 5 input	Port 6 input	Port 7 input
0101	8-bit timer 0 underflow	8-bit timer 1 underflow	8-bit timer 2 underflow	8-bit timer 3 underflow
0110	16-bit timer 0 compare B	16-bit timer 1 compare B	16-bit timer 2 compare B	16-bit timer 3 compare B
0111	16-bit timer 0 compare A	16-bit timer 1 compare A	16-bit timer 2 compare A	16-bit timer 3 compare A
1000	16-bit timer 4 compare B	16-bit timer 5 compare B	16-bit timer 4 compare B	16-bit timer 5 compare B
1001	16-bit timer 4 compare A	16-bit timer 5 compare A	16-bit timer 4 compare A	16-bit timer 5 compare A
1010	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full
1011	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty
1100	A/D conversion completion	A/D conversion completion	A/D conversion completion	A/D conversion completion
1101	reserved	FIFO SI/F Ch.0 Rx buffer full	reserved	FIFO SI/F Ch.0 Rx buffer full
1110	reserved	FIFO SI/F Ch.0 Tx buffer empty	reserved	FIFO SI/F Ch.0 Tx buffer empty

### Invoking HSDMA

By selecting an interrupt factor with the HSDMA trigger set-up register, the HSDMA channel is invoked when the selected interrupt factor occurs. The interrupt control bits (interrupt factor flag, interrupt enable register, IDMA request register, interrupt priority register) do not affect this invocation.

Since HSDMA does not reset the interrupt factor flag, an interrupt will occur when the DMA transfer is completed if the interrupt is enabled by ITC.

Before HSDMA can be invoked by the occurrence of an interrupt factor, it is necessary that DMA be enabled on the HSDMA side by setting the control register for HSDMA transfer.

For details about HSDMA, refer to "HSDMA (High-Speed DMA)".



## I/O Memory of Interrupt Controller

Table II.5.3 shows the control bits of the interrupt controller.

Table II.5.3 Control Bits of Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input 0/1 interrupt priority register	0040260 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP1L2	Port input 1 interrupt level	0 to 7	X	R/W	
		D5	PP1L1			X		
		D4	PP1L0			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP0L2	Port input 0 interrupt level	0 to 7	X	R/W	
D1	PP0L1	X						
D0	PP0L0	X						
Port input 2/3 interrupt priority register	0040261 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP3L2	Port input 3 interrupt level	0 to 7	X	R/W	
		D5	PP3L1			X		
		D4	PP3L0			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP2L2	Port input 2 interrupt level	0 to 7	X	R/W	
D1	PP2L1	X						
D0	PP2L0	X						
Key input interrupt priority register	0040262 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PK1L2	Key input 1 interrupt level	0 to 7	X	R/W	
		D5	PK1L1			X		
		D4	PK1L0			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	PK0L2	Key input 0 interrupt level	0 to 7	X	R/W	
D1	PK0L1	X						
D0	PK0L0	X						
High-speed DMA Ch.0/1 interrupt priority register	0040263 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PHSD1L2	High-speed DMA Ch.1 interrupt level	0 to 7	X	R/W	
		D5	PHSD1L1			X		
		D4	PHSD1L0			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	PHSD0L2	High-speed DMA Ch.0 interrupt level	0 to 7	X	R/W	
D1	PHSD0L1	X						
D0	PHSD0L0	X						
High-speed DMA Ch.2/3 interrupt priority register	0040264 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PHSD3L2	High-speed DMA Ch.3 interrupt level	0 to 7	X	R/W	
		D5	PHSD3L1			X		
		D4	PHSD3L0			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	PHSD2L2	High-speed DMA Ch.2 interrupt level	0 to 7	X	R/W	
D1	PHSD2L1	X						
D0	PHSD2L0	X						
IDMA interrupt priority register	0040265 (B)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	PDM2	IDMA interrupt level	0 to 7	X	R/W	
		D1	PDM1			X		
		D0	PDM0			X		
16-bit timer 0/1 interrupt priority register	0040266 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	P16T12	16-bit timer 1 interrupt level	0 to 7	X	R/W	
		D5	P16T11			X		
		D4	P16T10			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	P16T02	16-bit timer 0 interrupt level	0 to 7	X	R/W	
D1	P16T01	X						
D0	P16T00	X						
16-bit timer 2/3 interrupt priority register	0040267 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	P16T32	16-bit timer 3 interrupt level	0 to 7	X	R/W	
		D5	P16T31			X		
		D4	P16T30			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	P16T22	16-bit timer 2 interrupt level	0 to 7	X	R/W	
D1	P16T21	X						
D0	P16T20	X						

II

ITC

**II-5 CORE BLOCK: ITC (Interrupt Controller)**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
16-bit timer 4/5 interrupt priority register	0040268 (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	P16T52	16-bit timer 5 interrupt level	0 to 7	X	R/W			
		D5	P16T51							
		D4	P16T50							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	P16T42	16-bit timer 4 interrupt level	0 to 7	X	R/W			
D1	P16T41									
D0	P16T40									
8-bit timer, serial I/F Ch.0 interrupt priority register	0040269 (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PSIO02	Serial interface Ch.0 interrupt level	0 to 7	X	R/W			
		D5	PSIO01							
		D4	PSIO00							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	P8TM2	8-bit timer 0–5 interrupt level	0 to 7	X	R/W			
D1	P8TM1									
D0	P8TM0									
Serial I/F Ch.1, A/D interrupt priority register	004026A (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PAD2	A/D converter interrupt level	0 to 7	X	R/W			
		D5	PAD1							
		D4	PAD0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PSIO12	Serial interface Ch.1 interrupt level	0 to 7	X	R/W			
D1	PSIO11									
D0	PSIO10									
Clock timer interrupt priority register	004026B (B)	D7–3	–	reserved	–	–	–	Writing 1 not allowed.		
		D2	PCTM2	Clock timer interrupt level	0 to 7	X	R/W			
		D1	PCTM1							
		D0	PCTM0							
Port input 4/5 interrupt priority register	004026C (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PP5L2	Port input 5 interrupt level	0 to 7	X	R/W			
		D5	PP5L1							
		D4	PP5L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PP4L2	Port input 4 interrupt level	0 to 7	X	R/W			
D1	PP4L1									
D0	PP4L0									
Port input 6/7 interrupt priority register	004026D (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PP7L2	Port input 7 interrupt level	0 to 7	X	R/W			
		D5	PP7L1							
		D4	PP7L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PP6L2	Port input 6 interrupt level	0 to 7	X	R/W			
D1	PP6L1									
D0	PP6L0									
Serial I/F Ch.2/3 interrupt priority register	004026E (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PSIO32	Serial interface Ch.3 interrupt level	0 to 7	X	R/W			
		D5	PSIO31							
		D4	PSIO30							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PSIO22	Serial interface Ch.2 interrupt level	0 to 7	X	R/W			
D1	PSIO21									
D0	PSIO20									
Key input, port input 0–3 interrupt enable register	0040270 (B)	D7–6	–	reserved	–	–	–	0 when being read.		
		D5	EK1	Key input 1	1	Enabled	0	Disabled	0	R/W
		D4	EK0	Key input 0					0	R/W
		D3	EP3	Port input 3	0	R/W				
		D2	EP2	Port input 2	0	R/W				
		D1	EP1	Port input 1	0	R/W				
		D0	EP0	Port input 0	0	R/W				
DMA interrupt enable register	0040271 (B)	D7–5	–	reserved	–	–	–	0 when being read.		
		D4	EIDMA	IDMA	1	Enabled	0	Disabled	0	R/W
		D3	EHDM3	High-speed DMA Ch.3					0	R/W
		D2	EHDM2	High-speed DMA Ch.2	0	R/W				
		D1	EHDM1	High-speed DMA Ch.1	0	R/W				
		D0	EHDM0	High-speed DMA Ch.0	0	R/W				

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks							
16-bit timer 0/1 interrupt enable register	0040272 (B)	D7	E16TC1	16-bit timer 1 comparison A	1	Enabled	0	Disabled	0	R/W						
		D6	E16TU1	16-bit timer 1 comparison B					0	R/W						
		D5-4	-	reserved			-	-	-	-	0 when being read.					
		D3	E16TC0	16-bit timer 0 comparison A	1	Enabled	0	Disabled	0	R/W						
		D2	E16TU0	16-bit timer 0 comparison B					0	R/W						
		D1-0	-	reserved			-	-	-	-	0 when being read.					
16-bit timer 2/3 interrupt enable register	0040273 (B)	D7	E16TC3	16-bit timer 3 comparison A	1	Enabled	0	Disabled	0	R/W						
		D6	E16TU3	16-bit timer 3 comparison B					0	R/W						
		D5-4	-	reserved			-	-	-	-	0 when being read.					
		D3	E16TC2	16-bit timer 2 comparison A	1	Enabled	0	Disabled	0	R/W						
		D2	E16TU2	16-bit timer 2 comparison B					0	R/W						
		D1-0	-	reserved			-	-	-	-	0 when being read.					
16-bit timer 4/5 interrupt enable register	0040274 (B)	D7	E16TC5	16-bit timer 5 comparison A	1	Enabled	0	Disabled	0	R/W						
		D6	E16TU5	16-bit timer 5 comparison B					0	R/W						
		D5-4	-	reserved			-	-	-	-	0 when being read.					
		D3	E16TC4	16-bit timer 4 comparison A	1	Enabled	0	Disabled	0	R/W						
		D2	E16TU4	16-bit timer 4 comparison B					0	R/W						
		D1-0	-	reserved			-	-	-	-	0 when being read.					
8-bit timer 0-3 interrupt enable register	0040275 (B)	D7-6	-	reserved			-	-	-	-	0 when being read.					
		D3	E8TU3	8-bit timer 3 underflow	1	Enabled	0	Disabled	0	R/W						
		D2	E8TU2	8-bit timer 2 underflow					0	R/W						
		D1	E8TU1	8-bit timer 1 underflow					0	R/W						
		D0	E8TU0	8-bit timer 0 underflow					0	R/W						
Serial I/F Ch.0/1 interrupt enable register	0040276 (B)	D7-6	-	reserved			-	-	-	-	0 when being read.					
		D5	ESTX1	SIF Ch.1 transmit buffer empty	1	Enabled	0	Disabled	0	R/W						
		D4	ESRX1	SIF Ch.1 receive buffer full					0	R/W						
		D3	ESERR1	SIF Ch.1 receive error					0	R/W						
		D2	ESTX0	SIF Ch.0 transmit buffer empty					0	R/W						
		D1	ESRX0	SIF Ch.0 receive buffer full					0	R/W						
		D0	ESERR0	SIF Ch.0 receive error					0	R/W						
		Port input 4-7, clock timer, A/D interrupt enable register	0040277 (B)	D7-6					-	reserved				-	-	-
D5	EP7			Port input 7					1	Enabled		0	Disabled	0	R/W	
D4	EP6			Port input 6	0	R/W										
D3	EP5			Port input 5	0	R/W										
D2	EP4			Port input 4	0	R/W										
D1	ECTM			Clock timer	0	R/W										
D0	EADE			A/D converter	0	R/W										
8-bit timer 4/5 interrupt enable register	0040278 (B)			D7-2	-	reserved					-			-	-	
		D1	E8TU5	8-bit timer 5 underflow	1	Enabled	0	Disabled			0			R/W		
		D0	E8TU4	8-bit timer 4 underflow					0	R/W						
		Serial I/F Ch.2/3 interrupt enable register	0040279 (B)	D7-6	-	reserved			-	-	-	-	0 when being read.			
D5	ESTX3			SIF Ch.3 transmit buffer empty	1	Enabled	0	Disabled	0	R/W						
D4	ESRX3			SIF Ch.3 receive buffer full					0	R/W						
D3	ESERR3			SIF Ch.3 receive error					0	R/W						
D2	ESTX2			SIF Ch.2 transmit buffer empty					0	R/W						
D1	ESRX2			SIF Ch.2 receive buffer full					0	R/W						
D0	ESERR2			SIF Ch.2 receive error					0	R/W						
Key input, port input 0-3 interrupt factor flag register	0040280 (B)	D7-6	-	reserved								-	-	-	-	0 when being read.
		D5	FK1	Key input 1	1	Factor is generated	0	No factor is generated	X	R/W						
		D4	FK0	Key input 0					X	R/W						
		D3	FP3	Port input 3					X	R/W						
		D2	FP2	Port input 2					X	R/W						
		D1	FP1	Port input 1					X	R/W						
		D0	FP0	Port input 0					X	R/W						
DMA interrupt factor flag register	0040281 (B)	D7-5	-	reserved								-	-	-	-	0 when being read.
		D4	FIDMA	IDMA	1	Factor is generated	0	No factor is generated	X	R/W						
		D3	FHDM3	High-speed DMA Ch.3					X	R/W						
		D2	FHDM2	High-speed DMA Ch.2					X	R/W						
		D1	FHDM1	High-speed DMA Ch.1					X	R/W						
		D0	FHDM0	High-speed DMA Ch.0					X	R/W						
16-bit timer 0/1 interrupt factor flag register	0040282 (B)	D7	F16TC1	16-bit timer 1 comparison A					1	Factor is generated		0	No factor is generated	X	R/W	
		D6	F16TU1	16-bit timer 1 comparison B	X	R/W										
		D5-4	-	reserved			-	-	-	-	0 when being read.					
		D3	F16TC0	16-bit timer 0 comparison A	1	Factor is generated	0	No factor is generated	X	R/W						
		D2	F16TU0	16-bit timer 0 comparison B					X	R/W						
		D1-0	-	reserved			-	-	-	-	0 when being read.					

**II-5 CORE BLOCK: ITC (Interrupt Controller)**

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
16-bit timer 2/3 interrupt factor flag register	0040283 (B)	D7	F16TC3	16-bit timer 3 comparison A	1	Factor is generated	0	No factor is generated	X	R/W	0 when being read.
		D6	F16TU3	16-bit timer 3 comparison B					X	R/W	
		D5-4	-	reserved				-	-		
		D3	F16TC2	16-bit timer 2 comparison A	1	Factor is generated	0	No factor is generated	X	R/W	
		D2	F16TU2	16-bit timer 2 comparison B					X	R/W	
		D1-0	-	reserved				-	-		
16-bit timer 4/5 interrupt factor flag register	0040284 (B)	D7	F16TC5	16-bit timer 5 comparison A	1	Factor is generated	0	No factor is generated	X	R/W	0 when being read.
		D6	F16TU5	16-bit timer 5 comparison B					X	R/W	
		D5-4	-	reserved				-	-		
		D3	F16TC4	16-bit timer 4 comparison A	1	Factor is generated	0	No factor is generated	X	R/W	
		D2	F16TU4	16-bit timer 4 comparison B					X	R/W	
		D1-0	-	reserved				-	-		
8-bit timer 0-3 interrupt factor flag register	0040285 (B)	D7-4	-	reserved				-	-	0 when being read.	
		D3	F8TU3	8-bit timer 3 underflow	1	Factor is generated	0	No factor is generated	X	R/W	
		D2	F8TU2	8-bit timer 2 underflow					X	R/W	
		D1	F8TU1	8-bit timer 1 underflow					X	R/W	
		D0	F8TU0	8-bit timer 0 underflow					X	R/W	
Serial I/F Ch.0/1 interrupt factor flag register	0040286 (B)	D7-6	-	reserved				-	-	0 when being read.	
		D5	FSTX1	SIF Ch.1 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	FSRX1	SIF Ch.1 receive buffer full					X	R/W	
		D3	FSERR1	SIF Ch.1 receive error					X	R/W	
		D2	FSTX0	SIF Ch.0 transmit buffer empty					X	R/W	
		D1	FSRX0	SIF Ch.0 receive buffer full					X	R/W	
		D0	FSERR0	SIF Ch.0 receive error					X	R/W	
Port input 4-7, clock timer, A/D interrupt factor flag register	0040287 (B)	D7-6	-	reserved							
		D5	FP7	Port input 7	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	FP6	Port input 6					X	R/W	
		D3	FP5	Port input 5					X	R/W	
		D2	FP4	Port input 4					X	R/W	
		D1	FCTM	Clock timer					X	R/W	
		D0	FADE	A/D converter					X	R/W	
8-bit timer 4/5 interrupt factor flag register	0040288 (B)	D7-2	-	reserved							
		D1	F8TU5	8-bit timer 5 underflow	1	Factor is generated	0	No factor is generated	X	R/W	
		D0	F8TU4	8-bit timer 4 underflow					X	R/W	
Serial I/F Ch.2/3 interrupt factor flag register	0040289 (B)	D7-6	-	reserved				-	-	0 when being read.	
		D5	FSTX3	SIF Ch.3 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	FSRX3	SIF Ch.3 receive buffer full					X	R/W	
		D3	FSERR3	SIF Ch.3 receive error					X	R/W	
		D2	FSTX2	SIF Ch.2 transmit buffer empty					X	R/W	
		D1	FSRX2	SIF Ch.2 receive buffer full					X	R/W	
		D0	FSERR2	SIF Ch.2 receive error					X	R/W	
Port input 0-3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA request register	0040290 (B)	D7	R16TC0	16-bit timer 0 comparison A					1	IDMA request	0
		D6	R16TU0	16-bit timer 0 comparison B	0	R/W					
		D5	RHDM1	High-speed DMA Ch.1	0	R/W					
		D4	RHDM0	High-speed DMA Ch.0	0	R/W					
		D3	RP3	Port input 3	0	R/W					
		D2	RP2	Port input 2	0	R/W					
		D1	RP1	Port input 1	0	R/W					
		D0	RP0	Port input 0	0	R/W					
		16-bit timer 1-4 IDMA request register	0040291 (B)	D7	R16TC4	16-bit timer 4 comparison A	1	IDMA request			
D6	R16TU4			16-bit timer 4 comparison B	0	R/W					
D5	R16TC3			16-bit timer 3 comparison A	0	R/W					
D4	R16TU3			16-bit timer 3 comparison B	0	R/W					
D3	R16TC2			16-bit timer 2 comparison A	0	R/W					
D2	R16TU2			16-bit timer 2 comparison B	0	R/W					
D1	R16TC1			16-bit timer 1 comparison A	0	R/W					
D0	R16TU1			16-bit timer 1 comparison B	0	R/W					
16-bit timer 5, 8-bit timer 0-3, serial I/F Ch.0 IDMA request register	0040292 (B)			D7	RSTX0	SIF Ch.0 transmit buffer empty			1	IDMA request	0
		D6	RSRX0	SIF Ch.0 receive buffer full	0	R/W					
		D5	R8TU3	8-bit timer 3 underflow	0	R/W					
		D4	R8TU2	8-bit timer 2 underflow	0	R/W					
		D3	R8TU1	8-bit timer 1 underflow	0	R/W					
		D2	R8TU0	8-bit timer 0 underflow	0	R/W					
		D1	R16TC5	16-bit timer 5 comparison A	0	R/W					
		D0	R16TU5	16-bit timer 5 comparison B	0	R/W					

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Serial I/F Ch.1, A/D, port input 4-7 IDMA request register	0040293 (B)	D7	RP7	Port input 7	1 IDMA request	0	Interrupt request	0	R/W	
		D6	RP6	Port input 6				0	R/W	
		D5	RP5	Port input 5				0	R/W	
		D4	RP4	Port input 4				0	R/W	
		D3	-	reserved	-		-	-	-	0 when being read.
		D2	RADE	A/D converter	1 IDMA request	0	Interrupt request	0	R/W	
		D1	RSTX1	SIF Ch.1 transmit buffer empty				0	R/W	
		D0	RSRX1	SIF Ch.1 receive buffer full				0	R/W	
Port input 0-3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA enable register	0040294 (B)	D7	DE16TC0	16-bit timer 0 comparison A	1 IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DE16TU0	16-bit timer 0 comparison B				0	R/W	
		D5	DEHDM1	High-speed DMA Ch.1				0	R/W	
		D4	DEHDM0	High-speed DMA Ch.0				0	R/W	
		D3	DEP3	Port input 3				0	R/W	
		D2	DEP2	Port input 2				0	R/W	
		D1	DEP1	Port input 1				0	R/W	
		D0	DEP0	Port input 0				0	R/W	
16-bit timer 1-4 IDMA enable register	0040295 (B)	D7	DE16TC4	16-bit timer 4 comparison A	1 IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DE16TU4	16-bit timer 4 comparison B				0	R/W	
		D5	DE16TC3	16-bit timer 3 comparison A				0	R/W	
		D4	DE16TU3	16-bit timer 3 comparison B				0	R/W	
		D3	DE16TC2	16-bit timer 2 comparison A				0	R/W	
		D2	DE16TU2	16-bit timer 2 comparison B				0	R/W	
		D1	DE16TC1	16-bit timer 1 comparison A				0	R/W	
		D0	DE16TU1	16-bit timer 1 comparison B				0	R/W	
16-bit timer 5, 8-bit timer 0-3, serial I/F Ch.0 IDMA enable register	0040296 (B)	D7	DESTX0	SIF Ch.0 transmit buffer empty	1 IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DESRX0	SIF Ch.0 receive buffer full				0	R/W	
		D5	DE8TU3	8-bit timer 3 underflow				0	R/W	
		D4	DE8TU2	8-bit timer 2 underflow				0	R/W	
		D3	DE8TU1	8-bit timer 1 underflow				0	R/W	
		D2	DE8TU0	8-bit timer 0 underflow				0	R/W	
		D1	DE16TC5	16-bit timer 5 comparison A				0	R/W	
		D0	DE16TU5	16-bit timer 5 comparison B				0	R/W	
Serial I/F Ch.1, A/D, port input 4-7 IDMA enable register	0040297 (B)	D7	DEP7	Port input 7	1 IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DEP6	Port input 6				0	R/W	
		D5	DEP5	Port input 5				0	R/W	
		D4	DEP4	Port input 4				0	R/W	
		D3	-	reserved	-		-	-	0 when being read.	
		D2	DEADE	A/D converter	1 IDMA enabled	0	IDMA disabled	0	R/W	
		D1	DESTX1	SIF Ch.1 transmit buffer empty				0	R/W	
		D0	DESRX1	SIF Ch.1 receive buffer full				0	R/W	
High-speed DMA Ch.0/1 trigger set-up register	0040298 (B)	D7	HSD1S3	High-speed DMA Ch.1 trigger set-up	0	Software trigger	0	R/W		
		D6	HSD1S2		1	K51 input (falling edge)	0			
		D5	HSD1S1		2	K51 input (rising edge)	0			
		D4	HSD1S0		3	Port 1 input	0			
					4	Port 5 input				
					5	8-bit timer Ch.1 underflow				
					6	16-bit timer Ch.1 compare B				
					7	16-bit timer Ch.1 compare A				
					8	16-bit timer Ch.5 compare B				
					9	16-bit timer Ch.5 compare A				
				A	SI/F Ch.1 Rx buffer full					
				B	SI/F Ch.1 Tx buffer empty					
				C	A/D conversion completion					
				D	FSI/F Ch.0 Rx buffer full					
				E	FSI/F Ch.0 Tx buffer empty					
		D3	HSD0S3	High-speed DMA Ch.0 trigger set-up	0	Software trigger	0	R/W		
		D2	HSD0S2		1	K50 input (falling edge)	0			
		D1	HSD0S1		2	K50 input (rising edge)	0			
D0	HSD0S0	3	Port 0 input		0					
		4	Port 4 input							
		5	8-bit timer Ch.0 underflow							
		6	16-bit timer Ch.0 compare B							
		7	16-bit timer Ch.0 compare A							
		8	16-bit timer Ch.4 compare B							
		9	16-bit timer Ch.4 compare A							
		A	SI/F Ch.0 Rx buffer full							
		B	SI/F Ch.0 Tx buffer empty							
		C	A/D conversion completion							
		D	reserved							
		E	reserved							

## II-5 CORE BLOCK: ITC (Interrupt Controller)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks								
High-speed DMA Ch.2/3 trigger set-up register	0040299 (B)	D7	HSD3S3	High-speed DMA Ch.3 trigger set-up	0 Software trigger 1 K54 input (falling edge) 2 K54 input (rising edge) 3 Port 3 input 4 Port 7 input 5 8-bit timer Ch.3 underflow 6 16-bit timer Ch.3 compare B 7 16-bit timer Ch.3 compare A 8 16-bit timer Ch.5 compare B 9 16-bit timer Ch.5 compare A A SI/F Ch.1 Rx buffer full B SI/F Ch.1 Tx buffer empty C A/D conversion completion D FSI/F Ch.0 Rx buffer full E FSI/F Ch.0 Tx buffer empty	0	R/W									
		D6	HSD3S2													
		D5	HSD3S1													
		D4	HSD3S0													
		D3	HSD2S3			High-speed DMA Ch.2 trigger set-up	0 Software trigger 1 K53 input (falling edge) 2 K53 input (rising edge) 3 Port 2 input 4 Port 6 input 5 8-bit timer Ch.2 underflow 6 16-bit timer Ch.2 compare B 7 16-bit timer Ch.2 compare A 8 16-bit timer Ch.4 compare B 9 16-bit timer Ch.4 compare A A SI/F Ch.0 Rx buffer full B SI/F Ch.0 Tx buffer empty C A/D conversion completion D reserved E reserved		0	R/W						
		D2	HSD2S2													
		D1	HSD2S1													
		D0	HSD2S0													
		D7-6	–						reserved	–	–	0 when being read.				
		8-bit timer 4/5 serial I/F Ch.2/3 IDMA request register	004029B (B)						D5	RSTX3	SIF Ch.3 transmit buffer empty	1 IDMA request	0 Interrupt request	0	R/W	
									D4	RSRX3	SIF Ch.3 receive buffer full			0	R/W	
									D3	RSTX2	SIF Ch.2 transmit buffer empty			0	R/W	
									D2	RSRX2	SIF Ch.2 receive buffer full			0	R/W	
									D1	R8TU5	8-bit timer 5 underflow			0	R/W	
D0	R8TU4			8-bit timer 4 underflow	0			R/W								
D7-6	–			reserved	–			–	0 when being read.							
8-bit timer 4/5 serial I/F Ch.2/3 IDMA enable register	004029C (B)	D5	DESTX3	SIF Ch.3 transmit buffer empty	1 IDMA enabled			0 IDMA disabled	0	R/W						
		D4	DESRX3	SIF Ch.3 receive buffer full					0	R/W						
		D3	DESTX2	SIF Ch.2 transmit buffer empty		0	R/W									
		D2	DESRX2	SIF Ch.2 receive buffer full		0	R/W									
		D1	DE8TU5	8-bit timer 5 underflow		0	R/W									
		D0	DE8TU4	8-bit timer 4 underflow		0	R/W									
		D7-6	–	reserved		–	–		0 when being read.							
Flag set/reset method select register	004029F (B)	D7-3	–	reserved	–	–	–	–								
		D2	DENONLY	IDMA enable register set method selection	1 Set only	0 RD/WR	1	R/W								
		D1	IDMAONLY	IDMA request register set method selection	1 Set only	0 RD/WR	1	R/W								
		D0	RSTONLY	Interrupt factor flag reset method selection	1 Reset only	0 RD/WR	1	R/W								
Interrupt priority register for functions extended	00402A0 (B)	D7	–	reserved	–	–	–	–	0 when being read.							
		D6	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W									
		D5	–	Interrupt level of extended function (reserved)		X										
		D4	–	Interrupt level of extended function (reserved)		X										
		D3	–	reserved	–	–	–	–	0 when being read.							
		D2	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W									
		D1	–	Interrupt level of extended function (reserved)		X										
D0	–	Interrupt level of extended function (reserved)		X												
Interrupt priority register for functions extended	00402A1 (B)	D7	–	reserved	–	–	–	–	0 when being read.							
		D6	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W									
		D5	–	Interrupt level of extended function (reserved)		X										
		D4	–	Interrupt level of extended function (reserved)		X										
		D3	–	reserved	–	–	–	–	0 when being read.							
		D2	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W									
		D1	–	Interrupt level of extended function (reserved)		X										
D0	–	Interrupt level of extended function (reserved)		X												

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Interrupt priority register for functions extended	00402A2 (B)	D7	–	reserved	–	–	–	0 when being read.			
		D6	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W				
		D5	–				X				
		D4	–				X				
		D3	–	reserved	–	–	–	–	0 when being read.		
		D2	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W				
		D1	–				X				
D0	–				X						
Interrupt priority register for functions extended	00402A3 (B)	D7	–	reserved	–	–	–	0 when being read.			
		D6	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W				
		D5	–				X				
		D4	–				X				
		D3	–	reserved	–	–	–	–	0 when being read.		
		D2	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W				
		D1	–				X				
D0	–				X						
Interrupt priority register for functions extended	00402A4 (B)	D7	–	reserved	–	–	–	0 when being read.			
		D6	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W				
		D5	–				X				
		D4	–				X				
		D3	–	reserved	–	–	–	–	0 when being read.		
		D2	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W				
		D1	–				X				
D0	–				X						
Interrupt priority register for functions extended	00402A5 (B)	D7	–	reserved	–	–	–	0 when being read.			
		D6	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W				
		D5	–				X				
		D4	–				X				
		D3	–	reserved	–	–	–	–	0 when being read.		
		D2	–	Interrupt level of extended function (reserved)	0 to 7	X	R/W				
		D1	–				X				
D0	–				X						
Interrupt enable register for functions extended	00402A6 (B)	D7	–	Extended interrupt (reserved)	1	Enabled	0	Disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
Interrupt enable register for functions extended	00402A7 (B)	D7	–	Extended interrupt (reserved)	1	Enabled	0	Disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
Interrupt enable register for functions extended	00402A8 (B)	D7	–	Extended interrupt (reserved)	1	Enabled	0	Disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
Interrupt factor flag register for functions extended	00402A9 (B)	D7	–	Extended interrupt (reserved)	1	Factor is generated	0	No factor is generated	X	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					X	R/W	
		D5	–	Extended interrupt (reserved)					X	R/W	
		D4	–	Extended interrupt (reserved)					X	R/W	
		D3	–	Extended interrupt (reserved)					X	R/W	
		D2	–	Extended interrupt (reserved)					X	R/W	
		D1	–	Extended interrupt (reserved)					X	R/W	
		D0	–	Extended interrupt (reserved)					X	R/W	

## II-5 CORE BLOCK: ITC (Interrupt Controller)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Interrupt factor flag register for functions extended	00402AA (B)	D7	–	Extended interrupt (reserved)	1	Factor is generated	0	No factor is generated	X	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					X	R/W	
		D5	–	Extended interrupt (reserved)					X	R/W	
		D4	–	Extended interrupt (reserved)					X	R/W	
		D3	–	Extended interrupt (reserved)					X	R/W	
		D2	–	Extended interrupt (reserved)					X	R/W	
		D1	–	Extended interrupt (reserved)					X	R/W	
		D0	–	Extended interrupt (reserved)					X	R/W	
Interrupt factor flag register for functions extended	00402AB (B)	D7	–	Extended interrupt (reserved)	1	Factor is generated	0	No factor is generated	X	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					X	R/W	
		D5	–	Extended interrupt (reserved)					X	R/W	
		D4	–	Extended interrupt (reserved)					X	R/W	
		D3	–	Extended interrupt (reserved)					X	R/W	
		D2	–	Extended interrupt (reserved)					X	R/W	
		D1	–	Extended interrupt (reserved)					X	R/W	
		D0	–	Extended interrupt (reserved)					X	R/W	
IDMA request register for functions extended	00402AC (B)	D7	–	Extended interrupt (reserved)	1	IDMA request	0	Interrupt request	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
IDMA request register for functions extended	00402AD (B)	D7	–	Extended interrupt (reserved)	1	IDMA request	0	Interrupt request	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
IDMA enable register for functions extended	00402AE (B)	D7	–	Extended interrupt (reserved)	1	IDMA enabled	0	IDMA disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
IDMA enable register for functions extended	00402AF (B)	D7	–	Extended interrupt (reserved)	1	IDMA enabled	0	IDMA disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
FIFO serial I/F Ch.0 interrupt priority register	00402B0 (B)	D7	–	reserved		–		–	–	0 when being read.	
		D6	PFSIO02	FIFO serial interface Ch.0		0 to 7		X	R/W		
		D5	PFSIO01	FIFO serial interface Ch.0				X	R/W		
		D4	PFSIO00	interrupt level				X	R/W		
D3–0	–	reserved		–		–	–	–	0 when being read.		
FIFO serial I/F Ch.0 interrupt enable register	00402B1 (B)	D7–6	–	reserved		–		–	–	0 when being read.	
		D5	EFSTX0	FSI/F Ch.0 transmit buffer empty	1	Enabled	0	Disabled	0	R/W	
		D4	EFSRX0	FSI/F Ch.0 receive buffer full					0	R/W	
		D3	EFSERR0	FSI/F Ch.0 receive error					0	R/W	
		D2–0	–	reserved		–		–	–	–	0 when being read.
FIFO serial I/F Ch.0 interrupt factor flag register	00402B2 (B)	D7–6	–	reserved		–		–	–	0 when being read.	
		D5	FFSTX0	FSI/F Ch.0 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	FFSRX0	FSI/F Ch.0 receive buffer full					X	R/W	
		D3	FFSERR0	FSI/F Ch.0 receive error					X	R/W	
		D2–0	–	reserved		–		–	–	–	0 when being read.
FIFO serial I/F Ch.0 IDMA request register	00402B3 (B)	D7–4	–	reserved		–		–	–	0 when being read.	
		D3	RFSTX0	FSI/F Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	0	R/W	
		D2	RFSRX0	FSI/F Ch.0 receive buffer full					0	R/W	
		D1–0	–	reserved						–	



Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
FIFO serial I/F Ch.0 IDMA enable register	00402B4 (B)	D7-4	–	reserved	–	–	–	0 when being read.				
		D3	DEFSTX0	FSI/F Ch.0 transmit buffer empty	1 IDMA enabled	0 IDMA disabled	0	R/W				
		D2	DEFSRX0	FSI/F Ch.0 receive buffer full			0	R/W				
		D1-0	–	reserved	–	–	–	–	0 when being read.			
TTBR write protect register	004812D (B)	D7	TBRP7	TTBR register write protect  Writing 01011001 (0x59) removes the TTBR (0x48134) write protection. Writing other data sets the write protection.		0	W	Undefined in read.				
		D6	TBRP6			0						
		D5	TBRP5			0						
		D4	TBRP4			0						
		D3	TBRP3			0						
		D2	TBRP2			0						
		D1	TBRP1			0						
		D0	TBRP0			0						
		TTBR low- order register	0048134 (HW)			DF	TTBR15		Trap table base address [15:10]		0	R/W
DE	TTBR14			0								
DD	TTBR13			0								
DC	TTBR12			0								
DB	TTBR11			0								
DA	TTBR10			0								
D9	TTBR09			Trap table base address [9:0]	Fixed at 0	0	R	0 when being read. Writing 1 not allowed.				
D8	TTBR08					0						
D7	TTBR07					0						
D6	TTBR06					0						
D5	TTBR05					0						
D4	TTBR04					0						
D3	TTBR03					0						
D2	TTBR02					0						
D1	TTBR01					0						
D0	TTBR00			0								
TTBR high- order register	0048136 (HW)			DF	TTBR33	Trap table base address [31:28]	Fixed at 0	0	R	0 when being read. Writing 1 not allowed.		
				DE	TTBR32			0				
				DD	TTBR31			0				
		DC	TTBR30	0								
		DB	TTBR2B	Trap table base address [27:16]	0x0C0	0	R/W					
		DA	TTBR2A			0						
		D9	TTBR29			0						
		D8	TTBR28			0						
		D7	TTBR27			1						
		D6	TTBR26			1						
		D5	TTBR25			0						
		D4	TTBR24			0						
		D3	TTBR23			0						
		D2	TTBR22			0						
		D1	TTBR21			0						
		D0	TTBR20			0						

The following collectively explains the basic functions of each control register/bit. For details about individual interrupt systems and the contents classified by an interrupt factor, refer to the descriptions of the peripheral circuits in this manual.

**Pxxx2–Pxxx0: Interrupt priority register**

Set the priority levels of each interrupt system in the range of 0 to 7.

If this register is set below the IL value of the PSR, no interrupt is generated. The value of this register when initially reset is indeterminate.

**Exxx: Interrupt enable register**

Enable or disable interrupt generation to the CPU.

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

Interrupts are enabled when the corresponding bits of this register are set to "1" and are disabled when the bits are set to "0".

For the interrupt factors used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

When initially reset, this register is set to "0" (interrupt disabled).

## II-5 CORE BLOCK: ITC (Interrupt Controller)

---

### Fxxx: Interrupt factor flag

---

Indicate the status of interrupt factors generated.

#### When read

Read "1": Interrupt factor generated

Read "0": No interrupt factor generated

#### When written using the reset-only method (default)

Write "1": Factor flag is reset

Write "0": Invalid

#### When written using the read/write method

Write "1": Factor flag is set

Write "0": Factor flag is reset

The interrupt factor flag is set to "1" when an interrupt factor occurs in each peripheral circuit.

If the following conditions are met at this time, an interrupt is generated to the CPU:

1. The corresponding bit of the interrupt enable register is set to "1".
2. No other interrupt request of higher priority has occurred.
3. The IE bit of the PSR is set to "1" (interrupt enabled).
4. The corresponding interrupt priority register is set to a level higher than the CPU's interrupt level (IL).

When using an interrupt factor to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is always set to "1" when an interrupt factor occurs no matter how the interrupt enable and interrupt priority registers are set.

In order for the next interrupt to be accepted after interrupt generation, the interrupt factor flag must be reset and the PSR must be set up again (by setting the IL below the level indicated by the interrupt priority register and setting the IE bit to "1" or executing the reti instruction).

The interrupt factor flag can only be reset by a write instruction in the software application. If the PSR is again set up to accept interrupts (or the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt may occur again. Note also that the value to be written to reset the flag is "1" when using the reset-only method (RSTONLY = "1") and "0" when using the read/write method (RSTONLY = "0"). Be careful not to confuse these two conditions.

The interrupt factor flag becomes indeterminate when initially reset, so be sure to reset the flag in the software application.

---

### Rxxx: IDMA request register

---

Specify whether or not to invoke IDMA when an interrupt factor occurs.

#### When using the set-only method (default)

Write "1": IDMA request

Write "0": Not changed

Read: Valid

#### When using the read/write method

Write "1": IDMA request

Write "0": Interrupt request

Read: Valid

If a bit of this register is set to "1", IDMA is invoked when the corresponding interrupt factor occurs and the programmed data transfer is performed. If the register bit is set to "0", regular interrupt processing is performed, without ever invoking IDMA.

For details about IDMA, refer to "IDMA (Intelligent DMA)".

If interrupts are enabled on the IDMA side and the transfer counter reaches the terminal count of 0 after completion of DMA transfer, the IDMA request register is reset to "0" and an interrupt request for the interrupt factor that enabled IDMA invoking is generated.

After an initial reset, this register is set to "0" (Interrupt is requested).

**DExxx:** IDMA enable register

Enable or disable the IDMA request.

**When using the set-only method (default)**

Write "1": IDMA enabled

Write "0": Not changed

Read: Valid

**When using the read/write method**

Write "1": IDMA enabled

Write "0": IDMA disabled

Read: Valid

If a bit of this register is set to "1", the IDMA request by the interrupt factor is enabled. If the register bit is set to "0", the IDMA request is disabled.

After an initial reset, this register is set to "0" (IDMA is disabled).

**RSTONLY:** Interrupt factor flag reset method selection (D0) / Flag set/reset method select register (0x4029F)

Select the method for resetting the interrupt factor flag.

Write "1": Reset-only method

Write "0": Read/write method

Read: Valid

With the reset-only method, the interrupt factor flag is reset by writing "1".

The interrupt factor flags for which "0" has been written can neither be set nor reset. Therefore, this method ensures that only a specific factor flag is reset. However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that an interrupt factor flag that has been set to "1" is reset by writing. This method cannot be used to set any interrupt factor flag in the software application.

The read/write method is selected by writing "0" to RSTONLY. When this method is selected, interrupt factor flags can be read and written as for other registers. Therefore, the flag is reset by writing "0" and set by writing "1". In this case all factor flags for which "0" has been written are reset. Even in a read-modify-write operation, an interrupt factor can occur between read and write instructions, so be careful when using this method.

After an initial reset, RSTONLY is set to "1" (reset-only method).

**IDMAONLY:** IDMA request register set method selection (D1) / Flag set/reset method select register (0x4029F)

Select the method for setting the IDMA request registers.

Write "1": Set-only method

Write "0": Read/write method

Read: Valid

With the set-only method, IDMA request bits are set by writing "1".

The IDMA request bits for which "0" has been written can neither be set nor reset. Therefore, this method ensures that only a specific IDMA request bit is set. However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that an IDMA request bit that has been set to "1" is not reset by writing.

The read/write method is selected by writing "0" to IDMAONLY. When this method is selected, IDMA request bits can be read and written as for other registers. Therefore, the IDMA request bit is reset by writing "0" and set by writing "1". In this case all IDMA request bits for which "0" has been written are reset. Even in a read-modify-write operation, an IDMA request bit can be reset by the hardware between the read and the write, so be careful when using this method.

After an initial reset, IDMAONLY is set to "1" (set-only method).

---

**DENONLY: IDMA enable register set method selection (D2) / Flag set/reset method select register (0x4029F)**

---

Select the method for setting the IDMA enable registers.

Write "1": Set-only method

Write "0": Read/write method

Read: Valid

With the set-only method, IDMA enable bits are set by writing "1".

The IDMA enable bits for which "0" has been written can neither be set nor reset. Therefore, this method ensures that only a specific IDMA enable bit is set. However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that an IDMA enable bit that has been set to "1" is not reset by writing.

The read/write method is selected by writing "0" to DENONLY. When this method is selected, IDMA enable bits can be read and written as for other registers. Therefore, the IDMA enable bit is reset by writing "0" and set by writing "1". In this case all IDMA enable bits for which "0" has been written are reset. Even in a read-modify-write operation, an interrupt enable bit can be reset by the hardware between the read and the write, so be careful when using this method.

After an initial reset, DENONLY is set to "1" (set-only method).

---

**TBRP7–TBRP0: TTBR register write protection ([D[7:0]) / TTBR write-protect register (0x4812D)**

---

Remove write protection for the TTBR register.

Write 0x59: Write protection is removed

Write not the above: No operation (write protected)

Read: Valid

Before writing to the TTBR register, set TBRP to "0x59" to remove the write protection. Then when data is written to the most significant byte (0x48137) of the TTBR, the register once again becomes write-protected.

After an initial reset, TBRP is set to "0x0" (write protected).

---

**TTBR09–TTBR00: Trap table base address [9:0] (D[9:0]) / TTBR low-order register (0x48134[HW])****TTBR15–TTBR10: Trap table base address [15:10] (D[F:A]) / TTBR low-order register (0x48134[HW])****TTBR2B–TTBR20: Trap table base address [27:16] (D[B:0]) / TTBR high-order register (0x48136[HW])****TTBR33–TTBR30: Trap table base address [31:28] (D[F:C]) / TTBR high-order register (0x48136[HW])**

---

Set the starting address of the trap table. TTBR0 and TTBR3 are read-only registers and are fixed to "0". For this reason, the trap table starting address always begins with a 1KB boundary address.

The TTBR registers normally are write-protected to prevent them from being inadvertently rewritten. To remove this write protect function, another register, TBRP (D[7:0]) / TTBR write-protect register (0x4812D), is provided. A write to the TTBR register is enabled by writing "0x59" to TBRP and is disabled back again by a write to the most significant byte of the TTBR register (0x48137). Consequently, writes to the TTBR register need to begin with the low-order half-word first. However, since occurrences of NMI and the like between writes of the low-order and high-order half-words cause malfunctions, it is recommended that the register be written in words.

After an initial reset, the TTBR register is set to 0x0C00000.

## Programming Notes

---

- (1) In cases when an interrupt factor that is used for restarting from the standby mode has been set to invoke IDMA, IDMA is started up by the interrupt at its occurrence. In SLEEP mode, the high-speed (OSC3) oscillation circuit also starts operating. However, if an interrupt to be generated upon completion of IDMA is disabled at the setting of IDMA side, no interrupt request is signaled to the CPU. Therefore, the CPU remains idle until the next interrupt request is generated.
- (2) As the C33 STD Core CPU function, the IL allows interrupt levels to be set in the range of 0 to 15. However, since the interrupt priority register in the C33 Core Block consists of three bits, interrupt levels in each interrupt system can only be set for up to 8.
- (3) When the reset-only method is used to reset the interrupt factor flag (by writing "1"), if a read-modify-write instruction (e.g., bset, bclr, or bnot) is executed, the other interrupt factor flags at the same address that have been set to "1" are reset by a write. This requires caution. In cases when the read/write method is used to reset the interrupt factor flag (by writing "0"), all factor flags for which "0" has been written are reset. When a read-modify-write operation is performed, an interrupt factor may occur between reads and writes, so be careful when using this method.  
The same applies to the set-only method and read/write method for the IDMA request and IDMA enable registers.
- (4) After an initial reset, the interrupt factor flags and interrupt priority registers all become indeterminate. To prevent unwanted interrupts or IDMA requests from being generated inadvertently, be sure to reset these flags and registers in the software application.
- (5) To prevent another interrupt from being generated for the same factor again after generation of an interrupt, be sure to reset the interrupt factor flag before enabling interrupts and setting the PSR again or executing the reti instruction.

THIS PAGE IS BLANK.

## II-6 CLG (Clock Generator)

This section describes the method for controlling the system clock.

### Configuration of Clock Generator

The C33 Core Block has a built-in clock generator that consists of a high-speed oscillation circuit (OSC3) and a PLL.

The high-speed (OSC3) oscillation circuit generates the main clock for the CPU and internal peripheral circuits (e.g., DMA, serial interface, programmable timer, and A/D converter).

Furthermore, the clock generator can input a sub clock, such as low-speed (OSC1, 32.768 kHz, Typ.) clock generated by the Peripheral Block, for the clock timer and for operating the CPU at a low clock speed in order to reduce current consumption.

**Note:** When the Peripheral Block including the low-speed (OSC1) oscillation circuit is used, the source clocks for the CPU and the peripheral circuits (e.g., serial interface, programmable timer, and A/D converter) can be selected between the OSC3 clock and the OSC1 clock. For details, refer to "Setting and Switching Over the CPU Operating Clock" in this section and "Prescaler" and "Low-Speed (OSC1) Oscillation Circuit" of the Peripheral Block.

Figure II.6.1 shows the configuration of the clock generator.

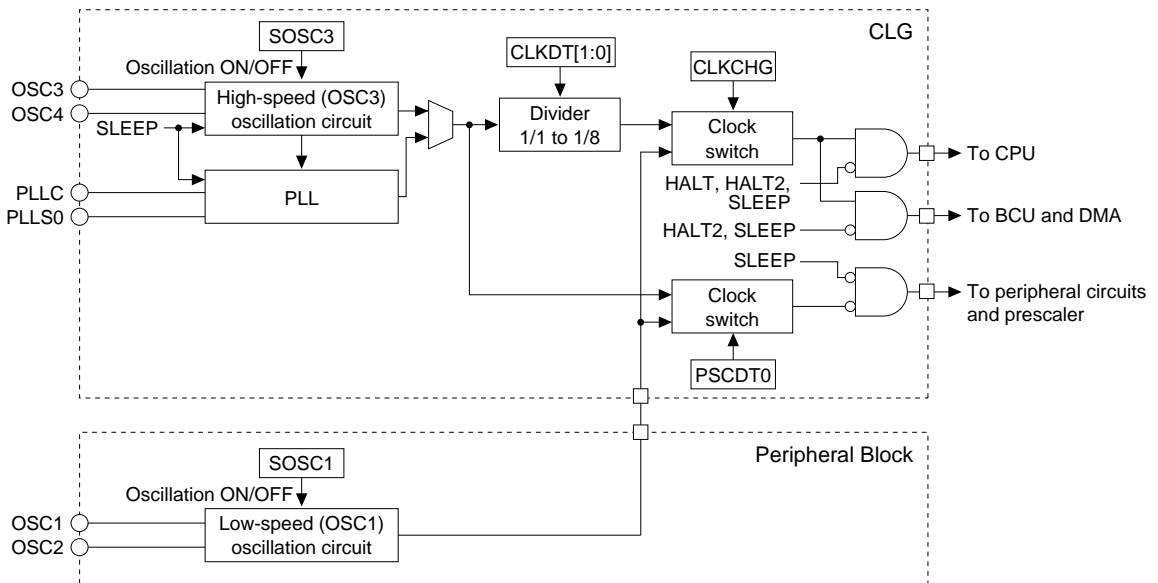


Figure II.6.1 Configuration of Clock Generator

After an initial reset, the output (OSC3 clock) of the high-speed (OSC3) oscillation circuit or the PLL output is set for the CPU operating clock.

When the low-speed (OSC1) oscillation circuit is used, the CPU operating clock can be switched to the output (OSC1 clock) of the low-speed (OSC1) oscillation circuit in a program. Furthermore, each oscillation circuit can be stopped in a program.

If the OSC3 clock is unnecessary such as when performing clock processing only, set the OSC1 clock for operation of the CPU and turn off the high-speed (OSC3) oscillation circuit in order to reduce current consumption. In addition, when SLEEP mode is set, the high-speed (OSC3) oscillation circuit is turned off, greatly reducing current consumption (no internal units except for the clock timer need to be operated).

## I/O Pins of Clock Generator

Table II.6.1 lists the I/O pins of the clock generator.

Table II.6.1 I/O Pins of Clock Generator

Pin name	I/O	Function
OSC3	I	High-speed (OSC3) oscillation input pin Crystal/ceramic oscillation or external clock input
OSC4	O	High-speed (OSC3) oscillation output pin Crystal/ceramic oscillation (open when external clock is used)
PLLC	-	Capasitor connecting pin for PLL
PLLS0	I	PLL set-up pins 0: PLL not used, 1: Frequency-doubling mode

### High-Speed (OSC3) Oscillation Circuit

The high-speed (OSC3) oscillation circuit generates the main clock for the CPU and internal peripheral circuits (e.g., DMA, serial interface, programmable timer, and A/D converter).

This circuit can be a crystal or a ceramic oscillation circuit. Optionally an external clock source can be used.

Figure II.6.2 shows the structure of the high-speed (OSC3) oscillation circuit.

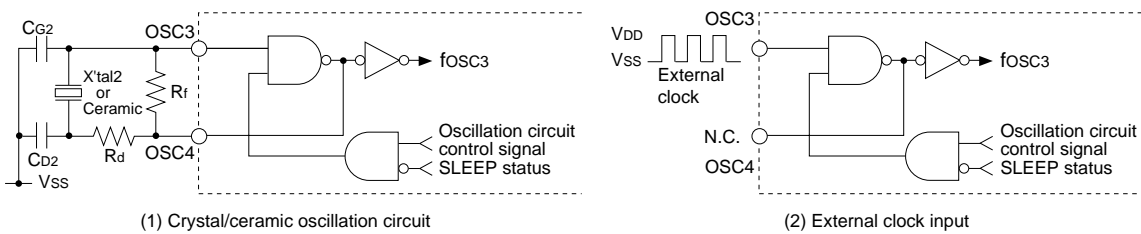


Figure II.6.2 High-Speed (OSC3) Oscillation Circuit

When using a crystal or ceramic oscillation for this circuit, connect a crystal (X'tal2) or ceramic (Ceramic) resonator, a feedback resistor (Rf), two capacitors (CG2, CD2), and a drain resistor (Rd), if necessary, to the OSC3 and OSC4 pins and VSS as shown in Figure II.6.2 (1).

When an external clock is used, leave the OSC4 pin open and input a square-wave clock to the OSC3 pin.

The range of oscillation frequencies is 5 MHz to 33 MHz. When an external clock is input, a clock within the range from 2 MHz to 33 MHz can be used.

For details on oscillation characteristics and the external clock input characteristics, refer to "Electrical Characteristics".



## PLL

The PLL inputs the OSC3 clock and multiply its frequency. The multiply mode should be set using the PLLS0 pin according to the OSC3 clock frequency.

Table II.6.2 Setting the PLLS0 Pin

PLLS0	Mode	fin (OSC3 clock)	fout
1	Frequency-doubling mode	15 to 25 MHz	30 to 50 MHz
0	PLL not used	–	Not used

Figure II.6.3 shows a basic external connection diagram for the PLL pins.

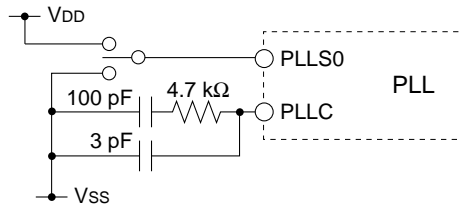


Figure II.6.3 External Connection Diagram

**Note:** When the PLL is not used, the OSC3 oscillation output is used as the source clock. In this case, the oscillation frequency range is 15 MHz to 25 MHz. Leave the PLLC pin open.

## Controlling Oscillation

The high-speed (OSC3) oscillation circuit can be turned on or off using SOSC3 (D1) / Power control register (0x40180).

The oscillation circuit is turned off by writing "0" to SOSC3 and turned back on again by writing "1". SOSC3 is set to "1" at initial reset, so the oscillation circuit is turned on.

- Notes:**
- When the high-speed (OSC3) oscillation circuit is used as the clock source for the CPU operating clock, it cannot be turned off. In this case, writing "0" to SOSC3 is ignored. Note also that writing to SOSC3 is allowed only when the power-control register protection flag is set to "0b10010110".
  - Immediately after the oscillation circuit is turned on, a certain period of time is required for oscillation to stabilize (for 3.3-V crystal resonator, this time is 10 ms max.). To prevent the device from operating erratically, do not use the clock until its oscillation has stabilized.

The high-speed (OSC3) oscillation circuit turns off when the CPU is set in SLEEP mode.

## Setting and Switching Over the CPU Operating Clock

### Setting the CPU operating clock frequency

When operating the CPU with the high-speed (OSC3) clock, the operating frequency can be switched over in four steps. Use CLKDTC[1:0] (D[7:6]) / Power control register (0x40180) for this switchover.

Table II.6.3 Setting of CPU Operating Clock

CLKDTC1	CLKDTC0	Division ratio
1	1	f <sub>out</sub> /8
1	0	f <sub>out</sub> /4
0	1	f <sub>out</sub> /2
0	0	f <sub>out</sub> /1

f<sub>out</sub>: PLL output

The clock thus set becomes the system clock, which is used as the CPU operating clock and the bus clock.

At initial reset, the division ratio is set to f<sub>out</sub>/1, so the CPU is operated directly by the PLL output clock. Since the device's current consumption can be decreased by reducing the CPU operating speed, switch over the operating frequency as necessary.

This setting is effective only for the high-speed (OSC3) clock, and has no effect when the low-speed (OSC1) clock is used as the system clock.

**Note:** Writing to CLKDTC[1:0] is effective only when the power-control register protection flag is set to "0b10010110".

### Switching over the CPU operating clock

**Note:** The CPU operating clock can be switched from OSC3 to OSC1 only when the low-speed (OSC1) oscillation circuit in the Peripheral Block is used.

After an initial reset, the CPU starts operating using the OSC3 clock. All internal peripheral circuits also operate.

In cases in which some peripheral circuits (e.g., programmable timer, serial interface, A/D converter, and ports) do not need to be operate or processing in low-speed operation is possible, and the CPU can process its jobs at a low clock speed, the CPU operating clock can be switched to the OSC1 clock, thereby reducing current consumption. Use CLKCHG (D2) / Power control register (0x40180) to switch over the operating clock.

#### Procedure for switching over from the OSC3 clock to the OSC1 clock

1. Turn on the low-speed (OSC1) oscillation circuit (by writing "1" to SOSC1).
2. Wait until the OSC1 oscillation stabilizes (three seconds or more).
3. Change the CPU operating clock (by writing "0" to CLKCHG).
4. Turn off the high-speed (OSC3) oscillation circuit (by writing "0" to SOS3).

\* Steps 1 and 2 are required only when the low-speed (OSC1) oscillation circuit is inactive.

**Notes:**

- Use separate instructions to switch from OSC3 to OSC1 and turn the OSC3 oscillation off. If these operations are processed simultaneously using one instruction, the CPU may operate erratically.

- Make sure the operation of the peripheral circuits, such as the programmable timer and serial interface is terminated before the OSC3 oscillation is turned off in order to prevent them from operating erratically or the prescaler clock is set as OSC1. In addition, in order to prevent incorrect operation, a setup of prescaler must be performed before changing the CPU clock.

#### Procedure for switching over from the OSC1 clock to the OSC3 clock

1. Turn on the high-speed (OSC3) oscillation circuit (by writing "1" to SOS3).
2. Wait until the OSC3 oscillation stabilizes (10 ms or more for a 3.3-V crystal resonator).
3. Switch over the CPU operating clock (by writing "1" to CLKCHG).

**Note:** The operating clock switchover by CLKCHG is effective only when both oscillation circuits are on and the power-control register protection flag is set to "0b10010110".

## Power-Control Register Protection Flag

The power-control register at address 0x40180, which is used to control the oscillation circuits and the CPU operating clock, is normally disabled against writing in order to prevent it from malfunctioning due to unnecessary writing.

To enable this register for writing, the power-control register protection flag CLGP[7:0] (D[7:0]) / Power-control protection register (0x4019E) must be set to "0b10010110". Note that this setting allows for the power-control register (0x40180) to be written to only once, so all bits of CLGP[7:0] are cleared to "0" when this address is written to. Therefore, CLGP[7:0] must be set to "0b10010110" each time the power-control register (0x40180) is written to. The flag CLGP[7:0] does not affect the readout from the power-control register (0x40180).

## Operation in Standby Mode

In HALT mode, which is entered by executing the halt instruction, the high-speed (OSC3) and low-speed (OSC1) oscillation circuits both retain their status before HALT mode is entered. Under normal conditions, therefore, there is no need to control the oscillation circuits before entering or after exiting HALT mode.

The high-speed (OSC3) oscillation circuit stops operating after SLEEP mode is entered, which is done by executing the slp (sleep) instruction. If the high-speed (OSC3) oscillation circuit was operating before SLEEP mode was entered, it automatically starts oscillating again after SLEEP mode is exited.

In addition, if the CPU was operating using the OSC3 clock before SLEEP mode was entered, the CPU starts operating using the OSC3 clock again even after SLEEP mode is exited. The high-speed (OSC3) oscillation circuit requires an oscillation stabilization time after oscillation starts. Furthermore, a lockup time is required when the PLL is used. To prevent the CPU from operating erratically upon restart during this period, the C33 Core Block is designed to allow the clock supply to the CPU to be disabled in the hardware after SLEEP mode is exited. Use 8T1ON (D2) / Clock option register (0x40190) to select this function. Use 8-bit programmable timer 1 to set the waiting time before clock supply is started.

The processing procedure and the operations to be performed when this function is used are as follows:

1. Disable the 8-bit programmable timer 1 interrupt.
2. Preset the initial count to 8-bit programmable timer 1.  
Set a value that will provide an ample stabilization waiting time. It is also necessary to set the input clock for 8-bit programmable timer 1 using the prescaler.
3. Enable the interrupt used to exit SLEEP mode.  
Before enabling the interrupt, be sure to reset the interrupt factor flag.
4. Write "0" to 8T1ON (turn on the function for waiting until the oscillation stabilizes after exiting SLEEP mode).
5. Activate 8-bit programmable timer 1 to start counting.
6. Enter SLEEP mode using the slp instruction.  
:  
SLEEP mode  
:  
:
7. Exit SLEEP mode using an NMI, input port, or timer interrupt.
8. The high-speed (OSC3) oscillation circuit starts oscillating when SLEEP mode is exited. 8-bit programmable timer 1 also is made to start counting using the OSC3 clock.
9. 8-bit programmable timer 1 underflows.  
The operating clock supply to the CPU is begun by the underflow signal, so that the CPU restarts.

For details on how to control the 8-bit programmable timer, prescaler, and interrupts, refer to the description of each item in this manual.

**Note:** The function for waiting until the high-speed (OSC3) oscillation is stabilized by 8T1ON is effective only when SLEEP mode is exited.

Writing to 8T1ON is effective only when the power-control register protection flag is set to "0b10010110".

## I/O Memory of Clock Generator

Table II.6.4 lists the control bits of clock generator.

Table II.6.4 Control Bits of Clock Generator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Power control register	0040180 (B)	D7	CLKDT1	System clock division ratio selection	CLKDTC[1:0]		0	R/W			
		D6	CLKDT0		1	1				1/8	
					1	0				1/4	
					0	1				1/2	
					0	0				1/1	
		D5	PSCON	Prescaler On/Off control	1	On	0	Off	1	R/W	
		D4-3	–	reserved					0	–	Writing 1 not allowed.
		D2	CLKCHG	CPU operating clock switch	1	OSC3	0	OSC1	1	R/W	
D1	SOSC3	High-speed (OSC3) oscillation On/Off	1	On	0	Off	1	R/W			
D0	SOSC1	Low-speed (OSC1) oscillation On/Off	1	On	0	Off	1	R/W			
Prescaler clock select register	0040181 (B)	D7-1	–	reserved				0	–		
		D0	PSCDT0	Prescaler clock selection	1	OSC1	0	OSC3/PLL	0	R/W	
Clock option register	0040190 (B)	D7-4	–	–				–	–	0 when being read.	
		D3	HLT2OP	HALT clock option	1	On	0	Off	0	R/W	
		D2	8T1ON	OSC3-stabilize waiting function	1	Off	0	On	1	R/W	
		D1	–	reserved					0	–	Do not write 1.
		D0	PF1ON	OSC1 external output control	1	On	0	Off	0	R/W	
Power control protect register	004019E (B)	D7	CLGP7	Power control register protect flag	Writing 10010110 (0x96)		0	R/W			
		D6	CLGP6		removes the write protection of the power control register					0	
		D5	CLGP5		the power control register (0x40180) and the clock option register (0x40190).					0	
		D4	CLGP4							0	
		D3	CLGP3							0	
		D2	CLGP2		Writing another value set the write protection.					0	
		D1	CLGP1							0	
		D0	CLGP0							0	

### SOSC1: Low-speed (OSC1) oscillation control (D0) / Power control register (0x40180)

Turns the low-speed (OSC1) oscillation on or off.

Write "1": OSC1 oscillation turned on

Write "0": OSC1 oscillation turned off

Read: Valid

The oscillation of the low-speed (OSC1) oscillation circuit is stopped by writing "0" to SOSC1, and started again by writing "1".

Since a duration of maximum three seconds is required for oscillation to stabilize after the oscillation has been restarted, at least this length of time must pass before the OSC1 clock can be used.

Writing to SOSC1 is allowed only when CLGP[7:0] is set to "0b10010110". Note also that if the CPU is operating using the OSC1 clock, writing "0" to SOSC1 is ignored and the oscillation is not turned off.

At initial reset, SOSC1 is set to "1" (OSC1 oscillation turned on).

**Note:** This control bit is effective only when the low-speed (OSC1) oscillation circuit in the Peripheral Block is used.

### SOSC3: High-speed (OSC3) oscillation control (D1) / Power control register (0x40180)

Turns the high-speed (OSC3) oscillation on or off.

Write "1": OSC3 oscillation turned on

Write "0": OSC3 oscillation turned off

Read: Valid

The oscillation of the high-speed (OSC3) oscillation circuit is stopped by writing "0" to SOSC3, and started again by writing "1".

Since a duration of maximum 10 ms (for a 3.3-V crystal resonator) is required for oscillation to stabilize after the oscillation has been restarted, at least this length of time must pass before the OSC3 clock can be used.

Writing to SOSC3 is allowed only when CLGP[7:0] is set to "0b10010110". Note also that if the CPU is operating using the OSC3 clock, writing "0" to SOSC3 is ignored and the oscillation is not turned off.

At initial reset, SOSC3 is set to "1" (OSC3 oscillation turned on).

**CLKCHG:** CPU operating clock switch (D2) / Power control register (0x40180)

Selects the CPU operating clock.

Write "1": OSC3 clock

Write "0": OSC1 clock

Read: Valid

The OSC3 clock is selected as the CPU operating clock by writing "1" to CLKCHG, and OSC1 is selected by writing "0". The operating clock can be switched over in this way only when both the high-speed (OSC3) and low-speed (OSC1) oscillation circuits are on. In addition, writing to CLKCHG is effective only when CLGP[7:0] is set to "0b10010110". Immediately after the oscillation circuit has started oscillating, wait for the oscillation to stabilize before switching over the CPU operating clock.

At initial reset, CLKCHG is set to "1" (OSC3 clock).

**Note:** This control bit is effective only when the low-speed (OSC1) oscillation circuit in the Peripheral Block is used.

**CLKDT1–CLKDT0:** CPU operating frequency selection (D[7:6]) / Power control register (0x40180)

Select the CPU operating clock frequency.

Table II.6.5 Setting of CPU Operating Clock

CLKDT1	CLKDT0	Division ratio
1	1	fout/8
1	0	fout/4
0	1	fout/2
0	0	fout/1

fout: PLL output

This setting is effective when the CPU is operated using the high-speed (OSC3) clock and has no effect on the low-speed (OSC1) clock. Writing to CLKDT[1:0] is allowed only when CLGP[7:0] is set to "0b10010110".

At initial reset, CLKDT is set to "0" (fout/1).

**PSCDT0:** Prescaler clock selection (D0) / Prescaler clock select register (0x40181)

Select the source clock for the prescaler.

Write "1": OSC1 clock

Write "0": OSC3 clock/PLL output clock

Read: Valid

When "1" is written to PSCDT0, the OSC1 clock (typ. 32 kHz) is selected.

When "0" is written, the OSC3 clock (when the PLL is not used) or the PLL output clock (when the PLL is used) is selected.

For the prescaler clock, the clock source same as the CPU operating clock must be selected.

At initial reset, PSCDT0 is set to "0" (OSC3 clock/PLL output clock).

**8T1ON:** High-speed (OSC3) oscillation waiting function (D2) / Clock option register (0x40190)

Sets the function for waiting until the high-speed (OSC3) oscillation stabilizes after SLEEP mode is exited.

Write "1": Off

Write "0": On

Read: Valid

After SLEEP mode is exited, the high-speed (OSC3) oscillation waiting function is effective by writing "1" to 8T1ON. For this function to be used, the waiting time must be set in 8-bit programmable timer 1 to allow it to start counting before entering SLEEP mode. After SLEEP mode is exited, the OSC3 clock is not supplied to the CPU until 8-bit programmable timer 1 underflows. This function will not work when 8T1ON is set to "0".

The high-speed (OSC3) oscillation waiting function is effective only when SLEEP mode is exited.

Writing to 8T1ON is effective only when CLGP[7:0] is set to "0b10010110".

When writing to 8T1ON, always be sure to write "0" to the reserved bits at address 0x40190.

At initial reset, 8T1ON is set to "1" (Off).

## II-6 CORE BLOCK: CLG (Clock Generator)

### HLT2OP: HALT clock option (D3) / Clock option register (0x40190)

Select a HALT condition (basic mode or HALT2 mode).

Write "1": HALT2 mode  
 Write "0": Basic mode  
 Read: Valid

When "1" is written to HLT2OP, the CPU will enter HALT2 mode when the HALT instruction is executed. When "0" is written, the CPU will enter basic mode.

Writing to HLT2OP is allowed only when CLGP[7:0] is set to "0b10010110".

At initial reset, HLT2OP is set to "0" (basic mode).

The following shows the operating status in HALT mode (basic mode and HALT2 mode) and SLEEP mode.

Table II.6.6 Operating Status in Standby Mode

Standby mode		Operating status	Reactivating factor
HALT mode	Basic mode	<ul style="list-style-type: none"> <li>The CPU clock is stopped. (CPU stop status)</li> <li>BCU clock is supplied. (BCU run status)</li> <li>DMA clock is not stopped. (DMA run status)</li> <li>Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop)</li> <li>The high-speed oscillation circuit maintains the status before entering HALT mode.</li> <li>The low-speed oscillation circuit maintains the status before entering HALT mode.</li> </ul>	<ul style="list-style-type: none"> <li>Reset, NMI</li> <li>Enabled (not masked) interrupt factors</li> </ul>
	HALT2 mode	<ul style="list-style-type: none"> <li>The CPU clock is stopped. (CPU stop status)</li> <li>BCU clock is stopped. (BCU stop status)</li> <li>DMA clock is stopped. (DMA stop status)</li> <li>Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop)</li> <li>The high-speed oscillation circuit maintains the status before entering HALT mode.</li> <li>The low-speed oscillation circuit maintains the status before entering HALT mode.</li> </ul>	A restart is possible only in the case of: <ul style="list-style-type: none"> <li>Reset, NMI</li> <li>Enabled (not masked) interrupt factors</li> </ul> Note, however, that an interrupt from a peripheral circuit can restart the CPU only when the operating clock is supplied to the peripheral circuit.
SLEEP mode		<ul style="list-style-type: none"> <li>The CPU clock is stopped. (CPU stop status)</li> <li>BCU clock is stopped. (BCU stop status)</li> <li>Clocks for the peripheral circuits are stopped.</li> <li>The high-speed oscillation circuit is stopped.</li> <li>The low-speed oscillation circuit maintains the status before entering SLEEP mode.</li> </ul>	<ul style="list-style-type: none"> <li>Reset, NMI</li> <li>Enabled (not masked) input port interrupt factors</li> <li>Clock timer interrupt when the low-speed oscillation circuit is being operated</li> </ul>

### CLGP7–CLGP0: Power-control register protection flag ([D][7:0]) / Power control protection register (0x4019E)

These bits remove the protection against writing to addresses 0x40180 and 0x40190.

Write "0b10010110": Write protection removed  
 Write other than the above: No operation (write-protected)  
 Read: Valid

Before writing to address 0x40180 or 0x40190, set CLGP[7:0] to "0b10010110" to remove the protection against writing to that address. This clearing of write protection is effective for only one writing, so the bits are cleared to "0b00000000" by one writing. Therefore, CLGP[7:0] must be set each time the protected address is written to.

At initial reset, CLGP is set to "0b00000000" (write-protected).

## Programming Notes

- (1) Immediately after the high-speed (OSC3) oscillation circuit is turned on, a certain period of time is required for oscillation to stabilize. To prevent the device from operating erratically, do not use the clock until its oscillation has stabilized.  
In particular, if the CPU is set in SLEEP mode during operation using the OSC3 clock, the high-speed (OSC3) oscillation circuit is turned off during in SLEEP mode and starts oscillating again after SLEEP mode is exited. To prevent the CPU from operating erratically at restart due to an unstable OSC3 clock, set a sufficient stabilization waiting time in 8-bit programmable timer 1 to turn on the oscillation stabilization waiting function after SLEEP mode is exited before entering SLEEP mode.
- (2) The oscillation circuit used for the CPU operating clock cannot be turned off.
- (3) The CPU operating clock can only be switched over when both the OSC3 and OSC1 oscillation circuits are on. Furthermore, when turning off an oscillation circuit that has become unnecessary as a result of the CPU operating clock switchover, be sure to use separate instructions for switchover and oscillation turnoff. If these two operations are processed simultaneously using one instruction, the CPU may operate erratically.
- (4) If the high-speed (OSC3) oscillation circuit is turned off, all peripheral circuits operated using the OSC3 clock will be inactive.
- (5) If the OSC3 clock is unnecessary, use the OSC1 clock to operate the CPU and turn the high-speed (OSC3) oscillation circuit off. This helps reduce current consumption.
- (6) In HALT mode, since the DMA and BCU clocks operate, if the next operation is performed in HALT mode, not HALT2 mode, with a setting of 0 in clock option register HLT2OP (D3/0x40190), that operation will be an unpredictable erroneous operation.  
If a DMA trigger occurs and DMA is invoked while the CPU is stopped after HALT mode execution, erroneous operation will result. Ensure that DMA is not invoked in HALT mode.  
In HALT2 mode, DMA is not invoked since the DMA and BCU clocks are stopped.
- (7) In the SLEEP mode, the oscillation circuit clock stops, and in the HALT2 mode, the clocks for peripheral circuits maintain the status before entering HALT2 (stop or run).  
When restarting from this state, interrupt input from a port can be used as a trigger.  
In this case, an interrupt will occur due to the input signal level even if edge interrupt is specified as an interrupt condition. The signal level to restart the CPU is as follows according to the signal edge selected:  
If a rising-edge interrupt is set, the CPU restarts when the input signal goes to a high level.  
If a falling-edge interrupt is set, the CPU restarts when the input signal goes to a low level.  
  
When a falling edge interrupt is selected to restart after the slp instruction is executed, the operation is as follows.  
If the interrupt port is already at a low level when the slp instruction is executed, the CPU enters SLEEP mode instantaneously and restarts immediately afterward.  
If the interrupt port is at a high level when the slp instruction is executed, the SLEEP mode continues until the port goes low.  
  
Therefore, design the system assuming that the CPU can restart normally due to the signal level at the interrupt port, not an edge interrupt, when restarting the CPU from HALT2 or SLEEP mode using a port input interrupt.

## II-6 CORE BLOCK: CLG (Clock Generator)

- (8) If the IC enters the debug mode through the connected S5U1C33000H (In-Circuit Debugger for S1C33 Family) when the OSC3 clock is divided by 2, 4, or 8 using the CLKDT[1:0] (D[7:6])/Power control register (0x40180) to generate the CPU clock (CPU\_CLK), the division ratio is automatically changed to 1/1. This may cause the CPU\_CLK frequency to exceed the range assumed. Also it affects the BCLK output clock as it is generated from CPU\_CLK. If the BCLK output clock frequency exceeds the access time condition or operating range of the devices driven with these clocks, debugging functions such as memory dump as well as program execution may not operate correctly. Therefore, prescribe remedies for malfunctions when debugging, for example, changing the number of wait cycles and other parameters in the BCU registers using the debugger, so that the program can be executed and debugged without problems even when the division ratio changes to 1/1.
- (9) When the base clock (CPU operating clock) is generated by dividing the source clock output from OSC3 or PLL by a value (2, 4, or 8) specified using CLKDT[1:0] (D[7:6])/Power control register (0x40180), the peripheral circuit clocks must be set lower than the base clock frequency using the prescaler. If the peripheral circuit clock frequency is equal to or higher than the base clock frequency, the peripheral circuit does not operate normally.



## II-7 POWER-DOWN CONTROL

This section describes the controls used to reduce power consumption of the device.

### Points on power saving

The current consumption of the device varies greatly with the CPU's operation mode, the system clocks used, and the peripheral circuits operated.

Current consumption	low←						→high
CPU/BCU	SLEEP	HALT2	Operating	HALT2	HALT(basic)	OSC3	Operating
System clock	-		OSC1	OSC3	OSC3	OSC3	OSC3
OSC3 oscillation circuit	OFF	OFF	OFF	ON	ON	ON	ON
Prescaler/peripheral circuit	STOP						RUN

To reduce power consumption of the device, it is important that as many unnecessary circuits as possible be turned off. In particular, peripheral circuits operating at a fast-clock rate consume a large amount of current, so design the program so that these circuits are turned off whenever unnecessary.

### Power-saving in standby modes

When CPU processing is unnecessary, such as when waiting for an interrupt from key entries or peripheral circuits, place the device in standby mode to reduce current consumption.

Standby mode	Method to enter the mode	Circuits/functions stopped
Basic HALT mode	Execute the halt instruction after setting HLT2OP (D3)/Clock option register (0x40190) to "0". When the #BUSREQ signal is asserted from an external bus master while SEPD (D1)/Bus control register (0x4812E) = "1".	CPU (DMA cannot be used.)
HALT2 mode	Execute the halt instruction after setting HLT2OP to "1".	CPU, BCU, bus clock, and DMA
SLEEP mode	Execute the slp instruction.	CPU, BCU, bus clock, DMA, high-speed (OSC3) oscillation circuit, prescaler, and peripheral circuits that use the prescaler output clocks

HLT2OP (D3)/Clock option register (0x40190) that is used to select a HALT mode is set to "0" (basic HALT mode) at initial reset.

- Notes:**
- The standby mode is cleared by interrupt generation (except for the basic HALT mode, which is set using an external bus master). Therefore, before entering standby mode, set the related registers to allow an interrupt to be used to clear the standby mode to be generated.
  - When clearing the standby mode with an interrupt from port input, the interrupt operates as a level interrupt regardless of the interrupt trigger setting. When edge trigger is set for the interrupt trigger, attention must be paid to the port level during standby mode.

The low-speed (OSC1) oscillation circuit and clock timer continue operating even during SLEEP mode. If they are unnecessary, these circuits can also be turned off.

Function	Control bit	"1"	"0"	Default
Low-speed (OSC1) oscillation ON/OFF control	SOSC1(D0)/Power control register(0x40180)	ON	OFF	ON

### Switching over the system clocks

Normally, the system is clocked by the high-speed (OSC3) oscillation clock. If high-speed operation is unnecessary, switch the system clock to the low-speed (OSC1) oscillation clock and turn off the high-speed (OSC3) oscillation circuit. This helps to reduce current consumption.

Even during operation using the high-speed (OSC3) oscillation clock, power reduction can also be achieved through the use of a system clock derived from the OSC3 clock by dividing it (1/1, 1/2, 1/4, or 1/8).

## II-7 CORE BLOCK: POWER-DOWN CONTROL

Function	Control bit	"1"	"0"	Default
System clock switch over	CLKCHG(D2)/Power control register(0x40180)	OSC3	OSC1	OSC3
High-speed (OSC3) oscillation ON/OFF control	SOSC3(D1)/Power control register(0x40180)	ON	OFF	ON
System clock division ratio selection	CLKDT(D[7:6])/Power control register(0x40180)	"11" = 1/8 "10" = 1/4 "01" = 1/2 "00" = 1/1		1/1

### Turning off the prescaler and peripheral circuits

Current consumption can be reduced by turning off the peripheral circuits operating at high speed as much as possible. The peripheral circuits are as follows.

#### 1) Peripheral circuits using the clock generated by the prescaler

- 16-bit programmable timers 0 to 5 (watchdog timer)
- 8-bit programmable timers 0 to 5 (serial interface)
- A/D converter

#### 2) Peripheral circuits using the clock (source clock for prescaler) supplied to the prescaler

- 16-bit programmable timers 0 to 5 (watchdog timer)
- 8-bit programmable timers 0 to 5
- A/D converter
- Serial interface
- Input/output ports

If none of all circuits of the above 1) and 2) need to be used, turn off the prescaler. If the circuit of the above 1) or 2) need to be used, do not turn off the prescaler. When operation of the prescaler is stopped, the clock supply to the circuits of the above 2) stops. When some these circuits of the above 1) need to be used, turn off all other unnecessary circuits and stop the clock supply from the prescaler to those circuits.

The prescaler operating control and the clock supply control bits for each peripheral circuit are shown in the table below.

Function	Control bit	"1"	"0"	Default
Prescaler ON/OFF	PSCON(D5)/Power control register(0x40180)	ON	OFF	ON
16-bit timer 0 clock control	P16TON0(D3)/16-bit timer 0 clock control register(0x40147)	ON	OFF	OFF
16-bit timer 0 Run/Stop	PRUN0(D0)/16-bit timer 0 control register(0x48186)	RUN	STOP	STOP
16-bit timer 1 clock control	P16TON1(D3)/16-bit timer 1 clock control register(0x40148)	ON	OFF	OFF
16-bit timer 1 Run/Stop	PRUN1(D0)/16-bit timer 1 control register(0x4818E)	RUN	STOP	STOP
16-bit timer 2 clock control	P16TON2(D3)/16-bit timer 2 clock control register(0x40149)	ON	OFF	OFF
16-bit timer 2 Run/Stop	PRUN2(D0)/16-bit timer 2 control register(0x48196)	RUN	STOP	STOP
16-bit timer 3 clock control	P16TON3(D3)/16-bit timer 3 clock control register(0x4014A)	ON	OFF	OFF
16-bit timer 3 Run/Stop	PRUN3(D0)/16-bit timer 3 control register(0x4819E)	RUN	STOP	STOP
16-bit timer 4 clock control	P16TON4(D3)/16-bit timer 4 clock control register(0x4014B)	ON	OFF	OFF
16-bit timer 4 Run/Stop	PRUN4(D0)/16-bit timer 4 control register(0x481A6)	RUN	STOP	STOP
16-bit timer 5 clock control	P16TON5(D3)/16-bit timer 5 clock control register(0x4014C)	ON	OFF	OFF
16-bit timer 5 Run/Stop	PRUN5(D0)/16-bit timer 5 control register(0x481AE)	RUN	STOP	STOP
8-bit timer 0 clock control	P8TON0(D3)/8-bit timer 0/1 clock control register(0x4014D)	ON	OFF	OFF
8-bit timer 0 Run/Stop	Ptrun0(D0)/8-bit timer 0 control register(0x40160)	RUN	STOP	STOP
8-bit timer 1 clock control	P8TON1(D7)/8-bit timer 0/1 clock control register(0x4014D)	ON	OFF	OFF
8-bit timer 1 Run/Stop	Ptrun1(D0)/8-bit timer 1 control register(0x40164)	RUN	STOP	STOP
8-bit timer 2 clock control	P8TON2(D3)/8-bit timer 2/3 clock control register(0x4014E)	ON	OFF	OFF
8-bit timer 2 Run/Stop	Ptrun2(D0)/8-bit timer 2 control register(0x40168)	RUN	STOP	STOP
8-bit timer 3 clock control	P8TON3(D7)/8-bit timer 2/3 clock control register(0x4014E)	ON	OFF	OFF
8-bit timer 3 Run/Stop	Ptrun3(D0)/8-bit timer 3 control register(0x4016C)	RUN	STOP	STOP
8-bit timer 4 clock control	P8TON4(D3)/8-bit timer 4/5 clock control register(0x40145)	ON	OFF	OFF
8-bit timer 4 Run/Stop	Ptrun4(D0)/8-bit timer 4 control register(0x40174)	RUN	STOP	STOP
8-bit timer 5 clock control	P8TON5(D7)/8-bit timer 4/5 clock control register(0x40145)	ON	OFF	OFF
8-bit timer 5 Run/Stop	Ptrun5(D0)/8-bit timer 5 control register(0x40178)	RUN	STOP	STOP
A/D converter clock control	PSONAD(D3)/A/D clock control register(0x4014F)	ON	OFF	OFF
A/D conversion enable	ADE(D2)/A/D enable register(0x40244)	RUN	STOP	STOP

The same clock source must be used for the prescaler operating clock and the CPU operating clock. Therefore, when operating the CPU in low-speed with the OSC1 clock, the prescaler input clock must be switched according to the CPU operating clock. In this case, in order to prevent a malfunction in the peripheral circuit, the prescaler should be turned off before switching the CPU operating clock. After the CPU operating clock has been switched, switch the prescaler operating clock and then turn the prescaler on.

Function	Control bit	"1"	"0"	Default
Prescaler operating clock switch over	PSCDT0(D0)/Prescaler clock select register(0x40181)	OSC1	OSC3/ PLL	OSC3/ PLL

THIS PAGE IS BLANK.

## II-8 DBG (Debug Unit)

### Debug Circuit

The C33 Core Block has a built-in debug circuit.

This functional block is provided to simply realize an advanced software development environment.

**Note:** The debug circuit does not work during normal operation. To construct a software development environment using the debug circuit, the S5U1C33000H/S5U1C33001H (In-Circuit Debugger for S1C33 Family) is separately required.

### I/O Pins of Debug Circuit

Six pins used to exclusively connect the S5U1C33000H/S5U1C33001H (In-Circuit Debugger for S1C33 Family) are reserved for the debug circuit. The I/O voltage level of these pins is 3.3 V.

Table II.8.1 lists the I/O pins of the debug circuit.

Table II.8.1 I/O Pins of Debug Circuit

Pin name	I/O	Pull-up	Function
DCLK(P14/FOSC1)	O	–	Clock output for debugging / I/O port / OSC1 clock output
DST2(P12/EXCL2/T8UF2)	O	–	Status output 2 for debugging / I/O port / 16-bit timer 2 event counter input / 8-bit timer 2 output
DST1(P11/EXCL1/T8UF1)	O	–	Status output 1 for debugging / I/O port / 16-bit timer 1 event counter input / 8-bit timer 1 output
DST0(P10/EXCL0/T8UF0)	O	–	Status output 0 for debugging / I/O port / 16-bit timer 0 event counter input / 8-bit timer 0 output
DPCO(P13/EXCL3/T8UF3)	O	–	PC output for debugging / I/O port / 16-bit timer 3 event counter input / 8-bit timer 3 output
DSIO	I/O	Pull-up	Serial I/O for debugging

The DCLK, DST[2:0] and DPCO outputs are extended functions of the I/O port pins P14, P1[2:0] and P13, respectively. At initial reset, these pins are set as debug signal outputs.

If the debug circuit is not used, these pins can be used for I/O ports or the redefined peripheral circuits by writing "0" to CFEX[1:0] (D[1:0]) / Port function extension register (0x402DF). Refer to "I/O Ports (P Ports)" for the pin functions.

**Note:** When these pins are set as debug signal outputs, only the S5U1C33000H/S5U1C33001H (In-Circuit Debugger for S1C33 Family) can be connected to these pins. Leave these pins open if the S5U1C33000H/S5U1C33001H is not connected. For connecting the S5U1C33000H/S5U1C33001H, refer to the "S5U1C33000H Manual (S1C33 Family In-Circuit Debugger)" or "S5U1C33001H Manual (S1C33 Family In-Circuit Debugger)".

Furthermore, the pin status is fixed as shown below after a user reset.

DCLK = "1"

DST2 = "0"

DST1 = "1"

DST0 = "1"

DPCO = "1"

DSIO = "1" (input)

THIS PAGE IS BLANK.

S1C33301 Technical Manual

# **III PERIPHERAL BLOCK**





# III-1 INTRODUCTION

The C33 peripheral block consists of a prescaler, six 8-bit programmable timer channels, six 16-bit programmable timer channels including watchdog timer and event counter functions, four serial interface channels, input and I/O ports, a low-speed (OSC1) oscillation circuit, and a clock timer.

The S1C33301 comes with extended peripheral circuits, chip ID and pin control registers, serial interface with FIFO, and extended I/O ports.

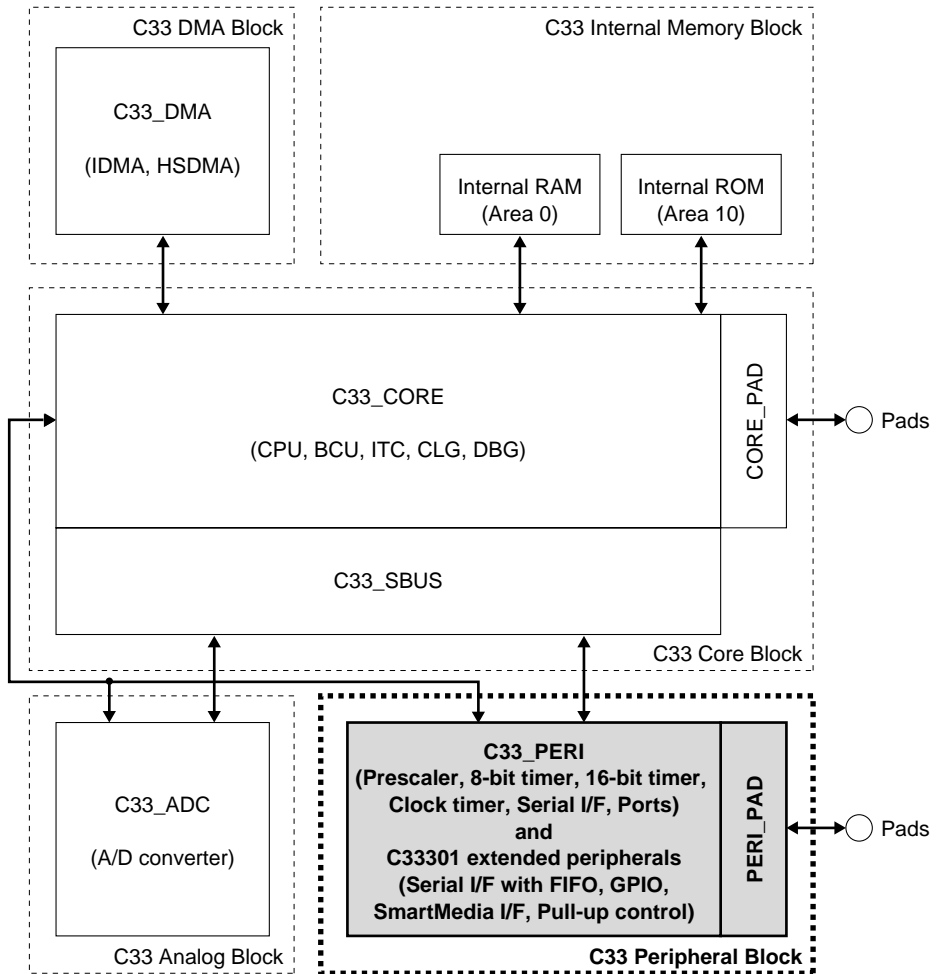


Figure III.1.1 Peripheral Block

**Note:** Internal ROM is not provided in the S1C33301.

THIS PAGE IS BLANK.

## III-2 CHIP ID / PIN STATUS CONTROL REGISTERS

This section explains the chip ID registers and the pull-up/low drive control registers for bus and I/O pins.

### **Chip ID register**

---

The S1C33301 has four built-in registers shown below that allow the application software to identify CPU type, model, and chip version.

#### **Core ID register (0x300000)**

This register provides an 8-bit ID code that indicates the chip core type.

ID	Chip Core Type
0x02	C33 standard macro core (C33 STD core CPU)
0x03	C33 mini-macro core
0x04	C33 advanced macro core

The S1C33301 has adopted the C33 standard macro core, so the chip core ID is 0x02.

#### **Product series ID register (0x300001)**

This register provides an 8-bit ID code that indicates the product series of the S1C33 Family.

ID	Product Series
0x03	S1C333xx Series
0x15	S1C33Lxx Series

The product series ID of the S1C33301 is 0x03.

#### **Model ID register (0x300002)**

This register provides an 8-bit ID code that indicates the model.  
The model ID of the S1C33301 is 0x01.

#### **Version register (0x300003)**

This register provides a 4-bit ID code that indicates the version number.  
0x00 represents version 1.0.

## Pull-up Control Resistors

The S1C33301 input/output pins have a pull-up resistor that can be connected/disconnected to/from the pin by software control, except some special pins. The pins are divided into groups and each group has a pull-up control bit to select whether the pull-up resistors of the pins that belong to the group are used or not.

Table III.2.1 lists the correspondence between the register/control bits and pins.

Table III.2.1 Correspondence Between Pull-up Control Bits and Pins

Control register	Control bit	Pin
Bus signal pull-up control register (0x300F00)	PUPCAS (D3)	#LCAS (P40/A24), #HCAS (P41/A25), BCLK (P60)
	PUPCE (D2)	#CE10EX, #CE[9:4] (P5[5:0])
	PUPAD (D1)	A[23:18] (P4[2:7]), A[17:0]
	PUPRW (D0)	#WRH, #WRL, #RD
Input port pull-up control register (0x300F02)	PUPK6H (D2)	K6[7:4]
	PUPK6L (D1)	K6[3:0]
	PUPK5 (D0)	K5[4:0]
I/O port pull-up control register (0x300F04)	PUPP6 (D6)	P6[3:0]
	PUPP3 (D3)	P3[5:0]
	PUPP2 (D2)	P2[7:0]
	PUPP1 (D1)	P1[6:0]
	PUPP0 (D0)	P0[7:0]

At initial reset, the pull-up control bits are set to "1" and the pins are pulled up. When not using pull-up resistors, set the corresponding pull-up control bits to "0".

- Notes:**
- The pull-up control bit is effective in both cases when the pin is used for the external bus and when used for the on-chip peripheral circuit or general-purpose I/O port.
  - The PFBGA-121pin package does not have K67–K64 and P63–P61 pins. However, PUPK6H and PUPP6 should not be set to "0" (no pull-up), since the chip is equipped with port circuits.

## Driving Bus Signals Low

The S1C33301 can drive the bus signal output pins forcibly low using a control register. This function is useful when turning off the power of the external device connected to the bus.

Table III.2.2 lists the correspondence between the register/control bits and bus signals.

Table III.2.2 Correspondence Between Low-Drive Control Bits and Bus Signals

Control register	Control bit	Bus signal
Bus signal low-drive control register (0x300F01)	LDRVDB (D4)	D[15:0]
	LDRVCAS (D3)	#LCAS (A24), #HCAS (A25), BCLK
	LDRVCE (D2)	#CE10EX, #CE[9:4]
	LDRVAD (D1)	A[23:0]
	LDRV RW (D0)	#WRH, #WRL, #RD

When the control bit is set to "1", the corresponding bus signal goes low. When the control bit is set to "0", the signal control goes back to the BCU.

- Notes:**
- The low-drive control bit is disabled when the pin is used as the general-purpose I/O port (Pxx).
  - If the above signals are forcibly driven low when the CPU is running by the instructions fetched from an external memory, the CPU will not be able to run after that point. To drive the signals low, the CPU must be running with the program stored in the internal RAM.

## ***Setting BCU to Access to the Registers***

---

The chip ID, pull-up control and low-drive control registers are mapped into Area 6 addresses 0x300000 to 0x300F04. Therefore, in order for the registers to be accessed, the BCU register for Area 6 must be set up in accordance with the procedure described below.

1. A6IO (D9) / Access control register (0x48132) = "1"  
This sets Area 6 so that the internal device can be accessed.
2. A6EC (D1) / Access control register (0x48132) = "0"  
This sets Area 6 so that it can be accessed in the little endian format.
3. A6WT[2:0] (D[A:8]) / Areas 6–4 setup register (0x4812A) = "001"  
This sets Area 6 so that it can be accessed with 1 wait state inserted.

## I/O Memory for Chip ID and Pin Status Control

Table III.2.3 shows the chip ID and pin status control bits.

Table III.2.3 Chip ID and Pin Status Control Bits

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Core ID register	0300000 (B)	D7	CID7	Chip core ID	0x02	0	R			
		D6	CID6	0x02: C33 standard macro core						
		D5	CID5	0x03: C33 mini-macro core						
		D4	CID4	0x04: C33 advanced macro core						
		D3	CID3							
		D2	CID2							
		D1	CID1							
		D0	CID0							
		D0	CID0							
Product series ID register	0300001 (B)	D7	MID7	Product series ID	0x03	0	R			
		D6	MID6	0x03: S1C333xx						
		D5	MID5	0x15: S1C33Lxx						
		D4	MID4							
		D3	MID3							
		D2	MID2							
		D1	MID1							
		D0	MID0							
		D0	MID0							
Model ID register	0300002 (B)	D7	NAME7	Model ID	0x01	0	R			
		D6	NAME6	0x01: S1C33301						
		D5	NAME5							
		D4	NAME4							
		D3	NAME3							
		D2	NAME2							
		D1	NAME1							
		D0	NAME0							
		D0	NAME0							
Version register	0300003 (B)	D7	VER3	Version code	0x00	0	R			
		D6	VER2	0x00: version 1.0						
		D5	VER1							
		D4	VER0							
		D3-0	-	reserved					-	-
Bus signal pull-up control register	0300F00 (B)	D7-4	-	reserved	-	-	-	0 when being read.		
		D3	PUPCAS	#LCAS, #HCAS, BCLK (P40, P41, P60) pull-up	1	Pulled up	0	No pull-up	1	R/W
		D2	PUPCE	#CE10-#CE4 (P55-P50) pull-up					1	R/W
		D1	PUPAD	A23-A0 (P42-P47) pull-up					1	R/W
		D0	PUPRW	#WRH, #WRL, #RD pull-up					1	R/W
Bus signal low drive control register	0300F01 (B)	D7-5	-	reserved	-	-	-	0 when being read.		
		D4	LDRVDB	D15-D0 low drive	1	Low drive	0	Normal output	0	R/W
		D3	LDRVCAS	#LCAS, #HCAS, BCLK low drive					0	R/W
		D2	LDRVCE	#CE10-#CE4 low drive					0	R/W
		D1	LDRVAD	A23-A0 low drive					0	R/W
		D0	LDRVRW	#WRH, #WRL, #RD low drive					0	R/W
Input port pull-up control register	0300F02 (B)	D7-3	-	reserved	-	-	-	0 when being read.		
		D2	PUPK6H	K67-K64 pull-up *1	1	Pulled up	0	No pull-up	1	R/W
		D1	PUPK6L	K63-K60 pull-up					1	R/W
		D0	PUPK5	K54-K50 pull-up					1	R/W
I/O port pull-up control register	0300F04 (B)	D7	-	reserved	-	-	-	0 when being read.		
		D6	PUPP6	P63-P60 pull-up *1	1	Pulled up	0	No pull-up	1	R/W
		D5-4	-	reserved	-	-	-	-	0 when being read.	
		D3	PUPP3	P35-P30 pull-up	1	Pulled up	0	No pull-up	1	R/W
		D2	PUPP2	P27-P20 pull-up					1	R/W
		D1	PUPP1	P16-P10 pull-up					1	R/W
		D0	PUPP0	P07-P00 pull-up					1	R/W

\*1 In case of PFBGA-121pin package, PUPK6H (K67-K64 pull-up control) and PUPP6 (P63-P60 pull-up control) must be set to "1" (pulled up).

**CID7–CID0:** Chip core ID (D[7:0]) / Core ID register (0x300000)

These bits provide a code (0x02 = C33 standard macro core) to identify the chip core type.

These bits are read-only, so writing to them is ignored.

**MID7–MID0:** Product series ID (D[7:0]) / Product series ID register (0x300001)

These bits provide a code (0x03 = S1C333xx series) to identify the S1C33 Family product series.

These bits are read-only, so writing to them is ignored.

**NAME7–NAME0:** Model ID (D[7:0]) / Model ID register (0x300002)

These bits provide a code (0x01 = S1C33301) to identify the model in the product series shown by MID[7:0].

These bits are read-only, so writing to them is ignored.

**VER3–VER0:** Version code (D[7:4]) / Version register (0x300003)

These bits provide a version code.

0x00 represents version 1.0.

These bits are read-only, so writing to them is ignored.

**PUPCAS:** #LCAS, #HCAS, BCLK pull-up control (D3) / Bus signal pull-up control register (0x300F00)**PUPCE:** #CE10–#CE4 pull-up control (D2) / <Bus signal pull-up control register (0x300F00)**PUPAD:** A23–A0 pull-up control (D1) / Bus signal pull-up control register (0x300F00)**PUPRW:** #WRH, #WRL, #RD pull-up control (D0) / Bus signal pull-up control register (0x300F00)

Controls the pull-up resistors at the bus signal pins.

Write "1": Pulled up

Write "0": No pull-up

Read: Valid

Each bit corresponds to the pins shown below.

PUPCAS #LCAS (P40/A24), #HCAS (P41/A25), BCLK (P60)

PUPCE #CE10EX, #CE[9:4] (P5[5:0])

PUPAD A[23:18] (P4[2:7]), A[17:0]

PUPRW #WRH, #WRL, #RD

When the pull-up control bit is set to "1", the corresponding pin is pulled up to high. When it is set to "0", the pin is not pulled up.

At initial reset, these bits are set to "1" (pulled up).

**PUPK6H:** K67–K64 pull-up control (D2) / Input port pull-up control register (0x300F02)**PUPK6L:** K63–K60 pull-up control (D1) / Input port pull-up control register (0x300F02)**PUPK5:** K54–K50 pull-up control (D0) / Input port pull-up control register (0x300F02)

Controls the pull-up resistors at the input port pins.

Write "1": Pulled up

Write "0": No pull-up

Read: Valid

When the pull-up control bit is set to "1", the corresponding pin is pulled up to high. When it is set to "0", the pin is not pulled up.

At initial reset, these bits are set to "1" (pulled up).

**Note:** The PFBGA-121pin package does not have K67–K64 pins. However, PUPK6H should not be set to "0" (no pull-up), since the chip is equipped with port circuits.

### III-2 PERIPHERAL BLOCK: CHIP ID / PIN STATUS CONTROL REGISTERS

---

**PUPP6:** P63–P60 pull-up control (D6) / I/O port pull-up control register (0x300F04)

**PUPP3:** P35–P30 pull-up control (D3) / I/O port pull-up control register (0x300F04)

**PUPP2:** P27–P20 pull-up control (D2) / I/O port pull-up control register (0x300F04)

**PUPP1:** P16–P10 pull-up control (D1) / I/O port pull-up control register (0x300F04)

**PUPP0:** P07–P00 pull-up control (D0) / I/O port pull-up control register (0x300F04)

---

Controls the pull-up resistors at the I/O port pins.

Write "1": Pulled up

Write "0": No pull-up

Read: Valid

When the pull-up control bit is set to "1", the corresponding pin is pulled up to high. When it is set to "0", the pin is not pulled up.

At initial reset, these bits are set to "1" (pulled up).

**Note:** The PFBGA-121pin package does not have P63–P61 pins. However, PUPP6 should not be set to "0" (no pull-up), since the chip is equipped with port circuits.

---

**LDRVDB:** D15–D0 low drive control (D4) / Bus signal low-drive control register (0x300F01)

**LDRVCAS:** #LCAS, #HCAS, BCLK Low drive control (D3) / Bus signal low-drive control register (0x300F01)

**LDRVCE:** #CE10–#CE4 low drive control (D2) / Bus signal low-drive control register (0x300F01)

**LDRVAD:** A23–A0 low drive control (D1) / Bus signal low-drive control register (0x300F01)

**LDRVRW:** #WRH, #WRL, #RD low drive control (D0) / Bus signal low-drive control register (0x300F01)

---

Drives the bus signals forcibly low.

Write "1": Low drive

Write "0": Normal output

Read: Valid

Each bit corresponds to the signals shown below.

LDRVDB D[15:0]

LDRVCAS #LCAS (A24), #HCAS (A25), BCLK

LDRVCE #CE10EX, #CE[9:4]

LDRVAD A[23:0]

LDRVRW #WRH, #WRL, #RD

When the low-drive control bit is set to "1", the corresponding signal is forcibly driven low. When it is set to "0", the signal is controlled by the BCU normally.

At initial reset, these bits are set to "0" (Normal output).

**Notes:**

- The low-drive control bit is disabled when the pin is used as the general-purpose I/O port (Pxx).

- If the above signals are forcibly driven low when the CPU is running by the instructions fetched from an external memory, the CPU will not be able to run after that point. To drive the signals low, the CPU must be running with the program stored in the internal RAM.



## III-3 PRESCALER

### Configuration of Prescaler

The prescaler divides the source clock (OSC3/PLL output clock or OSC1 clock) to generate the clocks for the internal peripheral circuits. The prescaler division ratio can be selected for each peripheral circuit in a program. A clock control circuit to control the clock supply to each peripheral circuit is also included.

The following are the peripheral circuits that use the output clock:

- 16-bit programmable timers 5 to 0 (and watchdog timer)
- 8-bit programmable timers 5 to 0 (and serial interface)
- A/D converter

Figure III.3.1 shows the configuration of the prescaler.

For details on control of each peripheral circuit, refer to each corresponding section in this manual.

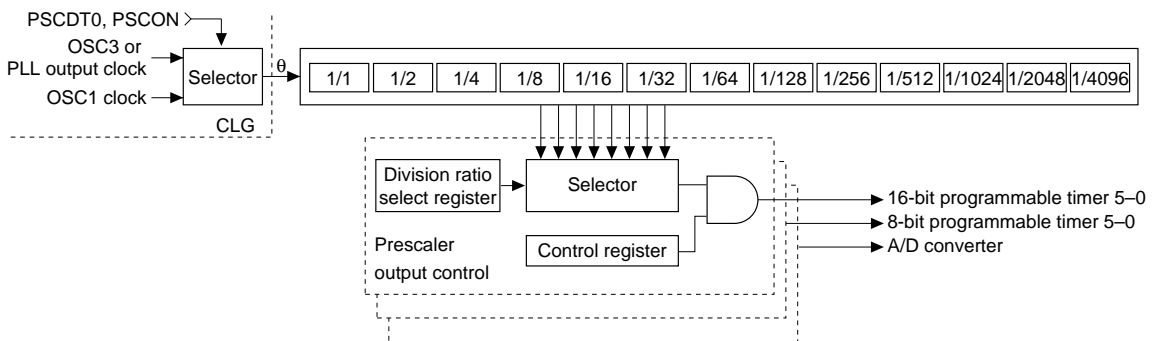


Figure III.3.1 Configuration of Prescaler and Clock Control Circuit

### Source Clock

The source clock for the prescaler can be selected using PSCDT0 (D0) / Prescaler clock select register (0x40181). When PSCDT0 = "0", the OSC3 clock (when the PLL is not used) or the PLL output clock (when the PLL is used) is selected.

When PSCDT0 = "1", the OSC1 clock (typ. 32 kHz) is selected.

At initial reset, the OSC3/PLL output clock is selected.

**Note:** For the prescaler clock, the clock source same as the CPU operating clock must be selected.

For details on how to control the oscillation circuit and CPU operating clock, refer to "CLG (Clock Generator)".

At initial reset, the OSC3 clock is selected.

The source clock is supplied to the prescaler by writing "1" to PSCON (D5) / Power control register (0x40180). At initial reset, PSCON is set to "1", so the prescaler is in an operating state. If all of said peripheral circuits can be turned off and the peripheral circuits (e.g., 16-bit programmable timers (watchdog timer), 8-bit programmable timers, A/D converter, serial interface, and ports) that use the prescaler input clock (the source clock for prescaler) can be turned off, stop the prescaler by writing "0" to PSCON. This helps to reduce current consumption.

## Selecting Division Ratio and Output Control for Prescaler

The prescaler has registers for selecting the division ratio and clock output control separately for each peripheral circuit described above, allowing each peripheral circuit to be controlled.

The prescaler's division ratio can be selected from among eight ratios set for each peripheral circuit through the use of the division ratio selection bits. The divided clock is output to the corresponding peripheral circuit by writing "1" to the clock control bit.

Table III.3.1 Control Bits of the Clock Control Registers

Peripheral circuit	Division ratio selection bit		Clock control bit
16-bit programmable timer 0	P16TS0[2:0]	(D[2:0]/0x40147) *1	P16TON0 (D3/0x40147)
16-bit programmable timer 1	P16TS1[2:0]	(D[2:0]/0x40148) *1	P16TON1 (D3/0x40148)
16-bit programmable timer 2	P16TS2[2:0]	(D[2:0]/0x40149) *1	P16TON2 (D3/0x40149)
16-bit programmable timer 3	P16TS3[2:0]	(D[2:0]/0x4014A) *1	P16TON3 (D3/0x4014A)
16-bit programmable timer 4	P16TS4[2:0]	(D[2:0]/0x4014B) *1	P16TON4 (D3/0x4014B)
16-bit programmable timer 5	P16TS5[2:0]	(D[2:0]/0x4014C) *1	P16TON5 (D3/0x4014C)
8-bit programmable timer 0	P8TS0[2:0]	(D[2:0]/0x4014D) *2	P8TON0 (D7/0x4014D)
8-bit programmable timer 1	P8TS1[2:0]	(D[6:4]/0x4014D) *3	P8TON1 (D7/0x4014D)
8-bit programmable timer 2	P8TS2[2:0]	(D[2:0]/0x4014E) *4	P8TON2 (D3/0x4014E)
8-bit programmable timer 3	P8TS3[2:0]	(D[6:4]/0x4014E) *2	P8TON3 (D7/0x4014E)
8-bit programmable timer 4	P8TS4[2:0]	(D[2:0]/0x40145) *4	P8TON4 (D3/0x40145)
8-bit programmable timer 5	P8TS5[2:0]	(D[6:4]/0x40145) *2	P8TON5 (D7/0x40145)
A/D converter	PSAD[2:0]	(D[2:0]/0x4014F) *2	PSONAD (D3/0x4014F)

\*1 to \*4: See Table III.3.2.

Table III.3.2 Division Ratio

Bit setting	7	6	5	4	3	2	1	0
*1	$\theta/4096$	$\theta/1024$	$\theta/256$	$\theta/64$	$\theta/16$	$\theta/4$	$\theta/2$	$\theta/1$
*2	$\theta/256$	$\theta/128$	$\theta/64$	$\theta/32$	$\theta/16$	$\theta/8$	$\theta/4$	$\theta/2$
*3	$\theta/4096$	$\theta/2048$	$\theta/1024$	$\theta/512$	$\theta/256$	$\theta/128$	$\theta/64$	$\theta/32$
*4	$\theta/4096$	$\theta/2048$	$\theta/64$	$\theta/32$	$\theta/16$	$\theta/8$	$\theta/4$	$\theta/2$

( $\theta$  = Source clock selected by PSCDT0)

Current consumption can be reduced by turning off the clock output to the peripheral circuits that are unused among those listed above.

**Note:** In the following cases, the prescaler output clock may contain a hazard:

- If, when a clock is output, its division ratio is changed
- When the clock output is switched between on and off
- When the oscillation circuit is turned off or the CPU operating clock is switched over

Before performing these operations, make sure the 16-bit and 8-bit programmable timers and the A/D converter are turned off.

## Source Clock Output to 8-Bit Programmable Timer

In addition to the divided clock, the prescaler can output the source clock directly to the 8-bit programmable timer. This function can be selected for each 8-bit timer using P8TPCKx bit.

8-bit timer 0: P8TPCK0 (D0) / 8-bit timer clock select register (0x40146)

8-bit timer 1: P8TPCK1 (D1) / 8-bit timer clock select register (0x40146)

8-bit timer 2: P8TPCK2 (D2) / 8-bit timer clock select register (0x40146)

8-bit timer 3: P8TPCK3 (D3) / 8-bit timer clock select register (0x40146)

8-bit timer 4: P8TPCK4 (D0) / 8-bit timer 4/5 clock select register (0x40140)

8-bit timer 5: P8TPCK5 (D1) / 8-bit timer 4/5 clock select register (0x40140)

When P8TPCKx is set to "1", the prescaler input clock ( $\theta/1$ ) is selected for the 8-bit timer x operating clock.

The clock output is controlled by the P8TONx bit even if P8TPCKx is set to "1".

When P8TPCKx is "0", the divided clock that is selected by P8TSx[2:0] will be output to the 8-bit timer x.

At initial reset, P8TPCKx is set to "0" and P8TSx[2:0] becomes effective.

## I/O Memory of Prescaler

Table III.3.3 shows the control bits of the prescaler.

Table III.3.3 Control Bits of Prescaler

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
8-bit timer 4/5 clock select register	0040140 (B)	D7-2	–	reserved	–	–	–	0 when being read.			
		D1	P8TPCK5	8-bit timer 5 clock selection	1 0/1	0 Divided clk.	0	R/W	0: selected by		
		D0	P8TPCK4	8-bit timer 4 clock selection	1 0/1	0 Divided clk.	0	R/W	Prescaler clock select register (0x40181)		
8-bit timer 4/5 clock control register	0040145 (B)	D7	P8TON5	8-bit timer 5 clock control	1 On	0 Off	0	R/W			
		D6	P8TS52	8-bit timer 5 clock division ratio selection	1 1 1	0/256	0	R/W	0: selected by Prescaler clock select register (0x40181)		
		D5	P8TS51		1 1 0	0/128	0	R/W			
		D4	P8TS50		1 0 1	0/64	0	R/W			
					1 0 0	0/32					
					0 1 1	0/16				8-bit timer 5 can generate the clock for the serial I/F Ch.3.	
				0 1 0	0/8						
				0 0 1	0/4						
				0 0 0	0/2						
				D3	P8TON4	8-bit timer 4 clock control	1 On	0 Off	0	R/W	
				D2	P8TS42	8-bit timer 4 clock division ratio selection	1 1 1	0/4096	0	R/W	0: selected by Prescaler clock select register (0x40181)
		D1	P8TS41	1 1 0	0/2048		0	R/W			
		D0	P8TS40	1 0 1	0/64		0	R/W			
				1 0 0	0/32						
				0 1 1	0/16				8-bit timer 4 can generate the clock for the serial I/F Ch.2.		
				0 1 0	0/8						
				0 0 1	0/4						
				0 0 0	0/2						
8-bit timer 0-3 clock select register	0040146 (B)	D7-4	–	reserved	–	–	–	0 when being read.			
		D3	P8TPCK3	8-bit timer 3 clock selection	1 0/1	0 Divided clk.	0	R/W	0: selected by		
		D2	P8TPCK2	8-bit timer 2 clock selection	1 0/1	0 Divided clk.	0	R/W	Prescaler clock select		
		D1	P8TPCK1	8-bit timer 1 clock selection	1 0/1	0 Divided clk.	0	R/W	register (0x40181)		
		D0	P8TPCK0	8-bit timer 0 clock selection	1 0/1	0 Divided clk.	0	R/W			
16-bit timer 0 clock control register	0040147 (B)	D7-4	–	reserved	–	–	–	0 when being read.			
		D3	P16TON0	16-bit timer 0 clock control	1 On	0 Off	0	R/W			
		D2	P16TS02	16-bit timer 0 clock division ratio selection	P16TS0[2:0] Division ratio		0	R/W	0: selected by Prescaler clock select register (0x40181)		
		D1	P16TS01		1 1 1	0/4096	0				
					1 1 0	0/1024	0				
					1 0 1	0/256					
					1 0 0	0/64					
				0 1 1	0/16			16-bit timer 0 can be used as a watchdog timer.			
		0 1 0	0/4								
		0 0 1	0/2								
				0 0 0	0/1						
16-bit timer 1 clock control register	0040148 (B)	D7-4	–	reserved	–	–	–	0 when being read.			
		D3	P16TON1	16-bit timer 1 clock control	1 On	0 Off	0	R/W			
		D2	P16TS12	16-bit timer 1 clock division ratio selection	P16TS1[2:0] Division ratio		0	R/W	0: selected by Prescaler clock select register (0x40181)		
		D1	P16TS11		1 1 1	0/4096	0				
					1 1 0	0/1024	0				
					1 0 1	0/256					
					1 0 0	0/64					
				0 1 1	0/16						
				0 1 0	0/4						
				0 0 1	0/2						
				0 0 0	0/1						
16-bit timer 2 clock control register	0040149 (B)	D7-4	–	reserved	–	–	–	0 when being read.			
		D3	P16TON2	16-bit timer 2 clock control	1 On	0 Off	0	R/W			
		D2	P16TS22	16-bit timer 2 clock division ratio selection	P16TS2[2:0] Division ratio		0	R/W	0: selected by Prescaler clock select register (0x40181)		
		D1	P16TS21		1 1 1	0/4096	0				
					1 1 0	0/1024	0				
					1 0 1	0/256					
					1 0 0	0/64					
				0 1 1	0/16						
		0 1 0	0/4								
		0 0 1	0/2								
				0 0 0	0/1						

III-3 PERIPHERAL BLOCK: PRESCALER

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 3 clock control register	004014A (B)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	P16TON3	16-bit timer 3 clock control	1 On	0 Off	0	R/W	
		D2	P16TS32	16-bit timer 3 clock division ratio selection	P16TS3[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P16TS31		1	1	1	0	θ/4096
		D0	P16TS30		1	1	0	0	θ/1024
					1	0	1	0	θ/256
					1	0	0	0	θ/64
					0	1	1	0	θ/16
					0	1	0	0	θ/4
					0	0	1	0	θ/2
			0	0	0	0	θ/1		
16-bit timer 4 clock control register	004014B (B)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	P16TON4	16-bit timer 4 clock control	1 On	0 Off	0	R/W	
		D2	P16TS42	16-bit timer 4 clock division ratio selection	P16TS4[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P16TS41		1	1	1	0	θ/4096
		D0	P16TS40		1	1	0	0	θ/1024
					1	0	1	0	θ/256
					1	0	0	0	θ/64
					0	1	1	0	θ/16
					0	1	0	0	θ/4
					0	0	1	0	θ/2
			0	0	0	0	θ/1		
16-bit timer 5 clock control register	004014C (B)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	P16TON5	16-bit timer 5 clock control	1 On	0 Off	0	R/W	
		D2	P16TS52	16-bit timer 5 clock division ratio selection	P16TS5[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P16TS51		1	1	1	0	θ/4096
		D0	P16TS50		1	1	0	0	θ/1024
					1	0	1	0	θ/256
					1	0	0	0	θ/64
					0	1	1	0	θ/16
					0	1	0	0	θ/4
					0	0	1	0	θ/2
			0	0	0	0	θ/1		
8-bit timer 0/1 clock control register	004014D (B)	D7	P8TON1	8-bit timer 1 clock control	1 On	0 Off	0	R/W	
		D6	P8TS12	8-bit timer 1 clock division ratio selection	P8TS1[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D5	P8TS11		1	1	1	0	θ/4096
		D4	P8TS10		1	1	0	0	θ/2048
					1	0	1	0	θ/1024
					1	0	0	0	θ/512
					0	1	1	0	θ/256
					0	1	0	0	θ/128
					0	0	1	0	θ/64
					0	0	0	0	θ/32
		D3	P8TON0	8-bit timer 0 clock control	1 On	0 Off	0	R/W	
		D2	P8TS02	8-bit timer 0 clock division ratio selection	P8TS0[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P8TS01		1	1	1	0	θ/256
		D0	P8TS00		1	1	0	0	θ/128
			1	0	1	0	θ/64		
			1	0	0	0	θ/32		
			0	1	1	0	θ/16		
			0	1	0	0	θ/8		
			0	0	1	0	θ/4		
			0	0	0	0	θ/2		
8-bit timer 2/3 clock control register	004014E (B)	D7	P8TON3	8-bit timer 3 clock control	1 On	0 Off	0	R/W	
		D6	P8TS32	8-bit timer 3 clock division ratio selection	P8TS3[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D5	P8TS31		1	1	1	0	θ/256
		D4	P8TS30		1	1	0	0	θ/128
					1	0	1	0	θ/64
					1	0	0	0	θ/32
					0	1	1	0	θ/16
					0	1	0	0	θ/8
					0	0	1	0	θ/4
					0	0	0	0	θ/2
		D3	P8TON2	8-bit timer 2 clock control	1 On	0 Off	0	R/W	
		D2	P8TS22	8-bit timer 2 clock division ratio selection	P8TS2[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P8TS21		1	1	1	0	θ/4096
		D0	P8TS20		1	1	0	0	θ/2048
			1	0	1	0	θ/64		
			1	0	0	0	θ/32		
			0	1	1	0	θ/16		
			0	1	0	0	θ/8		
			0	0	1	0	θ/4		
			0	0	0	0	θ/2		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
A/D clock control register	004014F (B)	D7-4	–	reserved	–	–	–	0 when being read.		
		D3	PSONAD	A/D converter clock control	1 On	0 Off	0	R/W		
		D2	PSAD2	A/D converter clock division ratio	PSAD[2:0]		Division ratio	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	PSAD1	selection	1	1	1	0/256	0	
		D0	PSAD0		1	1	0	0/128	0	
					1	0	1	0/64		
					1	0	0	0/32		
					0	1	1	0/16		
					0	1	0	0/8		
		0	0		1	0/4				
		0	0	0	0/2					
Power control register	0040180 (B)	D7	CLKDT1	System clock division ratio selection	CLKDT[1:0]		Division ratio	0	R/W	
		D6	CLKDT0		1	1	1/8	0		
					1	0	1/4			
					0	1	1/2			
				0	0	1/1				
		D5	PSCON	Prescaler On/Off control	1 On	0 Off	1	R/W		
		D4-3	–	reserved	–	–	0	–	Writing 1 not allowed.	
D2	CLKCHG	CPU operating clock switch	1 OSC3	0 OSC1	1	R/W				
D1	SOSC3	High-speed (OSC3) oscillation On/Off	1 On	0 Off	1	R/W				
D0	SOSC1	Low-speed (OSC1) oscillation On/Off	1 On	0 Off	1	R/W				
Prescaler clock select register	0040181 (B)	D7-1	–	reserved	–	–	0	–		
		D0	PSCDT0	Prescaler clock selection	1 OSC1	0 OSC3/PLL	0	R/W		
Power control protect register	004019E (B)	D7	CLGP7	Power control register protect flag	Writing 10010110 (0x96)		0	R/W		
		D6	CLGP6		removes the write protection of the power control register (0x40180) and the clock option register (0x40190).		0			
		D5	CLGP5		Writing another value set the write protection.		0			
		D4	CLGP4				0			
		D3	CLGP3				0			
		D2	CLGP2				0			
		D1	CLGP1				0			
		D0	CLGP0				0			

#### **PSCON:** Prescaler on/off control (D5) / Power control register (0x40180)

Turns the prescaler on or off.

Write "1": On

Write "0": Off

Read: Valid

The source clock is input to the prescaler by writing "1" to PSCON, thereby starting a dividing operation.

The prescaler is turned off by writing "0". If the peripheral circuits do not need to be operated, write "0" to this bit to reduce current consumption. Since PSCON is protected against writing the same as SOSC1, SOSC3, CLKCHG and CLKDT[1:0], CLGP[7:0] must be set to "0b10010110" before PSCON can be changed.

In addition, writing "0" (Off) to PSCON stops supplying the source clock to the prescaler and stops the peripheral circuits that use the same clock (e.g., 16-bit programmable timers, 8-bit programmable timers, A/D converter, serial interface, and ports). Therefore, do not turn off the prescaler when these peripheral circuits are used.

At initial reset, PSCON is set to "1" (On).

#### **CLGP7–CLGP0:** Power-control register protection flag ([D[7:0]]) / Power control protection register (0x4019E)

These bits remove the protection against writing to addresses 0x40180 and 0x40190.

Write "0b10010110": Write protection removed

Write other than the above: No operation (write-protected)

Read: Valid

Before writing to address 0x40180 or 0x40190, set CLGP[7:0] to "0b10010110" to remove the protection against writing to that address. This clearing of write protection is effective for only one writing, so the bits are cleared to "0b00000000" by one writing. Therefore, CLGP[7:0] must be set each time the protected address is written to.

At initial reset, CLGP is set to "0b00000000" (write-protected).

### III-3 PERIPHERAL BLOCK: PRESCALER

---

#### **PSCDT0:** Prescaler clock selection (D0) / Prescaler clock select register (0x40181)

---

Select the source clock for the prescaler.

Write "1": OSC1 clock

Write "0": OSC3 clock/PLL output clock

Read: Valid

When "1" is written to PSCDT0, the OSC1 clock (typ. 32 kHz) is selected.

When "0" is written, the OSC3 clock (when the PLL is not used) or the PLL output clock (when the PLL is used) is selected.

For the prescaler clock, the clock source same as the CPU operating clock must be selected.

At initial reset, PSCDT0 is set to "0" (OSC3 clock/PLL output clock).

---

**P16TS0[2:0]:** 16-bit timer 0 clock division ratio (D[2:0]) / 16-bit timer 0 clock control register (0x40147)

**P16TS1[2:0]:** 16-bit timer 1 clock division ratio (D[2:0]) / 16-bit timer 1 clock control register (0x40148)

**P16TS2[2:0]:** 16-bit timer 2 clock division ratio (D[2:0]) / 16-bit timer 2 clock control register (0x40149)

**P16TS3[2:0]:** 16-bit timer 3 clock division ratio (D[2:0]) / 16-bit timer 3 clock control register (0x4014A)

**P16TS4[2:0]:** 16-bit timer 4 clock division ratio (D[2:0]) / 16-bit timer 4 clock control register (0x4014B)

**P16TS5[2:0]:** 16-bit timer 5 clock division ratio (D[2:0]) / 16-bit timer 5 clock control register (0x4014C)

**P8TS0[2:0]:** 8-bit timer 0 clock division ratio (D[2:0]) / 8-bit timer 0/1 clock control register (0x4014D)

**P8TS1[2:0]:** 8-bit timer 1 clock division ratio (D[6:4]) / 8-bit timer 0/1 clock control register (0x4014D)

**P8TS2[2:0]:** 8-bit timer 2 clock division ratio (D[2:0]) / 8-bit timer 2/3 clock control register (0x4014E)

**P8TS3[2:0]:** 8-bit timer 3 clock division ratio (D[6:4]) / 8-bit timer 2/3 clock control register (0x4014E)

**P8TS4[2:0]:** 8-bit timer 4 clock division ratio (D[2:0]) / 8-bit timer 4/5 clock control register (0x40145)

**P8TS5[2:0]:** 8-bit timer 5 clock division ratio (D[6:4]) / 8-bit timer 4/5 clock control register (0x40145)

**PSAD[2:0]:** A/D converter clock division ratio (D[2:0]) / A/D clock control register (0x4014F)

---

Select a clock for each peripheral circuit.

The desired division ratio can be selected from among the eight ratios shown on the I/O map. Note that the division ratio differs for each peripheral circuit.

These bits can also be read out.

At initial reset, all of these bits are set to "0b000" (highest frequency available).

---

**P16TON0:** 16-bit timer 0 clock control (D3) / 16-bit timer 0 clock control register (0x40147)

**P16TON1:** 16-bit timer 1 clock control (D3) / 16-bit timer 1 clock control register (0x40148)

**P16TON2:** 16-bit timer 2 clock control (D3) / 16-bit timer 2 clock control register (0x40149)

**P16TON3:** 16-bit timer 3 clock control (D3) / 16-bit timer 3 clock control register (0x4014A)

**P16TON4:** 16-bit timer 4 clock control (D3) / 16-bit timer 4 clock control register (0x4014B)

**P16TON5:** 16-bit timer 5 clock control (D3) / 16-bit timer 5 clock control register (0x4014C)

**P8TON0:** 8-bit timer 0 clock control (D3) / 8-bit timer 0/1 clock control register (0x4014D)

**P8TON1:** 8-bit timer 1 clock control (D7) / 8-bit timer 0/1 clock control register (0x4014D)

**P8TON2:** 8-bit timer 2 clock control (D3) / 8-bit timer 2/3 clock control register (0x4014E)

**P8TON3:** 8-bit timer 3 clock control (D7) / 8-bit timer 2/3 clock control register (0x4014E)

**P8TON4:** 8-bit timer 4 clock control (D3) / 8-bit timer 4/5 clock control register (0x40145)

**P8TON5:** 8-bit timer 5 clock control (D7) / 8-bit timer 4/5 clock control register (0x40145)

**PSONAD:** A/D converter clock control (D3) / A/D clock control register (0x4014F)

---

Control the clock supply to each peripheral circuit.

Write "1": On

Write "0": Off

Read: Valid

The clock selected using the division ratio setup bits is output to the corresponding peripheral circuit by writing "1" to these bits.

The clock is not output by writing "0". If the peripheral circuits do not need to be operated, write "0" to these bits.

This helps to reduce current consumption.

At initial reset, all of these bits are set to "0" (Off).

---

**P8TPCK0:** 8-bit timer 0 clock selection (D0) / 8-bit timer clock select register (0x40146)  
**P8TPCK1:** 8-bit timer 1 clock selection (D1) / 8-bit timer clock select register (0x40146)  
**P8TPCK2:** 8-bit timer 2 clock selection (D2) / 8-bit timer clock select register (0x40146)  
**P8TPCK3:** 8-bit timer 3 clock selection (D3) / 8-bit timer clock select register (0x40146)  
**P8TPCK4:** 8-bit timer 4 clock selection (D0) / 8-bit timer 4/5 clock select register (0x40140)  
**P8TPCK5:** 8-bit timer 5 clock selection (D1) / 8-bit timer 4/5 clock select register (0x40140)

---

Select the operating clock for the 8-bit programmable timer.

Write "1": Prescaler input clock ( $\theta/1$ )

Write "0": Divided clock

Read: Valid

When "1" is written to P8TPCKx, the prescaler input clock ( $\theta/1$ ) is selected for the 8-bit timer x operating clock.

The clock output is controlled by the P8TONx bit even if P8TPCKx is set to "1".

When "0" is written, the divided clock that is selected by P8TSx[2:0] will be output to the 8-bit timer x.

At initial reset, P8TPCKx is set to "0" (divided clock).

## Programming Notes

---

- (1) For the prescaler clock, the clock source same as the CPU operating clock must be selected.
- (2) In the following cases, the prescaler output clock may contain a hazard:
  - If, during outputting of a clock, its division ratio is changed
  - When the clock output is switched between on and off
  - When the oscillation circuit is turned off or the CPU operating clock is switched over
 Before performing these operations, make sure the 16-bit and 8-bit programmable timers and the A/D converter are turned off.
- (3) When the 16-bit and 8-bit programmable timers and the A/D converter do not need to be operated, turn off the clock supply to those peripheral circuits. This helps to reduce current consumption.
- (4) Be aware that some peripheral circuits stops operating when the prescaler is turned off (PSCON (D5) / Power control register (0x40180) = "0") as well as the peripheral circuits that use the prescaler output clock. The prescaler status affects the peripheral circuits shown below.
  - (A) Peripheral circuits that use the clock generated by the prescaler
    - 16-bit programmable timers (watchdog timer)
    - 8-bit programmable timers (serial interface)
    - A/D converter
  - (B) Peripheral circuits that use the clock supplied to the prescaler (the source clock for prescaler)
    - 16-bit programmable timers (watchdog timer)
    - 8-bit programmable timers
    - A/D converter
    - Serial interface
    - Input/output ports

If none of all circuits of the above (A) and (B) need to be used, turn off the prescaler (PSCON = "0"). If a circuit of the above (A) or (B) need to be used, do not turn off the prescaler. When the prescaler is turned off, the clock supply to the circuits of the above (B) stops. When some these circuits of the above (A) need to be used, turn off all other unnecessary circuits and stop the clock supply from the prescaler to those circuits.

THIS PAGE IS BLANK.



## III-4 8-BIT PROGRAMMABLE TIMERS

### *Configuration of 8-Bit Programmable Timer*

The Peripheral Block contains six channels of 8-bit programmable timers (timers 0 to 5).

Figure III.4.1 shows the structure of the 8-bit programmable timer.

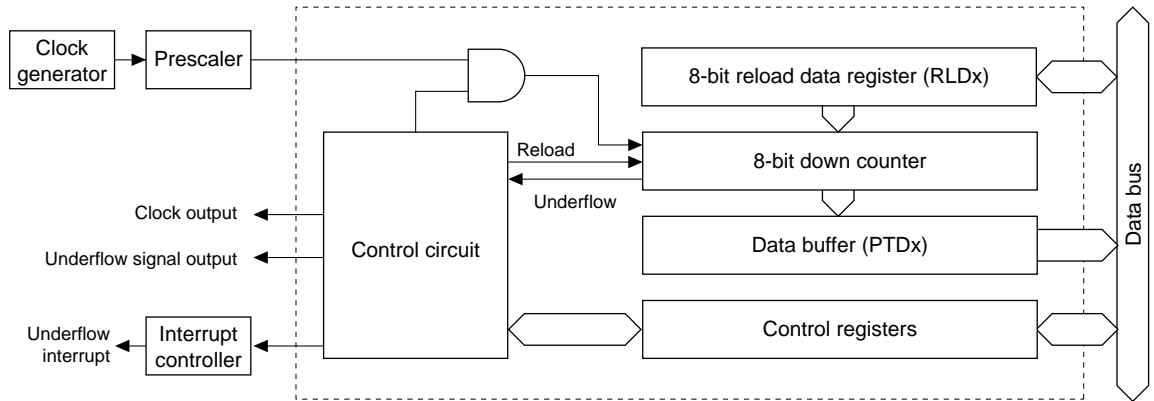


Figure III.4.1 Structure of 8-Bit Programmable Timer

Each timer consists of an 8-bit programmable counter and can output a clock generated by the counter's underflow signal to the internal peripheral circuits or external devices. The output clock cycle can be selected from a wide range of cycles by setting the preset data that can be set in the software and the input clock in the prescaler.

## Output Pins of 8-Bit Programmable Timers

The underflow signals of 8-bit programmable timers 0 to 3 can be output to external devices.

Table III.4.1 shows the pins that are used to output the underflow signals of the 8-bit programmable timers to external devices.

Table III.4.1 Output Pins of 8-Bit Programmable Timers

Pin name	I/O	Function	Function select bit
DST0(P10/EXCL0/T8UF0)	I/O	DST0 output / I/O port / 16-bit timer 0 event counter input / 8-bit timer 0 output	CFP10(D0)/P1 function select register (0x402D4) CFEX1(D1)/Port function extension register (0x402DF)
DST1(P11/EXCL1/T8UF1)	I/O	DST1 output / I/O port / 16-bit timer 1 event counter input / 8-bit timer 1 output	CFP11(D1)/P1 function select register (0x402D4) CFEX1(D1)/Port function extension register (0x402DF)
DST2(P12/EXCL2/T8UF2)	I/O	DST2 output / I/O port / 16-bit timer 2 event counter input / 8-bit timer 2 output	CFP12(D2)/P1 function select register (0x402D4) CFEX0(D0)/Port function extension register (0x402DF)
DPCO(P13/EXCL3/T8UF3)	I/O	DPCO output / I/O port / 16-bit timer 3 event counter input / 8-bit timer 3 output	CFP13(D3)/P1 function select register (0x402D4) CFEX1(D1)/Port function extension register (0x402DF)

### T8UFx (output pin of the 8-bit programmable timer)

This pin outputs a clock divided in each 8-bit programmable timer. The pulse width is equal to that of input clock of the 8-bit programmable timer (prescaler output). Therefore, the pulse width varies according to the prescaler setting.

### How to set the output pins of the 8-bit programmable timer

All pins used by the 8-bit programmable timers are shared with I/O ports, event counter inputs of the 16-bit programmable timers and debug signal outputs. At cold start, all these pins are set for the debug signal outputs (function select bit CFP1[3:0] = "0", port extended function bit CFEX[1:0] = "1"). When using the clock output function of the 8-bit programmable timer, write "0" to the port extended function bit CFEXx and write "1" to the function select bit CFP1x for the corresponding pin.

Then, after setting the above, write "1" to the I/O port's I/O control bit IOC1x (D[3:0]) / P1 I/O control register (0x402D6) to set to output mode. In input mode, the pin functions as the 16-bit programmable timer's event counter input and cannot be used to output a clock of the 8-bit programmable timer. At cold start, the register is set to input mode. At hot start, the register retains its status from prior to the reset.

## Uses of 8-Bit Programmable Timers

The down-counter of the 8-bit programmable timer cyclically outputs an underflow signal according to the preset data that is set in the software. This underflow signal is used to generate an interrupt request to the CPU or to control the internal peripheral circuits. In addition, this signal can be output to external devices.

Furthermore, each 8-bit programmable timer generates a clock from the underflow signal by dividing it by 2, and the resulting clock is output to a specific internal peripheral circuit.

### CPU interrupt request/IDMA invocation request

Each timer's underflow condition can be used as an interrupt factor to output an interrupt request to the CPU.

Therefore, an interrupt can be generated at an interval that is set in the software.

This interrupt factor also can be used to invoke IDMA or HSDMA.

### Clock output to external devices

The underflow signal can be output from the chip to the outside. This output can be used to control external devices. The output pins of each timer are described in the preceding section.

### Control of and clock supply to internal peripheral circuits

The following describes the functions controlled by the underflow signal from the 8-bit programmable timer and the internal peripheral circuits that use the timer's output clock.

#### 8-bit programmable timer 0

##### • A/D conversion start trigger

The A/D converter enables a trigger for starting the A/D conversion to be selected from among four available types. One of these is the underflow signal of the 8-bit programmable timer 0. This makes it possible to perform the A/D conversion at programmable intervals.

To use this function, write "10" to the A/D converter control bit TS[1:0] (D[4:3]) / A/D trigger register (0x40242) to select the 8-bit programmable timer 0 as the trigger.

#### 8-bit programmable timer 1

##### • Oscillation stabilization wait time of the high-speed (OSC3) oscillation circuit

When SLEEP mode is cleared by an external interrupt, the high-speed (OSC3) oscillation circuit starts oscillating. To prevent the CPU from being operated erratically by an unstable clock before the oscillation stabilizes, the C33 Core Block enables setting of the waiting time before the CPU starts operating after SLEEP is cleared. Use the 8-bit programmable timer 1 to generate this waiting time. If the 8-bit programmable timer 1 is set so that the timer is actuated when the high-speed (OSC3) oscillation circuit starts oscillating the timer and, after the oscillation stabilization time elapses, an underflow signal is generated, then the CPU can be started up by that underflow signal.

To use this function, write "0" to the oscillation circuit control bit 8T1ON (D2) / Clock option register (0x40190) to enable the oscillation stabilization waiting function.

#### 8-bit programmable timer 2

##### • Clock supply to the Ch.0 serial interface

When using the Ch.0 serial interface in the clock-synchronized master mode or the internal clock-based asynchronous mode, the output clock derived from the underflow signal of the 8-bit programmable timer 2 by dividing it by 2 is supplied to the serial interface as its operating clock. This enables the transfer rate of the serial interface to be programmed.

To use this function, write "0" to the serial interface control bit SSCK0 (D2) / Serial I/F Ch.0 control register (0x401E3) to select the internal clock.

### 8-bit programmable timer 3

- **Clock supply to the Ch.1 serial interface**

When using the Ch.1 serial interface in the clock-synchronized master mode or the internal clock-based asynchronous mode, the output clock derived from the underflow signal of the 8-bit programmable timer 3 by dividing it by 2 is supplied to the serial interface as its operating clock. This enables the transfer rate of the serial interface to be programmed.

To use this function, write "0" to the serial interface control bit SSCK1 (D2) / Serial I/F Ch.1 control register (0x401E8) to select the internal clock.

### 8-bit programmable timer 4

- **Clock supply to the Ch.2 serial interface**

When using the Ch.2 serial interface in the clock-synchronized master mode or the internal clock-based asynchronous mode, the output clock derived from the underflow signal of the 8-bit programmable timer 4 by dividing it by 2 is supplied to the serial interface as its operating clock. This enables the transfer rate of the serial interface to be programmed.

To use this function, write "0" to the serial interface control bit SSCK2 (D2) / Serial I/F Ch.2 control register (0x401F3) to select the internal clock.

### 8-bit programmable timer 5

- **Clock supply to the Ch.3 serial interface**

When using the Ch.3 serial interface in the clock-synchronized master mode or the internal clock-based asynchronous mode, the output clock derived from the underflow signal of the 8-bit programmable timer 5 by dividing it by 2 is supplied to the serial interface as its operating clock. This enables the transfer rate of the serial interface to be programmed.

To use this function, write "0" to the serial interface control bit SSCK3 (D2) / Serial I/F Ch.3 control register (0x401F8) to select the internal clock.

## Control and Operation of 8-Bit Programmable Timer

With the 8-bit programmable timer, the following settings must first be made before it starts counting:

1. Setting the output pin (only when necessary)
2. Setting the input clock
3. Setting the preset data (initial counter value)
4. Setting the interrupt/IDMA/HSDMA

Setting of an output pin is necessary only when the output clock of the 8-bit programmable timer is supplied to external devices. For details on how to set the pin, refer to "Output Pins of 8-Bit Programmable Timers".

For details on how to set interrupts and DMA, refer to "8-Bit Programmable Timer Interrupts and DMA".

**Note:** The 8-bit programmable timers 0 through 5 all operate in the same way during counting, and the structure of their control registers is also the same. The control bit names are assigned the numerals "0" through "5" to denote the timer numbers. Since all these timers have common functions, timer numbers here are represented by "x" unless necessary to specify a timer number.

### Setting the input clock

The 8-bit programmable timer is operated by the prescaler's output clock. The prescaler's division ratio can be selected for each timer.

	Division ratio select bit	Clock control bit	Register
8-bit timer 0:	P8TS0[2:0] (D[2:0])	P8TON0 (D3)	8-bit timer 0/1 clock control register (0x4014D)
8-bit timer 1:	P8TS1[2:0] (D[6:4])	P8TON1 (D7)	8-bit timer 0/1 clock control register (0x4014D)
8-bit timer 2:	P8TS2[2:0] (D[2:0])	P8TON2 (D3)	8-bit timer 2/3 clock control register (0x4014E)
8-bit timer 3:	P8TS3[2:0] (D[6:4])	P8TON3 (D7)	8-bit timer 2/3 clock control register (0x4014E)
8-bit timer 4:	P8TS4[2:0] (D[2:0])	P8TON4 (D3)	8-bit timer 4/5 clock control register (0x40145)
8-bit timer 5:	P8TS5[2:0] (D[6:4])	P8TON5 (D7)	8-bit timer 4/5 clock control register (0x40145)

Note that the division ratios differ for each timer (see Table III.4.2).

Furthermore, the prescaler input clock can be directly supplied to the 8-bit timer by writing "1" to the P8TPCKx bit in the 8-bit timer clock select register (0x40146).

Timer 0 clock selection: P8TPCK0 (D0) / 8-bit timer clock select register (0x40146)

Timer 1 clock selection: P8TPCK1 (D1) / 8-bit timer clock select register (0x40146)

Timer 2 clock selection: P8TPCK2 (D2) / 8-bit timer clock select register (0x40146)

Timer 3 clock selection: P8TPCK3 (D3) / 8-bit timer clock select register (0x40146)

Timer 4 clock selection: P8TPCK4 (D0) / 8-bit timer clock select register (0x40140)

Timer 5 clock selection: P8TPCK5 (D1) / 8-bit timer clock select register (0x40140)

When using the divided clock selected by P8TSx, set P8TPCKx to "0".

Table III.4.2 Input Clock Selection

Timer	P8TSx = 7	P8TSx = 6	P8TSx = 5	P8TSx = 4	P8TSx = 3	P8TSx = 2	P8TSx = 1	P8TSx = 0	P8TPCK = 1
Timer 0	fPSCIN/256	fPSCIN/128	fPSCIN/64	fPSCIN/32	fPSCIN/16	fPSCIN/8	fPSCIN/4	fPSCIN/2	fPSCIN/1
Timer 1	fPSCIN/4096	fPSCIN/2048	fPSCIN/1024	fPSCIN/512	fPSCIN/256	fPSCIN/128	fPSCIN/64	fPSCIN/32	fPSCIN/1
Timer 2	fPSCIN/4096	fPSCIN/2048	fPSCIN/64	fPSCIN/32	fPSCIN/16	fPSCIN/8	fPSCIN/4	fPSCIN/2	fPSCIN/1
Timer 3	fPSCIN/256	fPSCIN/128	fPSCIN/64	fPSCIN/32	fPSCIN/16	fPSCIN/8	fPSCIN/4	fPSCIN/2	fPSCIN/1
Timer 4	fPSCIN/4096	fPSCIN/2048	fPSCIN/64	fPSCIN/32	fPSCIN/16	fPSCIN/8	fPSCIN/4	fPSCIN/2	fPSCIN/1
Timer 5	fPSCIN/256	fPSCIN/128	fPSCIN/64	fPSCIN/32	fPSCIN/16	fPSCIN/8	fPSCIN/4	fPSCIN/2	fPSCIN/1

fPSCIN: Prescaler input clock frequency

The selected clock is output from the prescaler to the 8-bit programmable timer by writing "1" to P8TONx.

- Notes:**
- The 8-bit programmable timer operates only when the prescaler is operating. (Refer to "Prescaler".)
  - Do not use a clock that is faster than the CPU operating clock as the 8-bit programmable timer.
  - When setting an input clock, make sure the 8-bit programmable timer is turned off.

### Setting preset data (initial counter value)

Each timer has an 8-bit down-counter and a reload data register. The reload data register RLD<sub>x</sub> is used to set the initial value of the down-counter of each timer.

Timer 0 reload data: RLD0[7:0] (D[7:0]) / 8-bit timer 0 reload data register (0x40161)

Timer 1 reload data: RLD1[7:0] (D[7:0]) / 8-bit timer 1 reload data register (0x40165)

Timer 2 reload data: RLD2[7:0] (D[7:0]) / 8-bit timer 2 reload data register (0x40169)

Timer 3 reload data: RLD3[7:0] (D[7:0]) / 8-bit timer 3 reload data register (0x4016D)

Timer 4 reload data: RLD4[7:0] (D[7:0]) / 8-bit timer 4 reload data register (0x40175)

Timer 5 reload data: RLD5[7:0] (D[7:0]) / 8-bit timer 5 reload data register (0x40179)

The reload data registers can be read and written. At initial reset, the reload data registers are not initialized.

The data written to this register is preset in the down-counter, and the counter starts counting down from the preset value.

Data is thus preset in the down-counter in the following two cases:

#### 1. When it is preset in the software

Presetting in the software is performed using the preset control bit PSET<sub>x</sub>. When this bit is set to "1", the content of the reload data register is loaded into the down-counter at that point.

Timer 0 preset: PSET0 (D1) / 8-bit timer 0 control register (0x40160)

Timer 1 preset: PSET1 (D1) / 8-bit timer 1 control register (0x40164)

Timer 2 preset: PSET2 (D1) / 8-bit timer 2 control register (0x40168)

Timer 3 preset: PSET3 (D1) / 8-bit timer 3 control register (0x4016C)

Timer 4 preset: PSET4 (D1) / 8-bit timer 4 control register (0x40174)

Timer 5 preset: PSET5 (D1) / 8-bit timer 5 control register (0x40178)

#### 2. When the down-counter underflows during counting

Since the reload data is preset in the down-counter upon underflow, its underflow cycle is determined by the value that is set in the reload data register. This underflow signal controls each function described in the preceding section.

Before starting the 8-bit programmable timer, set the initial value in the reload data register and use the PSET<sub>x</sub> bit to preset the data in the down-counter.

The underflow cycle is determined by the prescaler setting and the reload data. The relationship between these two parameters is expressed by the following equation:

$$\text{Under flow cycle} = \frac{\text{RLD}_x + 1}{f_{\text{PSCIN}} \times \text{pdr}} \quad [\text{sec.}]$$

$f_{\text{PSCIN}}$ : Prescaler input clock frequency [Hz]

pdr: Prescaler division ratio set by P8TS<sub>x</sub>

RLD<sub>x</sub>: Set value of the RLD<sub>x</sub> register (0 to 255)

### Timer RUN/STOP control

Each timer has a PTRUN<sub>x</sub> bit to control RUN/STOP.

Timer 0 RUN/STOP control: PTRUN0 (D0) / 8-bit timer 0 control register (0x40160)

Timer 1 RUN/STOP control: PTRUN1 (D0) / 8-bit timer 1 control register (0x40164)

Timer 2 RUN/STOP control: PTRUN2 (D0) / 8-bit timer 2 control register (0x40168)

Timer 3 RUN/STOP control: PTRUN3 (D0) / 8-bit timer 3 control register (0x4016C)

Timer 4 RUN/STOP control: PTRUN4 (D0) / 8-bit timer 4 control register (0x40174)

Timer 5 RUN/STOP control: PTRUN5 (D0) / 8-bit timer 5 control register (0x40178)

The timer is initiated to start counting down by writing "1" to PTRUN<sub>x</sub>. Writing "0" to PTRUN<sub>x</sub> disables the clock input and causes the timer to stop counting.

This RUN/STOP control does not affect the counter data. Even when the timer has stopped counting, the counter retains its count so that it can start counting again from that point.

When the terminal count is reached and the counter underflows, the initial value is reloaded from the reload data register into the counter.

When both the timer RUN/STOP control bit (PTRUN<sub>x</sub>) and the timer preset bit (PSET<sub>x</sub>) are set to "1" at the same time, the timer starts counting after presetting the reload register value into the counter.

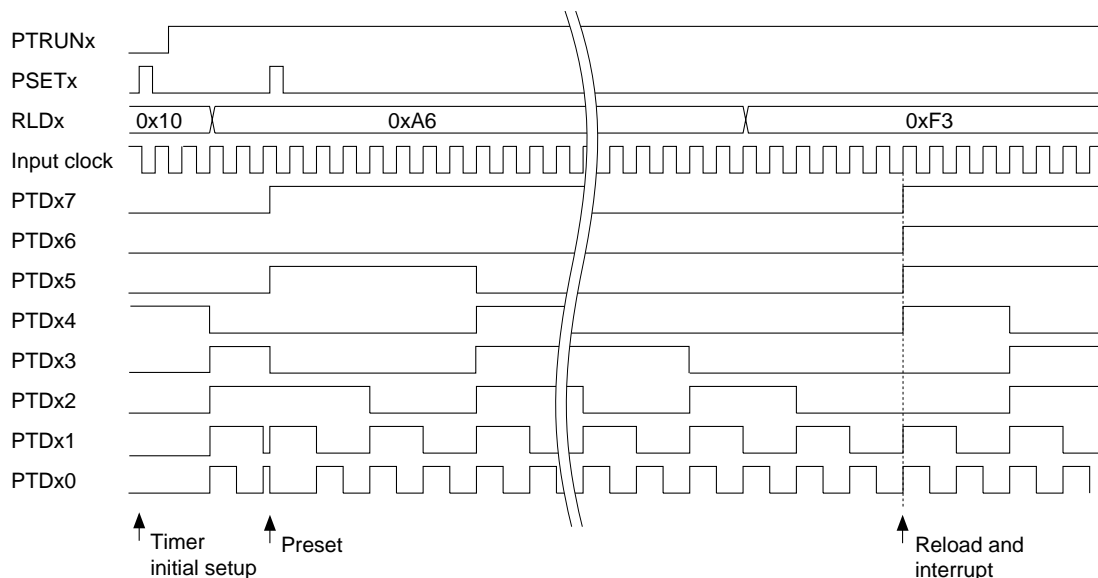


Figure III.4.2 Basic Operation Timing of Counter

### Reading out counter data

The counter data is read out via a PTDX data buffer. The counter data can be read out at any time.

- Timer 0 data: PTD0[7:0] (D[7:0]) / 8-bit timer 0 counter data register (0x40162)
- Timer 1 data: PTD1[7:0] (D[7:0]) / 8-bit timer 1 counter data register (0x40166)
- Timer 2 data: PTD2[7:0] (D[7:0]) / 8-bit timer 2 counter data register (0x4016A)
- Timer 3 data: PTD3[7:0] (D[7:0]) / 8-bit timer 3 counter data register (0x4016E)
- Timer 4 data: PTD4[7:0] (D[7:0]) / 8-bit timer 4 counter data register (0x40176)
- Timer 5 data: PTD5[7:0] (D[7:0]) / 8-bit timer 5 counter data register (0x4017A)

## Control of Clock Output

When outputting an underflow signal of the 8-bit programmable timer to external devices, or when supplying a clock generated by the underflow signal to the serial interface, it is necessary to control the clock output of the timer.

Timer 0 clock output control: PTOUT0 (D2) / 8-bit timer 0 control register (0x40160)

Timer 1 clock output control: PTOUT1 (D2) / 8-bit timer 1 control register (0x40164)

Timer 2 clock output control: PTOUT2 (D2) / 8-bit timer 2 control register (0x40168)

Timer 3 clock output control: PTOUT3 (D2) / 8-bit timer 3 control register (0x4016C)

Timer 4 clock output control: PTOUT4 (D2) / 8-bit timer 4 control register (0x40174) \* for serial interface only

Timer 5 clock output control: PTOUT5 (D2) / 8-bit timer 5 control register (0x40178) \* for serial interface only

To output the underflow signal/clock, write "1" to PTOUTx. If an output pin has been set, the underflow signal is output from that pin.

The same applies when one of timers 2 to 5 has been set as the clock source of the serial interface. A clock generated from the underflow signal by dividing it by 2 is output to the serial interface through this control. The clock output is turned off by writing "0" to PTOUTx, and the external output is fixed at "0" and the internal clock output is fixed at "1".

Figure III.4.3 shows the waveforms of the output signals.

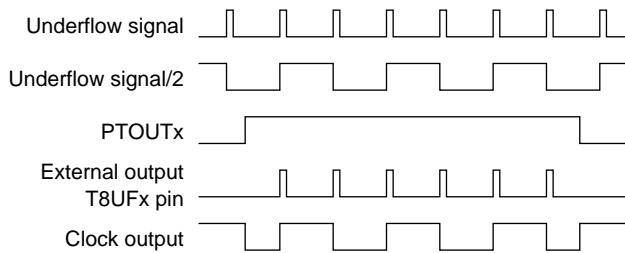


Figure III.4.3 8-Bit Programmable Timer Output Waveform

The underflow signal's pulse width (duration of the high period) is equal to that of the timer's input clock (prescaler's output).

### 8-bit timer external output (P10–P13 ports)

- 1) After an initial reset (cold start), the ports (P10–P13) are set to debug signal output ports.
- 2) The port (P10–P13) outputs "0" when it is set to the 8-bit timer output (timer output is off status).
- 3) The timer output is left as "0" when the timer output is turned on after setting the input clock and timer initial value.
- 4) When an underflow occurs after starting the timer, the port outputs a pulse with the same width as the 8-bit timer input clock pulse (prescaler's output).



## 8-Bit Programmable Timer Interrupts and DMA

The 8-bit programmable timer has a function to generate an interrupt based on the underflow state of the timer 0 to 5. The timing at which an interrupt is generated is shown in Figure III.4.2 in the preceding section.

### Control registers of the interrupt controller

Table III.4.3 shows the interrupt controller's control register provided for each timer.

Table III.4.3 Control Registers of Interrupt Controller

Timer	Interrupt factor flag	Interrupt enable register	Interrupt priority register
Timer 0	F8TU0(D0/0x40285)	E8TU0(D0/0x40275)	P8TM[2:0](D[2:0]/0x40269)
Timer 1	F8TU1(D1/0x40285)	E8TU1(D1/0x40275)	
Timer 2	F8TU2(D2/0x40285)	E8TU2(D2/0x40275)	
Timer 3	F8TU3(D3/0x40285)	E8TU3(D3/0x40275)	
Timer 4	F8TU4(D0/0x40288)	E8TU4(D0/0x40278)	
Timer 5	F8TU5(D1/0x40288)	E8TU5(D1/0x40278)	

When the timer underflows, the corresponding interrupt factor flag is set to "1". If the interrupt enable register bit corresponding to that interrupt factor flag has been set to "1", an interrupt request is generated.

An interrupt caused by a timer can be disabled by leaving the interrupt enable register bit for that timer set to "0". The interrupt factor flag is set to "1" whenever the timer underflows, regardless of how the interrupt enable register is set (even when it is set to "0").

The interrupt priority register sets an interrupt priority level (0 to 7) for the six timers as one interrupt source.

Within 8-bit programmable timers, timer 0 has the highest priority and timer 5 the lowest. An interrupt request to the CPU is accepted on the condition that no other interrupt request of a higher priority has been generated.

It is only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the timer interrupt level set by the interrupt priority register, that a timer interrupt request is actually accepted by the CPU. For details on these interrupt control registers and device operation when an interrupt has occurred, refer to "ITC (Interrupt Controller)".

III

### Intelligent DMA

The underflow interrupt factor of the timer 0 to 5 can invoke intelligent DMA (IDMA). This enables memory-to-memory DMA transfers to be performed cyclically.

The following shows the IDMA channel numbers set to each timer:

IDMA channel

Timer 0:	0x13
Timer 1:	0x14
Timer 2:	0x15
Timer 3:	0x16
Timer 4:	0x20
Timer 5:	0x21

For IDMA to be invoked, the IDMA request and IDMA enable bits shown in Table III.4.4 must be set to "1" in advance. Transfer conditions, etc. must also be set on the IDMA side in advance.

Table III.4.4 Control Bits for IDMA Transfer

Timer	IDMA request bit	IDMA enable bit
Timer 0	R8TU0(D2/0x40292)	DE8TU0(D2/0x40296)
Timer 1	R8TU1(D3/0x40292)	DE8TU1(D3/0x40296)
Timer 2	R8TU2(D4/0x40292)	DE8TU2(D4/0x40296)
Timer 3	R8TU3(D5/0x40292)	DE8TU3(D5/0x40296)
Timer 4	R8TU4(D0/0x4029B)	DE8TU4(D0/0x4029C)
Timer 5	R8TU5(D1/0x4029B)	DE8TU5(D1/0x4029C)

If the IDMA request and enable bits are set to "1", IDMA is invoked through generation of an interrupt factor. No interrupt request is generated at that point. An interrupt request is generated after the DMA transfer is completed. The registers can also be set so as not to generate an interrupt, with only a DMA transfer performed. For details on IDMA transfers and interrupt control upon completion of IDMA transfer, refer to "IDMA (Intelligent DMA)".

STM

## High-speed DMA

The underflow interrupt factor of the timer 0 to 3 can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit corresponding to the timer 0 to 3:

Table III.4.5 HSDMA Trigger Set-up Bits

Timer	HSDMA channel	Trigger set-up bits
Timer 0	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0/1 trigger set-up register (0x40298)
Timer 1	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298)
Timer 2	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299)
Timer 3	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299)

For HSDMA to be invoked, the trigger set-up bits should be set to "0101" in advance. Transfer conditions, etc. must also be set on the HSDMA side.

If the 8-bit timer is selected as the HSDMA trigger, the HSDMA channel is invoked through generation of the interrupt factor.

For details on HSDMA transfer, refer to "HSDMA (High-Speed DMA)".

## Trap vectors

The trap vector addresses for individual underflow interrupt factors are set by default as shown below:

- Timer 0 underflow interrupt: 0x0C000D0
- Timer 1 underflow interrupt: 0x0C000D4
- Timer 2 underflow interrupt: 0x0C000D8
- Timer 3 underflow interrupt: 0x0C000DC
- Timer 4 underflow interrupt: 0x0C00120
- Timer 5 underflow interrupt: 0x0C00124

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

## I/O Memory of 8-Bit Programmable Timers

Table III.4.6 shows the control bits of the 8-bit programmable timers.

For details on the I/O memory of the prescaler used to set a clock, refer to "Prescaler".

Table III.4.6 Control Bits of 8-Bit Programmable Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit timer 0 control register	0040160 (B)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	PTOUT0	8-bit timer 0 clock output control	1 On 0 Off	0	R/W	
		D1	PSET0	8-bit timer 0 preset	1 Preset 0 Invalid	–	W	0 when being read.
		D0	PTRUN0	8-bit timer 0 Run/Stop control	1 Run 0 Stop	0	R/W	
8-bit timer 0 reload data register	0040161 (B)	D7	RLD07	8-bit timer 0 reload data	0 to 255	X	R/W	
		D6	RLD06	RLD07 = MSB		X		
		D5	RLD05	RLD00 = LSB		X		
		D4	RLD04			X		
		D3	RLD03			X		
		D2	RLD02			X		
		D1	RLD01			X		
		D0	RLD00			X		
8-bit timer 0 counter data register	0040162 (B)	D7	PTD07	8-bit timer 0 counter data	0 to 255	X	R	
		D6	PTD06	PTD07 = MSB		X		
		D5	PTD05	PTD00 = LSB		X		
		D4	PTD04			X		
		D3	PTD03			X		
		D2	PTD02			X		
		D1	PTD01			X		
		D0	PTD00			X		
8-bit timer 1 control register	0040164 (B)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	PTOUT1	8-bit timer 1 clock output control	1 On 0 Off	0	R/W	
		D1	PSET1	8-bit timer 1 preset	1 Preset 0 Invalid	–	W	0 when being read.
		D0	PTRUN1	8-bit timer 1 Run/Stop control	1 Run 0 Stop	0	R/W	
8-bit timer 1 reload data register	0040165 (B)	D7	RLD17	8-bit timer 1 reload data	0 to 255	X	R/W	
		D6	RLD16	RLD17 = MSB		X		
		D5	RLD15	RLD10 = LSB		X		
		D4	RLD14			X		
		D3	RLD13			X		
		D2	RLD12			X		
		D1	RLD11			X		
		D0	RLD10			X		
8-bit timer 1 counter data register	0040166 (B)	D7	PTD17	8-bit timer 1 counter data	0 to 255	X	R	
		D6	PTD16	PTD17 = MSB		X		
		D5	PTD15	PTD10 = LSB		X		
		D4	PTD14			X		
		D3	PTD13			X		
		D2	PTD12			X		
		D1	PTD11			X		
		D0	PTD10			X		
8-bit timer 2 control register	0040168 (B)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	PTOUT2	8-bit timer 2 clock output control	1 On 0 Off	0	R/W	
		D1	PSET2	8-bit timer 2 preset	1 Preset 0 Invalid	–	W	0 when being read.
		D0	PTRUN2	8-bit timer 2 Run/Stop control	1 Run 0 Stop	0	R/W	
8-bit timer 2 reload data register	0040169 (B)	D7	RLD27	8-bit timer 2 reload data	0 to 255	X	R/W	
		D6	RLD26	RLD27 = MSB		X		
		D5	RLD25	RLD20 = LSB		X		
		D4	RLD24			X		
		D3	RLD23			X		
		D2	RLD22			X		
		D1	RLD21			X		
		D0	RLD20			X		
8-bit timer 2 counter data register	004016A (B)	D7	PTD27	8-bit timer 2 counter data	0 to 255	X	R	
		D6	PTD26	PTD27 = MSB		X		
		D5	PTD25	PTD20 = LSB		X		
		D4	PTD24			X		
		D3	PTD23			X		
		D2	PTD22			X		
		D1	PTD21			X		
		D0	PTD20			X		

### III-4 PERIPHERAL BLOCK: 8-BIT PROGRAMMABLE TIMERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
8-bit timer 3 control register	004016C (B)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	PTOUT3	8-bit timer 3 clock output control	1 On 0 Off	0	R/W		
		D1	PSET3	8-bit timer 3 preset	1 Preset 0 Invalid	–	W	0 when being read.	
		D0	PTRUN3	8-bit timer 3 Run/Stop control	1 Run 0 Stop	0	R/W		
8-bit timer 3 reload data register	004016D (B)	D7	RLD37	8-bit timer 3 reload data RLD37 = MSB RLD30 = LSB	0 to 255		X	R/W	
		D6	RLD36		X				
		D5	RLD35		X				
		D4	RLD34		X				
		D3	RLD33		X				
		D2	RLD32		X				
		D1	RLD31		X				
		D0	RLD30		X				
8-bit timer 3 counter data register	004016E (B)	D7	PTD37	8-bit timer 3 counter data PTD37 = MSB PTD30 = LSB	0 to 255		X	R	
		D6	PTD36		X				
		D5	PTD35		X				
		D4	PTD34		X				
		D3	PTD33		X				
		D2	PTD32		X				
		D1	PTD31		X				
		D0	PTD30		X				
8-bit timer 4 control register	0040174 (B)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	PTOUT4	8-bit timer 4 clock output control	1 On 0 Off	0	R/W		
		D1	PSET4	8-bit timer 4 preset	1 Preset 0 Invalid	–	W	0 when being read.	
		D0	PTRUN4	8-bit timer 4 Run/Stop control	1 Run 0 Stop	0	R/W		
8-bit timer 4 reload data register	0040175 (B)	D7	RLD47	8-bit timer 4 reload data RLD47 = MSB RLD40 = LSB	0 to 255		X	R/W	
		D6	RLD46		X				
		D5	RLD45		X				
		D4	RLD44		X				
		D3	RLD43		X				
		D2	RLD42		X				
		D1	RLD41		X				
		D0	RLD40		X				
8-bit timer 4 counter data register	0040176 (B)	D7	PTD47	8-bit timer 4 counter data PTD47 = MSB PTD40 = LSB	0 to 255		X	R	
		D6	PTD46		X				
		D5	PTD45		X				
		D4	PTD44		X				
		D3	PTD43		X				
		D2	PTD42		X				
		D1	PTD41		X				
		D0	PTD40		X				
8-bit timer 5 control register	0040178 (B)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	PTOUT5	8-bit timer 5 clock output control	1 On 0 Off	0	R/W		
		D1	PSET5	8-bit timer 5 preset	1 Preset 0 Invalid	–	W	0 when being read.	
		D0	PTRUN5	8-bit timer 5 Run/Stop control	1 Run 0 Stop	0	R/W		
8-bit timer 5 reload data register	0040179 (B)	D7	RLD57	8-bit timer 5 reload data RLD57 = MSB RLD50 = LSB	0 to 255		X	R/W	
		D6	RLD56		X				
		D5	RLD55		X				
		D4	RLD54		X				
		D3	RLD53		X				
		D2	RLD52		X				
		D1	RLD51		X				
		D0	RLD50		X				
8-bit timer 5 counter data register	004017A (B)	D7	PTD57	8-bit timer 5 counter data PTD57 = MSB PTD50 = LSB	0 to 255		X	R	
		D6	PTD56		X				
		D5	PTD55		X				
		D4	PTD54		X				
		D3	PTD53		X				
		D2	PTD52		X				
		D1	PTD51		X				
		D0	PTD50		X				
8-bit timer, serial I/F Ch.0 interrupt priority register	0040269 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PSIO02	Serial interface Ch.0 interrupt level	0 to 7		X	R/W	
		D5	PSIO01		X				
		D4	PSIO00		X				
		D3	–	reserved	–	–	–	0 when being read.	
		D2	P8TM2	8-bit timer 0-5 interrupt level	0 to 7		X	R/W	
D1	P8TM1	X							
D0	P8TM0	X							

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
8-bit timer 0-3 interrupt enable register	0040275 (B)	D7-4	–	reserved		–	–	–	0 when being read.	
		D3	E8TU3	8-bit timer 3 underflow	1	Enabled	0	Disabled	0	R/W
		D2	E8TU2	8-bit timer 2 underflow					0	R/W
		D1	E8TU1	8-bit timer 1 underflow					0	R/W
		D0	E8TU0	8-bit timer 0 underflow					0	R/W
8-bit timer 4/5 interrupt enable register	0040278 (B)	D7-2	–	reserved		–	–	–	0 when being read.	
		D1	E8TU5	8-bit timer 5 underflow	1	Enabled	0	Disabled	0	R/W
		D0	E8TU4	8-bit timer 4 underflow					0	R/W
8-bit timer 0-3 interrupt factor flag register	0040285 (B)	D7-4	–	reserved		–	–	–	0 when being read.	
		D3	F8TU3	8-bit timer 3 underflow	1	Factor is generated	0	No factor is generated	X	R/W
		D2	F8TU2	8-bit timer 2 underflow					X	R/W
		D1	F8TU1	8-bit timer 1 underflow					X	R/W
		D0	F8TU0	8-bit timer 0 underflow					X	R/W
8-bit timer 4/5 interrupt factor flag register	0040288 (B)	D7-2	–	reserved		–	–	–	0 when being read.	
		D1	F8TU5	8-bit timer 5 underflow	1	Factor is generated	0	No factor is generated	X	R/W
		D0	F8TU4	8-bit timer 4 underflow					X	R/W
16-bit timer 5, 8-bit timer 0-3, serial I/F Ch.0 IDMA request register	0040292 (B)	D7	RSTX0	SIF Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	0	R/W
		D6	RSRX0	SIF Ch.0 receive buffer full					0	R/W
		D5	R8TU3	8-bit timer 3 underflow					0	R/W
		D4	R8TU2	8-bit timer 2 underflow					0	R/W
		D3	R8TU1	8-bit timer 1 underflow					0	R/W
		D2	R8TU0	8-bit timer 0 underflow					0	R/W
		D1	R16TC5	16-bit timer 5 comparison A					0	R/W
		D0	R16TU5	16-bit timer 5 comparison B					0	R/W
		16-bit timer 5, 8-bit timer 0-3, serial I/F Ch.0 IDMA enable register	0040296 (B)	D7	DESTX0	SIF Ch.0 transmit buffer empty	1	IDMA enabled	0	IDMA disabled
D6	DESRX0			SIF Ch.0 receive buffer full					0	R/W
D5	DE8TU3			8-bit timer 3 underflow					0	R/W
D4	DE8TU2			8-bit timer 2 underflow					0	R/W
D3	DE8TU1			8-bit timer 1 underflow					0	R/W
D2	DE8TU0			8-bit timer 0 underflow					0	R/W
D1	DE16TC5			16-bit timer 5 comparison A					0	R/W
D0	DE16TU5			16-bit timer 5 comparison B					0	R/W
8-bit timer 4/5 serial I/F Ch.2/3 IDMA request register	004029B (B)			D7-6	–	reserved		–	–	–
		D5	RSTX3	SIF Ch.3 transmit buffer empty	1	IDMA request	0	Interrupt request	0	R/W
		D4	RSRX3	SIF Ch.3 receive buffer full					0	R/W
		D3	RSTX2	SIF Ch.2 transmit buffer empty					0	R/W
		D2	RSRX2	SIF Ch.2 receive buffer full					0	R/W
		D1	R8TU5	8-bit timer 5 underflow					0	R/W
		D0	R8TU4	8-bit timer 4 underflow					0	R/W
		8-bit timer 4/5 serial I/F Ch.2/3 IDMA enable register	004029C (B)	D7-6	–	reserved		–	–	–
D5	DESTX3			SIF Ch.3 transmit buffer empty	1	IDMA enabled	0	IDMA disabled	0	R/W
D4	DESRX3			SIF Ch.3 receive buffer full					0	R/W
D3	DESTX2			SIF Ch.2 transmit buffer empty					0	R/W
D2	DESRX2			SIF Ch.2 receive buffer full					0	R/W
D1	DE8TU5			8-bit timer 5 underflow					0	R/W
D0	DE8TU4			8-bit timer 4 underflow					0	R/W
P1 function select register	00402D4 (B)			D7	–	reserved		–	–	–
		D6	CFP16	P16 function selection	1	EXCL5 #DMAEND1	0	P16	0	R/W
		D5	CFP15	P15 function selection	1	EXCL4 #DMAEND0	0	P15	0	R/W
		D4	CFP14	P14 function selection	1	FOSC1	0	P14	0	R/W
		D3	CFP13	P13 function selection	1	EXCL3 T8UF3	0	P13	0	R/W
		D2	CFP12	P12 function selection	1	EXCL2 T8UF2	0	P12	0	R/W
		D1	CFP11	P11 function selection	1	EXCL1 T8UF1	0	P11	0	R/W
		D0	CFP10	P10 function selection	1	EXCL0 T8UF0	0	P10	0	R/W

### III-4 PERIPHERAL BLOCK: 8-BIT PROGRAMMABLE TIMERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
P1 I/O control register	00402D6 (B)	D7	–	reserved		–	–	–	0 when being read.		
		D6	IOC16	P16 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D5	IOC15	P15 I/O control					0	R/W	
		D4	IOC14	P14 I/O control					0	R/W	
		D3	IOC13	P13 I/O control					0	R/W	
		D2	IOC12	P12 I/O control					0	R/W	
		D1	IOC11	P11 I/O control					0	R/W	
		D0	IOC10	P10 I/O control					0	R/W	
Port function extension register	00402DF (B)	D7	CFEX7	P07 port extended function	1	#DMAEND3	0	P07, etc.	0	R/W	
		D6	CFEX6	P06 port extended function	1	#DMAACK3	0	P06, etc.	0	R/W	
		D5	CFEX5	P05 port extended function	1	#DMAEND2	0	P05, etc.	0	R/W	
		D4	CFEX4	P04 port extended function	1	#DMAACK2	0	P04, etc.	0	R/W	
		D3	CFEX3	P31 port extended function	1	#GARD	0	P31, etc.	0	R/W	
		D2	CFEX2	P21 port extended function	1	#GAAS	0	P21, etc.	0	R/W	
		D1	CFEX1	P10, P11, P13 port extended function	1	DST0 DST1 DPCO	0	P10, etc. P11, etc. P13, etc.	1	R/W	
		D0	CFEX0	P12, P14 port extended function	1	DST2 DCLK	0	P12, etc. P14, etc.	1	R/W	

#### CFP13–CFP10: P1[3:0] pin function selection (D[3:0]) / P1 function select register (0x402D4)

Selects the pin that is used to output a timer underflow signal to external devices.

Write "1": Underflow signal output pin  
 Write "0": I/O port pin  
 Read: Valid

Select the pin used to output a timer underflow signal to external devices from among P10 through P13 by writing "1" to the corresponding bit, CFP10 through CFP13. P10 through P13 correspond to timers 0 through 3, respectively. If "0" is written to CFP1x, the pin is set for an I/O port.

At cold start, CFP1x is set to "0" (I/O port). At hot start, the bit retains its state from prior to the initial reset.

#### IOC13–IOC10: P1[3:0] port I/O control (D[3:0]) / P1 I/O control register (0x402D6)

Directs P10 through P13 for input or output and indicates the I/O control signal value of the port.

##### When writing data

Write "1": Output mode  
 Write "0": Input mode

If a pin chosen from among P10 through P13 is used to output an underflow signal, write "1" to the corresponding I/O control bit to set it to output mode. If the pin is set to input mode, even if its CFP1x is set to "1", it functions as the event counter input pin of a 16-bit programmable timer cannot be used to output a timer underflow signal.

##### When reading data

Read "1": I/O control signal (output)  
 Read "0": I/O control signal (input)

The I/O control signal value for the port pin is read from this register. When I/O port function is selected using the CFEX and CFP1x registers, the value written to the IOC register is read out as is. When peripheral function is selected, the read value depends on the peripheral circuit status and may not indicate the value written to the IOC register.

At cold start, IOC1x is set to "0" (input mode). At hot start, the bit retains its state from prior to the initial reset.

---

**CFEX1:** P10, P11, P13 port extended function (D1) / Port function extension register (0x402DF)

**CFEX0:** P12, P14 port extended function (D0) / Port function extension register (0x402DF)

---

Sets whether the function of an I/O-port pin is to be extended.

Write "1": Function-extended pin

Write "0": I/O-port/peripheral-circuit pin

Read: Valid

When CFEX[1:0] is set to "1", the P13–P10 ports function as debug signal output ports. When CFEX[1:0] = "0", the CFP1[3:0] bit becomes effective, so the settings of these bits determine whether the P13–P10 ports function as I/O ports or timer underflow signal output ports.

At cold start, CFEX[1:0] is set to "1" (function-extended pins). At hot start, CFEX[1:0] retains its state from prior to the initial reset.

---

**RLD07–RLD00:** Timer 0 reload data (D[7:0]) / 8-bit timer 0 reload data register (0x40161)

**RLD17–RLD10:** Timer 1 reload data (D[7:0]) / 8-bit timer 1 reload data register (0x40165)

**RLD27–RLD20:** Timer 2 reload data (D[7:0]) / 8-bit timer 2 reload data register (0x40169)

**RLD37–RLD30:** Timer 3 reload data (D[7:0]) / 8-bit timer 3 reload data register (0x4016D)

**RLD47–RLD40:** Timer 4 reload data (D[7:0]) / 8-bit timer 4 reload data register (0x40175)

**RLD57–RLD50:** Timer 5 reload data (D[7:0]) / 8-bit timer 5 reload data register (0x40179)

---

Set the initial counter value of each timer.

The reload data set in this register is loaded into each counter, and the counter starts counting down beginning with this data, which is used as the initial count.

There are two cases in which the reload data is loaded into the counter: when data is preset after "1" is written to PSETx, or when data is automatically reloaded upon counter underflow.

At initial reset, RLD is not initialized.

---

**PTD07–PTD00:** Timer 0 counter data (D[7:0]) / 8-bit timer 0 counter data (0x40162)

**PTD17–PTD10:** Timer 1 counter data (D[7:0]) / 8-bit timer 1 counter data (0x40166)

**PTD27–PTD20:** Timer 2 counter data (D[7:0]) / 8-bit timer 2 counter data (0x4016A)

**PTD37–PTD30:** Timer 3 counter data (D[7:0]) / 8-bit timer 3 counter data (0x4016E)

**PTD47–PTD40:** Timer 4 counter data (D[7:0]) / 8-bit timer 4 counter data (0x40176)

**PTD57–PTD50:** Timer 5 counter data (D[7:0]) / 8-bit timer 5 counter data (0x4017A)

---

The 8-bit programmable timer data can be read out from these bits.

These bits function as buffers that retain the counter data when read out, enabling the data to be read out at any time.

At initial reset, PTD is not initialized.

---

**PSET0:** Timer 0 preset (D1) / 8-bit timer 0 control register (0x40160)

**PSET1:** Timer 1 preset (D1) / 8-bit timer 1 control register (0x40164)

**PSET2:** Timer 2 preset (D1) / 8-bit timer 2 control register (0x40168)

**PSET3:** Timer 3 preset (D1) / 8-bit timer 3 control register (0x4016C)

**PSET4:** Timer 4 preset (D1) / 8-bit timer 4 control register (0x40174)

**PSET5:** Timer 5 preset (D1) / 8-bit timer 5 control register (0x40178)

---

Preset the reload data in the counter.

Write "1": Preset

Write "0": Invalid

Read: Always "0"

The reload data of RLDx is preset in the counter of timer x by writing "1" to PSETx. If the counter is preset when in a RUN state, the counter starts counting immediately after the reload data is preset.

If the counter is preset when in a STOP state, the reload data that has been preset is retained.

Writing "0" results in No Operation.

Since PSETx is a write-only bit, its content when read is always "0".

### III-4 PERIPHERAL BLOCK: 8-BIT PROGRAMMABLE TIMERS

---

**PTRUN0:** Timer 0 RUN/STOP control (D0) / 8-bit timer 0 control register (0x40160)

**PTRUN1:** Timer 1 RUN/STOP control (D0) / 8-bit timer 1 control register (0x40164)

**PTRUN2:** Timer 2 RUN/STOP control (D0) / 8-bit timer 2 control register (0x40168)

**PTRUN3:** Timer 3 RUN/STOP control (D0) / 8-bit timer 3 control register (0x4016C)

**PTRUN4:** Timer 4 RUN/STOP control (D0) / 8-bit timer 4 control register (0x40174)

**PTRUN5:** Timer 5 RUN/STOP control (D0) / 8-bit timer 5 control register (0x40178)

---

Controls the counter's RUN/STOP states.

Write "1": RUN

Write "0": STOP

Read: Valid

The counter of each timer starts counting down when "1" written to PTRUNx, and stops counting when "0" is written.

While in a STOP state, the counter retains its count until it is preset with reload data or placed in a RUN state. When the state is changed from STOP to RUN, the counter can restart counting beginning with the retained count.

At initial reset, PTRUNx is set to "0" (STOP).

---

**PTOUT0:** Timer 0 clock output control register (D2) / 8-bit timer 0 control register (0x40160)

**PTOUT1:** Timer 1 clock output control register (D2) / 8-bit timer 1 control register (0x40164)

**PTOUT2:** Timer 2 clock output control register (D2) / 8-bit timer 2 control register (0x40168)

**PTOUT3:** Timer 3 clock output control register (D2) / 8-bit timer 3 control register (0x4016C)

**PTOUT4:** Timer 4 clock output control register (D2) / 8-bit timer 4 control register (0x40174)

**PTOUT5:** Timer 5 clock output control register (D2) / 8-bit timer 5 control register (0x40178)

---

Controls the clock output of each timer.

Write "1": On

Write "0": Off

Read: Valid

The underflow signal of timer x (timer 0 to 3) is output from the external output pin set by CFP1x by writing "1" to PTOUTx. When using timer 2 to 5 as the clock source of the serial interface, a clock generated from the underflow signal by dividing it by 2 is output to the corresponding channel of the serial interface.

The clock output is turned off by writing "0" to PTOUT, and the external output is fixed at "0" and the internal clock output is fixed at "1".

At initial reset, PTOUT is set to "0" (off).

---

**P8TM2–P8TM0:** 8-bit timer interrupt level (D[2:0]) / 8-bit timer, serial I/F Ch.0 interrupt priority register (0x40269)

Set the priority level of the 8-bit programmable timer interrupt in the range of 0 to 7.

At initial reset, the content of the P8TM register becomes indeterminate.

---

**E8TU0:** Timer 0 interrupt enable (D0) / 8-bit timer 0–3 interrupt enable register (0x40275)

**E8TU1:** Timer 1 interrupt enable (D1) / 8-bit timer 0–3 interrupt enable register (0x40275)

**E8TU2:** Timer 2 interrupt enable (D2) / 8-bit timer 0–3 interrupt enable register (0x40275)

**E8TU3:** Timer 3 interrupt enable (D3) / 8-bit timer 0–3 interrupt enable register (0x40275)

**E8TU4:** Timer 4 interrupt enable (D0) / 8-bit timer 4/5 interrupt enable register (0x40278)

**E8TU5:** Timer 5 interrupt enable (D1) / 8-bit timer 4/5 interrupt enable register (0x40278)

---

Enables or disables generation of an interrupt to the CPU.

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

E8TUX is the interrupt enable bit which controls the interrupt generated by each timer. The interrupt set to "1" by this bit is enabled, and the interrupt set to "0" by this bit is disabled.

At initial reset, E8TUX is set to "0" (interrupt disabled).



---

**F8TU0:** Timer 0 interrupt factor flag (D0) / 8-bit timer 0–3 interrupt factor flag register (0x40285)

**F8TU1:** Timer 1 interrupt factor flag (D1) / 8-bit timer 0–3 interrupt factor flag register (0x40285)

**F8TU2:** Timer 2 interrupt factor flag (D2) / 8-bit timer 0–3 interrupt factor flag register (0x40285)

**F8TU3:** Timer 3 interrupt factor flag (D3) / 8-bit timer 0–3 interrupt factor flag register (0x40285)

**F8TU4:** Timer 4 interrupt factor flag (D0) / 8-bit timer 4/5 interrupt factor flag register (0x40288)

**F8TU5:** Timer 5 interrupt factor flag (D1) / 8-bit timer 4/5 interrupt factor flag register (0x40288)

---

Indicates the interrupt generation status of the 8-bit programmable timer.

#### When read

Read "1": Interrupt factor has occurred

Read "0": No interrupt factor has occurred

#### When written using the reset-only method (default)

Write "1": Interrupt factor flag is reset

Write "0": Invalid

#### When written using the read/write method

Write "1": Interrupt flag is set

Write "0": Interrupt flag is reset

F8TUX is the interrupt factor flag corresponding to each timer. It is set to "1" when the counter underflows.

At this time, if the following conditions are met, an interrupt to the CPU is generated:

1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The IE bit of the PSR is set to "1" (interrupts enabled).
4. The value set in the corresponding interrupt priority register is higher than the interrupt level (IL) of the CPU.

When using the interrupt factor of the 8-bit programmable timer to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of how the interrupt enable and interrupt priority registers are set.

If the next interrupt is to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).

The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.

At initial reset, the content of F8TUX becomes indeterminate, so be sure to reset it in the software.

---

**R8TU0:** Timer 0 IDMA request (D2) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA request register (0x40292)  
**R8TU1:** Timer 1 IDMA request (D3) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA request register (0x40292)  
**R8TU2:** Timer 2 IDMA request (D4) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA request register (0x40292)  
**R8TU3:** Timer 3 IDMA request (D5) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA request register (0x40292)  
**R8TU4:** Timer 4 IDMA request (D0) / 8-bit timer 4/5, serial I/F Ch.2/3 IDMA request register (0x4029B)  
**R8TU5:** Timer 5 IDMA request (D1) / 8-bit timer 4/5, serial I/F Ch.2/3 IDMA request register (0x4029B)

---

Specifies whether IDMA is to be invoked at the occurrence of an interrupt factor.

**When using the set-only method (default)**

Write "1": IDMA request  
 Write "0": Not changed  
 Read: Valid

**When using the read/write method**

Write "1": IDMA request  
 Write "0": Interrupt request  
 Read: Valid

R8TUx is the IDMA request bit for each timer. If this bit is set to "1", IDMA can be invoked when an interrupt factor occurs, and thus programmed data transfers are performed. If the bit is set to "0", normal interrupt processing is performed and IDMA is not invoked.

For details on IDMA, refer to "IDMA (Intelligent DMA)".

At initial reset, R8TUx is set to "0" (interrupt request).

---

**DE8TU0:** Timer 0 IDMA enable (D2) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA enable register (0x40296)  
**DE8TU1:** Timer 1 IDMA enable (D3) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA enable register (0x40296)  
**DE8TU2:** Timer 2 IDMA enable (D4) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA enable register (0x40296)  
**DE8TU3:** Timer 3 IDMA enable (D5) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA enable register (0x40296)  
**DE8TU4:** Timer 4 IDMA enable (D0) / 8-bit timer 4/5, serial I/F Ch.2/3 IDMA enable register (0x4029C)  
**DE8TU5:** Timer 5 IDMA enable (D1) / 8-bit timer 4/5, serial I/F Ch.2/3 IDMA enable register (0x4029C)

---

Enables IDMA transfer by means of an interrupt factor.

**When using the set-only method (default)**

Write "1": IDMA enabled  
 Write "0": Not changed  
 Read: Valid

**When using the read/write method**

Write "1": IDMA enabled  
 Write "0": IDMA disabled  
 Read: Valid

If DE8TUx is set to "1", the IDMA request by the interrupt factor is enabled. If the register bit is set to "0", the IDMA request is disabled.

After an initial reset, DE8TUx is set to "0" (IDMA disabled).

## ***Programming Notes***

---

- (1) The 8-bit programmable timer operates only when the prescaler is operating.
- (2) Do not use a clock that is faster than the CPU operating clock for the 8-bit programmable timer.
- (3) When setting an input clock, make sure the 8-bit programmable timer is turned off.
- (4) Since the underflow interrupt condition and the timer output status are undefined after an initial reset, the counter initial value should be set to the 8-bit timer before resetting the interrupt factor flag or turning the timer output on.
- (5) After an initial reset, the interrupt factor flag (F8TUx) becomes indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag in the software.
- (6) To prevent another interrupt from being generated again by the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag (F8TUx) before setting the PSR again or executing the reti instruction.

THIS PAGE IS BLANK.

## III-5 16-BIT PROGRAMMABLE TIMERS

### Configuration of 16-Bit Programmable Timer

The Peripheral Block contains six systems of 16-bit programmable timers (timers 0 to 5). They also have an event counter function using an I/O port pin.

**Note:** On the following pages, each timer is identified as timer  $x$  ( $x = 0$  to  $5$ ). The functions and control register structures of 16-bit programmable timers 0 to 5 are the same. Control bit names are assigned numerals "0" to "5" denoting timer numbers. Since explanations are common to all timers, timer numbers are represented by "x" unless it is necessary to specify a timer number.

Figure III.5.1 shows the structure of one channel of the 16-bit programmable timer.

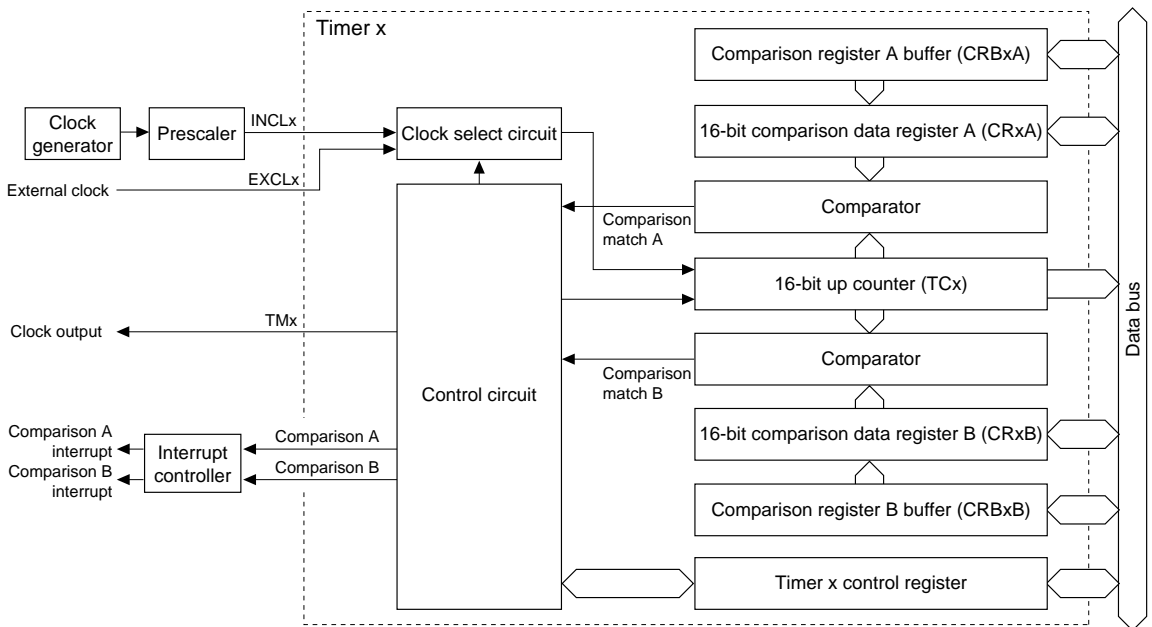


Figure III.5.1 Structure of 16-Bit Programmable Timer

In each timer, a 16-bit up-counter (TCx), as well as two 16-bit comparison data registers (CRxA, CRxB) and their buffers (CRBxA, CRBxB), are provided.

The 16-bit counter can be reset to "0" by software and counts up using the prescaler output clock or an external signal input from the I/O port. The counter value can be read by software.

The comparison data registers A and B are used to store the data to be compared with the content of the up-counter. This register can be directly read and written. Furthermore, comparison data can be set via the comparison register buffer. In this case, the set value is loaded to the comparison data register when the counter is reset by the comparison match B signal or software (by writing "1" to PRESETx bit). The software can select whether comparison data is written to the comparison data register or the buffer.

When the counter value matches to the content of each comparison data register, the comparator outputs a signal that controls the interrupt and the output signal. Thus the registers allow interrupt generating intervals and the timer's output clock frequency and duty ratio to be programmed.

## I/O Pins of 16-Bit Programmable Timers

Table III.5.1 shows the input/output pins used for the 16-bit programmable timers.

Table III.5.1 I/O Pins of 16-Bit Programmable Timer

Pin name	I/O	Function	Function select bit
DST0(P10/EXCL0/T8UF0)	I/O	DST0 output (Ex) / I/O port / 16-bit timer 0 event counter input (I) / 8-bit timer 0 output (O)	CFP10(D0)/P1 function select register(0x402D4) CFEX1(D1)/Port function extension register(0x402DF)
DST1(P11/EXCL1/T8UF1)	I/O	DST1 output (Ex) / I/O port / 16-bit timer 1 event counter input (I) / 8-bit timer 1 output (O)	CFP11(D1)/P1 function select register(0x402D4) CFEX1(D1)/Port function extension register(0x402DF)
DST2(P12/EXCL2/T8UF2)	I/O	DST2 output (Ex) / I/O port / 16-bit timer 2 event counter input (I) / 8-bit timer 2 output (O)	CFP12(D2)/P1 function select register(0x402D4) CFEX0(D0)/Port function extension register(0x402DF)
DPCO(P13/EXCL3/T8UF3)	I/O	DPCO output (Ex) / I/O port / 16-bit timer 3 event counter input (I) / 8-bit timer 3 output (O)	CFP13(D3)/P1 function select register(0x402D4) CFEX1(D1)/Port function extension register(0x402DF)
P15(EXCL4/#DMAEND0)	I/O	I/O port / 16-bit timer 4 event counter input (I) / High-speed DMA Ch.0 end signal output (O)	CFP15(D5)/P1 function select register(0x402D4)
P16(EXCL5/#DMAEND1)	I/O	I/O port / 16-bit timer 5 event counter input (I) / High-speed DMA Ch.1 end signal output (O)	CFP16(D6)/P1 function select register(0x402D4)
P22(TM0)	I/O	I/O port / 16-bit timer 0 output	CFP22(D2)/P2 function select register(0x402D8)
P23(TM1)	I/O	I/O port / 16-bit timer 1 output	CFP23(D3)/P2 function select register(0x402D8)
P24(TM2)	I/O	I/O port / 16-bit timer 2 output	CFP24(D4)/P2 function select register(0x402D8)
P25(TM3)	I/O	I/O port / 16-bit timer 3 output	CFP25(D5)/P2 function select register(0x402D8)
P26(TM4)	I/O	I/O port / 16-bit timer 4 output	CFP26(D6)/P2 function select register(0x402D8)
P27(TM5)	I/O	I/O port / 16-bit timer 5 output	CFP27(D7)/P2 function select register(0x402D8)

(I): Input mode, (O): Output mode, (Ex): Extended function

### TMx (output pin of the 16-bit programmable timer)

This pin outputs a clock generated by the timer x.

### EXCLx (event counter input pin)

When using the timer x as an event counter, input count pulses from an external source to this pin.

### How to set the input/output pins of 16-bit programmable timers

All clock output pins used by the 16-bit programmable timers are shared with I/O ports. At cold start, all these pins are set for the I/O port pins P2x (function select bit CFP2x = "0"), and go into high-impedance.

When using the clock output function of the 16-bit programmable timer, select the desired timer and write "1" to the function select bit CFP2x for the corresponding pin. At hot start, these pins retain their status before from prior to the reset.

All event-counter input pins are also shared with I/O-ports. At cold start, the EXCL[3:0] pins are set for debug signal output pins (function extension bit CFEX[1:0] = "1") and the EXCL[5:4] pins are set for I/O-port pins P1[5:4] (function select bit CFP1[5:4] = "0"). When using the event counter function, select the desired timer and write "1" to the function select bit CFP1x and write "0" to the function select bit CFEXx for the corresponding pin.

Note that these pins are also shared with output pins for the 8-bit programmer timers, etc. When the input/output pins are set in input mode, they function as event counter inputs. Therefore, it is necessary to set the I/O port's I/O control bit IOC1x to "0" in advance. At cold start, these pins are set in input mode. At hot start, they retain their status from prior to the reset.

## ***Uses of 16-Bit Programmable Timers***

---

The up-counters of the 16-bit programmable timer cyclically output a comparison-match signal in accordance with the comparison data that are set in the software. This signal is used to generate an interrupt request to the CPU or control the internal peripheral circuits. A clock generated from the signal can also be output to external devices.

### **CPU interrupt request/IDMA invocation request**

Each timer's comparison match (matching of counter and comparison data) can be used as an interrupt factor to generate an interrupt request to the CPU. Therefore, an interrupt can be generated at an interval that is set in the software.

Furthermore, this interrupt factor can also be used to invoke IDMA or HSDMA.

### **Clock output to external devices**

A clock generated from the comparison-match signal can be output from the chip to the outside. The clock cycle is determined by comparison data B, and the duty ratio is determined by comparison data A. This output can be used to control external devices. The output pins of each timer are described in the preceding section.

### **A/D converter start trigger**

The A/D converter allows a trigger to start the A/D conversion to be selected from among four available types. One is the comparison-match B of the 16-bit programmable timer 0. This makes it possible to perform the A/D conversion at programmable intervals.

To use this function, write "01" to the A/D converter control TS[1:0] (D[4:3]) / A/D trigger register (0x40242) to select the 16-bit programmable timer 0 as the trigger.

### **Watchdog timer**

The 16-bit programmable timer 0 can be used as a watchdog timer to monitor CPU crash. In this case, the comparison-match B of this timer serves as an NMI request signal to the CPU.

To use this function, write "1" to the watchdog timer control bit EWD (D1) / Watchdog timer enable register (0x40171) to enable the NMI. For details on how to control the watchdog timer, refer to "Watchdog Timer".

## Control and Operation of 16-Bit Programmable Timer

The following settings must first be made before the 16-bit programmable timer starts counting:

1. Setting pins for input/output (only when necessary)
2. Setting input clock
3. Selecting comparison data register/buffer
4. Setting clock output conditions (signal active level, fine mode)
5. Setting comparison data
6. Setting interrupt/DMA

For details on how to set clock output conditions and interrupts and DMA, refer to "Controlling Clock Output" and "16-Bit Programmable Timer Interrupts and DMA".

### Setting pin for input/output

The pin must be set for output for the output clock of the 16-bit programmable timer to be fed to external devices.

The pin for input must be set for the 16-bit programmable timer to be used as an event counter that counts external clock pulses.

For details on how to set the pin, refer to "I/O Pins of 16-Bit Programmable Timers".

### Setting the input clock

The count clock for each timer can be selected from between an internal clock and an external clock. Use the following control bits to select the input clock:

- Timer 0 input clock selection: CKSL0 (D3) / 16-bit timer 0 control register (0x48186)
- Timer 1 input clock selection: CKSL1 (D3) / 16-bit timer 1 control register (0x4818E)
- Timer 2 input clock selection: CKSL2 (D3) / 16-bit timer 2 control register (0x48196)
- Timer 3 input clock selection: CKSL3 (D3) / 16-bit timer 3 control register (0x4819E)
- Timer 4 input clock selection: CKSL4 (D3) / 16-bit timer 4 control register (0x481A6)
- Timer 5 input clock selection: CKSL5 (D3) / 16-bit timer 5 control register (0x481AE)

An external clock is selected by writing "1" to CKSLx, and the internal clock is selected by writing "0".

At initial reset, CKSLx is set for the internal clock.

An external clock can be used for the timer for which the pin is set for input.

### Internal clock

When the internal clock is selected as a timer, the timer is operated by the prescaler output clock. The prescaler division ratio can be selected for each timer.

Table III.5.2 Setting the Internal Clock

Timer	Control register	Division ratio select bit	Clock control bit
Timer 0	16-bit timer 0 clock control register (0x40147)	P16TS0[2:0] (D[2:0])	P16TON0 (D3)
Timer 1	16-bit timer 1 clock control register (0x40148)	P16TS1[2:0] (D[2:0])	P16TON1 (D3)
Timer 2	16-bit timer 2 clock control register (0x40149)	P16TS2[2:0] (D[2:0])	P16TON2 (D3)
Timer 3	16-bit timer 3 clock control register (0x4014A)	P16TS3[2:0] (D[2:0])	P16TON3 (D3)
Timer 4	16-bit timer 4 clock control register (0x4014B)	P16TS4[2:0] (D[2:0])	P16TON4 (D3)
Timer 5	16-bit timer 5 clock control register (0x4014C)	P16TS5[2:0] (D[2:0])	P16TON5 (D3)

The division ratio can be selected from among eight types as shown in Table III.5.3.

Table III.5.3 Input Clock Selection

P16TS = 7	P16TS = 6	P16TS = 5	P16TS = 4	P16TS = 3	P16TS = 2	P16TS = 1	P16TS = 0
fPSCIN/4096	fPSCIN/1024	fPSCIN/256	fPSCIN/64	fPSCIN/16	fPSCIN/4	fPSCIN/2	fPSCIN/1

fPSCIN: Prescaler input clock frequency

The selected clock is output from the prescaler to the 16-bit programmable timer by writing "1" to P16TONx.

- Notes:**
- When the internal clock is used, the 16-bit programmable timer operates only when the prescaler is operating (refer to "Prescaler").
  - When setting an input clock, make sure the 16-bit programmable timer is turned off.



### External clock

When using the timer as an event counter by supplying clock pulses from an external source, make sure the event cycle is at least the CPU operating clock period.

### Selecting comparison data register/buffer

The comparison data registers A and B are used to store the data to be compared with the content of the up-counter. This register can be directly read and written. Furthermore, comparison data can be set via the comparison register buffer. In this case, the set value is loaded to the comparison data register when the counter is reset by the comparison match B signal or software (by writing "1" to PRESETx bit).

Select whether comparison data is written to the comparison data register or the buffer using the following control bits:

Timer 0 comparison register buffer enable: SELCRB0 (D5) / 16-bit timer 0 control register (0x48186)

Timer 1 comparison register buffer enable: SELCRB1 (D5) / 16-bit timer 1 control register (0x4818E)

Timer 2 comparison register buffer enable: SELCRB2 (D5) / 16-bit timer 2 control register (0x48196)

Timer 3 comparison register buffer enable: SELCRB3 (D5) / 16-bit timer 3 control register (0x4819E)

Timer 4 comparison register buffer enable: SELCRB4 (D5) / 16-bit timer 4 control register (0x481A6)

Timer 5 comparison register buffer enable: SELCRB5 (D5) / 16-bit timer 5 control register (0x481AE)

When "1" is written to SELCRBx, the comparison register buffer is selected and when "0" is written, the comparison data register is selected.

At initial reset, the comparison data register is selected.

### Setting comparison data

The programmable timer contains two data comparators that allows the count data to be compared with given values. The following registers are used to set these values.

Timer 0 comparison data A: CR0A[15:0] (D[F:0]) / 16-bit timer 0 comparison data A set-up register (0x48180)

Timer 0 comparison data B: CR0B[15:0] (D[F:0]) / 16-bit timer 0 comparison data B set-up register (0x48182)

Timer 1 comparison data A: CR1A[15:0] (D[F:0]) / 16-bit timer 1 comparison data A set-up register (0x48188)

Timer 1 comparison data B: CR1B[15:0] (D[F:0]) / 16-bit timer 1 comparison data B set-up register (0x4818A)

Timer 2 comparison data A: CR2A[15:0] (D[F:0]) / 16-bit timer 2 comparison data A set-up register (0x48190)

Timer 2 comparison data B: CR2B[15:0] (D[F:0]) / 16-bit timer 2 comparison data B set-up register (0x48192)

Timer 3 comparison data A: CR3A[15:0] (D[F:0]) / 16-bit timer 3 comparison data A set-up register (0x48198)

Timer 3 comparison data B: CR3B[15:0] (D[F:0]) / 16-bit timer 3 comparison data B set-up register (0x4819A)

Timer 4 comparison data A: CR4A[15:0] (D[F:0]) / 16-bit timer 4 comparison data A set-up register (0x481A0)

Timer 4 comparison data B: CR4B[15:0] (D[F:0]) / 16-bit timer 4 comparison data B set-up register (0x481A2)

Timer 5 comparison data A: CR5A[15:0] (D[F:0]) / 16-bit timer 5 comparison data A set-up register (0x481A8)

Timer 5 comparison data B: CR5B[15:0] (D[F:0]) / 16-bit timer 5 comparison data B set-up register (0x481AA)

When SELCRBx is set to "0", these registers allow direct reading/writing from/to the comparison data register.

When SELCRBx is set to "1", these registers are used to read/write from/to the comparison register buffer. The content of the buffer is loaded to the comparison data register when the counter is reset.

At initial reset, the comparison data registers/buffers are not initialized.

The programmable timer compares the comparison data register and count data and, when the two values are equal, generates a comparison match signal. This comparison match signal controls the clock output (TMx signal) to external devices, in addition to generating an interrupt.

The comparison data B is also used to reset the counter.

### Resetting the counter

Each timer includes the PRESETx bit to reset the counter.

Timer 0 reset: PRESET0 (D1) / 16-bit timer 0 control register (0x48186)

Timer 1 reset: PRESET1 (D1) / 16-bit timer 1 control register (0x4818E)

Timer 2 reset: PRESET2 (D1) / 16-bit timer 2 control register (0x48196)

Timer 3 reset: PRESET3 (D1) / 16-bit timer 3 control register (0x4819E)

Timer 4 reset: PRESET4 (D1) / 16-bit timer 4 control register (0x481A6)

Timer 5 reset: PRESET5 (D1) / 16-bit timer 5 control register (0x481AE)

Normally, reset the counter before starting count-up by writing "1" to this control bit.

After the counter starts counting, it will be reset by comparison match B.

### Timer RUN/STOP control

Each timer includes the PRUNx bit to control RUN/STOP.

Timer 0 RUN/STOP control: PRUN0 (D0) / 16-bit timer 0 control register (0x48186)

Timer 1 RUN/STOP control: PRUN1 (D0) / 16-bit timer 1 control register (0x4818E)

Timer 2 RUN/STOP control: PRUN2 (D0) / 16-bit timer 2 control register (0x48196)

Timer 3 RUN/STOP control: PRUN3 (D0) / 16-bit timer 3 control register (0x4819E)

Timer 4 RUN/STOP control: PRUN4 (D0) / 16-bit timer 4 control register (0x481A6)

Timer 5 RUN/STOP control: PRUN5 (D0) / 16-bit timer 5 control register (0x481AE)

The timer starts counting when "1" is written to PRUNx. The clock input is disabled and the timer stops counting when "0" is written to PRUNx.

This RUN/STOP control does not affect the counter data. Even when the timer has stopped counting, the counter retains its count so that the timer can start counting again from that point.

If the count of the counter matches the set value of the comparison data register during count-up, the timer generates a comparison match interrupt.

When the counter matches comparison data B, an interrupt is generated and the counter is reset. At the same time, the values set in the compare register buffer are loaded to the compare data register if SELCRBx is set to "1".

The counter continues counting up regardless of which interrupt has occurred. In the case of a comparison B interrupt, the counter starts counting beginning with 0.

When both the timer RUN/STOP control bit (PRUNx) and the timer reset bit (PRESETx) are set to "1" at the same time, the timer starts counting after resetting the counter.

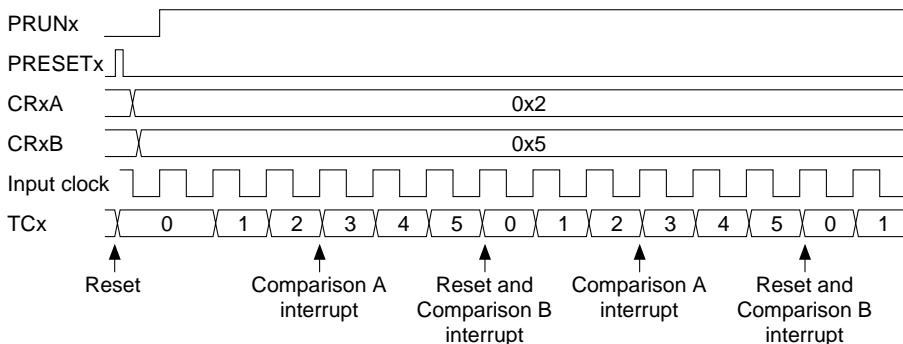


Figure III.5.2 Basic Operation Timing of Counter

### Reading counter data

The counter data can be read out from the following addresses shown below at any time:

Timer 0 counter data: TC0[15:0] (D[F:0]) / 16-bit timer 0 counter data register (0x48184)

Timer 1 counter data: TC1[15:0] (D[F:0]) / 16-bit timer 1 counter data register (0x4818C)

Timer 2 counter data: TC2[15:0] (D[F:0]) / 16-bit timer 2 counter data register (0x48194)

Timer 3 counter data: TC3[15:0] (D[F:0]) / 16-bit timer 3 counter data register (0x4819C)

Timer 4 counter data: TC4[15:0] (D[F:0]) / 16-bit timer 4 counter data register (0x481A4)

Timer 5 counter data: TC5[15:0] (D[F:0]) / 16-bit timer 5 counter data register (0x481AC)

## Controlling Clock Output

The timers can generate a TMx signal using the comparison match signals from the counter.

### Setting the signal active level

By default, an active high signal (normal low) is generated. This logic can be inverted using the OUTINVx bit. When "1" is written to the OUTINVx bit, the timer generates an active low (normal high) signal.

Timer 0 clock output inversion: OUTINV0 (D4) / 16-bit timer 0 control register (0x48186)

Timer 1 clock output inversion: OUTINV1 (D4) / 16-bit timer 1 control register (0x4818E)

Timer 2 clock output inversion: OUTINV2 (D4) / 16-bit timer 2 control register (0x48196)

Timer 3 clock output inversion: OUTINV3 (D4) / 16-bit timer 3 control register (0x4819E)

Timer 4 clock output inversion: OUTINV4 (D4) / 16-bit timer 4 control register (0x481A6)

Timer 5 clock output inversion: OUTINV5 (D4) / 16-bit timer 5 control register (0x481AE)

See Figure III.5.3 for the waveforms.

### Setting the output port

The TMx signal generated here can be output from the clock output pins (see Table III.5.1), enabling a programmable clock to be supplied to external devices.

After a cold start, the output pins are set for the I/O ports and set in input mode. The pins go into high-impedance status.

When the pin function is switched to the timer output, the pin goes low if OUTINVx is set to "0" or goes high if OUTINVx is set to "1".

### Starting clock output

To output the TMx clock, write "1" to the clock output control bit PTMx. Clock output is stopped by writing "0" to PTMx and goes to the off level according to the OUTINVx setting (low when OUTINVx = "0" or high when OUTINVx = "1").

Timer 0 clock output control: PTM0 (D2) / 16-bit timer 0 control register (0x48186)

Timer 1 clock output control: PTM1 (D2) / 16-bit timer 1 control register (0x4818E)

Timer 2 clock output control: PTM2 (D2) / 16-bit timer 2 control register (0x48196)

Timer 3 clock output control: PTM3 (D2) / 16-bit timer 3 control register (0x4819E)

Timer 4 clock output control: PTM4 (D2) / 16-bit timer 4 control register (0x481A6)

Timer 5 clock output control: PTM5 (D2) / 16-bit timer 5 control register (0x481AE)

Figure III.5.3 shows the waveform of the output signal.

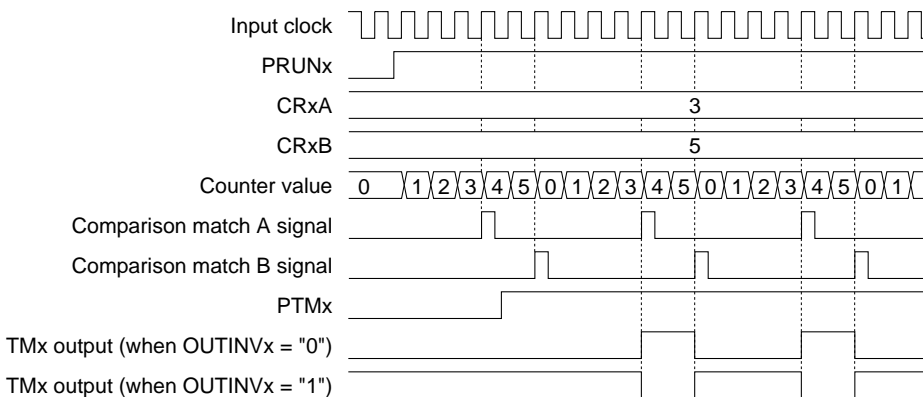


Figure III.5.3 Waveform of 16-Bit Programmable Timer Output

**When OUTINVx = "0" (active high):**

The timer outputs a low level until the counter becomes equal to the comparison data A set in the CRxA register. When the counter is incremented to the next value from the comparison data A, the output pin goes high and a comparison A interrupt occurs. When the counter becomes equal to the comparison data B set in the CRxB register, the counter is reset and the output pin goes low. At the same time a comparison B interrupt occurs.

**When OUTINVx = "1" (active low):**

The timer outputs a high level until the counter becomes equal to the comparison data A set in the CRxA register. When the counter is incremented to the next value from the comparison data A, the output pin goes low and a comparison A interrupt occurs. When the counter becomes equal to the comparison data B set in the CRxB register, the counter is reset and the output pin goes high. At the same time a comparison B interrupt occurs.

**Setting clock output fine mode**

By default (after an initial reset), the clock output signal changes at the rising edge of the input clock when CRxA[15:0] becomes equal to TCx[15:0].

In fine mode, the output signal changes according to CRxA[0] when CRxA[15:1] becomes equal to TCx[14:0].

When CRxA[0] is "0", the output signal changes at the rising edge of the input clock.

When CRxA[0] is "1", the output signal changes at the falling edge of the input clock a half cycle from the default setting.

Example) CRxA = 3, CRxB = 5

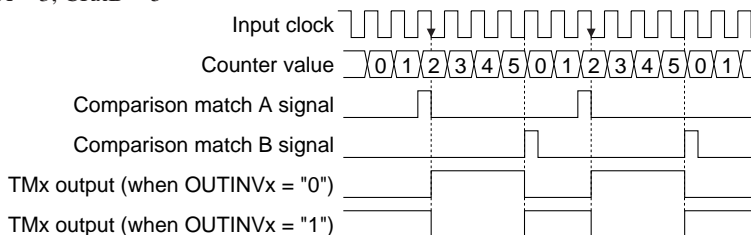


Figure III.5.4 Clock Output in Fine Mode

As shown in the figure above, in fine mode the output clock duty ratio can be adjusted in the half cycle of the input clock. However, when the CRxA value is "0", the timer outputs a pulse with a 1-cycle width as the input clock, the same as the default setting.

In fine mode, the maximum value of CRxB is  $2^{15} - 1 = 32,767$  and the range of CRxA that can be set is 0 to  $(2 \times CRxB - 1)$ .

The fine mode is set by the following registers:

Timer 0 fine mode selection: SELFM0 (D6) / 16-bit timer 0 control register (0x48186)

Timer 1 fine mode selection: SELFM1 (D6) / 16-bit timer 1 control register (0x4818E)

Timer 2 fine mode selection: SELFM2 (D6) / 16-bit timer 2 control register (0x48196)

Timer 3 fine mode selection: SELFM3 (D6) / 16-bit timer 3 control register (0x4819E)

Timer 4 fine mode selection: SELFM4 (D6) / 16-bit timer 4 control register (0x481A6)

Timer 5 fine mode selection: SELFM5 (D6) / 16-bit timer 5 control register (0x481AE)

When "1" is written to the SELFMx bit, fine mode is set. At initial reset, the fine mode is disabled.

**Precautions**

- 1) If a same value is set to the comparison data A and B registers, a hazard may be generated in the output signal. Therefore, do not set the comparison registers as A = B.  
There is no problem when the interrupt function only is used.
- 2) When using the output clock, set the comparison data registers as  $A \geq 0$  and  $B \geq 1$ . The minimum settings are A = 0 and B = 1. In this case, the timer output clock cycle is the input clock  $\times 1/2$ .
- 3) When the comparison data registers are set as  $A > B$ , no comparison A signal is generated. In this case, the output signal is fixed at the off level.

## 16-Bit Programmable Timer Interrupts and DMA

The 16-bit programmable timer has a function for generating an interrupt using the comparison match A and B states.

The timing at which an interrupt is generated is shown in Figure III.5.2 in the preceding section.

### Control registers of the interrupt controller

Table III.5.4 shows the control registers of the interrupt controller provided for each timer.

Table III.5.4 Control Registers of Interrupt Controller

Interrupt factor	Interrupt factor flag	Interrupt enable register	Interrupt priority register
Timer 0 comparison A	F16TC0 (D3/0x40282)	E16TC0 (D3/0x40272)	P16T0[2:0] (D[2:0]/0x40266)
Timer 0 comparison B	F16TU0 (D2/0x40282)	E16TU0 (D2/0x40272)	
Timer 1 comparison A	F16TC1 (D7/0x40282)	E16TC1 (D7/0x40272)	P16T1[2:0] (D[6:4]/0x40266)
Timer 1 comparison B	F16TU1 (D6/0x40282)	E16TU1 (D6/0x40272)	
Timer 2 comparison A	F16TC2 (D3/0x40283)	E16TC2 (D3/0x40273)	P16T2[2:0] (D[2:0]/0x40267)
Timer 2 comparison B	F16TU2 (D2/0x40283)	E16TU2 (D2/0x40273)	
Timer 3 comparison A	F16TC3 (D7/0x40283)	E16TC3 (D7/0x40273)	P16T3[2:0] (D[6:4]/0x40267)
Timer 3 comparison B	F16TU3 (D6/0x40283)	E16TU3 (D6/0x40273)	
Timer 4 comparison A	F16TC4 (D3/0x40284)	E16TC4 (D3/0x40274)	P16T4[2:0] (D[2:0]/0x40268)
Timer 4 comparison B	F16TU4 (D2/0x40284)	E16TU4 (D2/0x40274)	
Timer 5 comparison A	F16TC5 (D7/0x40284)	E16TC5 (D7/0x40274)	P16T5[2:0] (D[6:4]/0x40268)
Timer 5 comparison B	F16TU5 (D6/0x40284)	E16TU5 (D6/0x40274)	

When a comparison match state occurs in the timer, the corresponding interrupt factor flag is set to "1".

If the interrupt enable register bit corresponding to that interrupt factor flag has been set to "1", an interrupt request is generated.

An interrupt caused by a timer can be disabled by leaving the interrupt enable register bit for that timer set to "0". The interrupt factor flag is always set to "1" by the timer's comparison match state, regardless of how the interrupt enable register is set (even when set to "0").

The interrupt priority register sets an interrupt priority level (0 to 7) for each timer. Priorities within a timer block are such that timers of smaller numbers have a higher priority. Priorities between interrupt types are such that the comparison B interrupt has priority over the comparison A interrupt. An interrupt request to the CPU is accepted only when no other interrupt request of a higher priority has been generated.

It is only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the timer interrupt level set by the interrupt priority register, that a timer interrupt request is actually accepted by the CPU. For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to "ITC (Interrupt Controller)".

### Intelligent DMA

The interrupt factor of each timer can also invoke intelligent DMA (IDMA). This allows memory-to-memory DMA transfers to be performed cyclically.

The following shows the IDMA channel numbers set for each interrupt factor of timer:

IDMA Ch.	IDMA Ch.
Timer 0 comparison B: 0x07	Timer 0 comparison A: 0x08
Timer 1 comparison B: 0x09	Timer 1 comparison A: 0x0A
Timer 2 comparison B: 0x0B	Timer 2 comparison A: 0x0C
Timer 3 comparison B: 0x0D	Timer 3 comparison A: 0x0E
Timer 4 comparison B: 0x0F	Timer 4 comparison A: 0x10
Timer 5 comparison B: 0x11	Timer 5 comparison A: 0x12

### III-5 PERIPHERAL BLOCK: 16-BIT PROGRAMMABLE TIMERS

For IDMA to be invoked, the IDMA request and IDMA enable bits shown in Table III.5.5 must be set to "1" in advance. Transfer conditions, etc. must also be set on the IDMA side in advance.

Table III.5.5 Control Bits for IDMA Transfer

Interrupt factor	IDMA request bit	IDMA enable bit
Timer 0 comparison A	R16TC0(D7/0x40290)	DE16TC0(D7/0x40294)
Timer 0 comparison B	R16TU0(D6/0x40290)	DE16TU0(D6/0x40294)
Timer 1 comparison A	R16TC1(D1/0x40291)	DE16TC1(D1/0x40295)
Timer 1 comparison B	R16TU1(D0/0x40291)	DE16TU1(D0/0x40295)
Timer 2 comparison A	R16TC2(D3/0x40291)	DE16TC2(D3/0x40295)
Timer 2 comparison B	R16TU2(D2/0x40291)	DE16TU2(D2/0x40295)
Timer 3 comparison A	R16TC3(D5/0x40291)	DE16TC3(D5/0x40295)
Timer 3 comparison B	R16TU3(D4/0x40291)	DE16TU3(D4/0x40295)
Timer 4 comparison A	R16TC4(D7/0x40291)	DE16TC4(D7/0x40295)
Timer 4 comparison B	R16TU4(D6/0x40291)	DE16TU4(D6/0x40295)
Timer 5 comparison A	R16TC5(D1/0x40292)	DE16TC5(D1/0x40296)
Timer 5 comparison B	R16TU5(D0/0x40292)	DE16TU5(D0/0x40296)

If the IDMA request and enable bits are set to "1", IDMA is invoked through generation of an interrupt factor. No interrupt request is generated at that point. An interrupt request is generated after the DMA transfer is completed. The registers can also be set so as not to generate an interrupt, with only a DMA transfer performed. For details on IDMA transfers and interrupt control upon completion of IDMA transfer, refer to "IDMA (Intelligent DMA)".

### High-speed DMA

The interrupt factor of each timer can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit corresponding to each timer:

Table III.5.6 HSDMA Trigger Set-up Bits

Interrupt factor	HSDMA channel	Trigger set-up bits
Timer 0 comparison A	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "0111"
Timer 0 comparison B	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "0110"
Timer 1 comparison A	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "0111"
Timer 1 comparison B	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "0110"
Timer 2 comparison A	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "0111"
Timer 2 comparison B	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "0110"
Timer 3 comparison A	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "0111"
Timer 3 comparison B	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "0110"
Timer 4 comparison A	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "1001"
Timer 4 comparison B	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "1001"
	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "1000"
Timer 5 comparison A	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "1000"
	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "1001"
Timer 5 comparison B	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "1001"
	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "1000"
Timer 5 comparison B	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "1000"
	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "1000"

For HSDMA to be invoked, a 16-bit timer interrupt factor should be selected using the trigger set-up bits in advance. Transfer conditions, etc. must also be set on the HSDMA side.

If a 16-bit timer is selected as the HSDMA trigger, the HSDMA channel is invoked through generation of the interrupt factor.

For details on HSDMA transfer, refer to "HSDMA (High-Speed DMA)".

## Trap vectors

The trap vector addresses for each default interrupt factor are set as shown below:

Timer 0 comparison B: 0x0C00078

Timer 0 comparison A: 0x0C0007C

Timer 1 comparison B: 0x0C00088

Timer 1 comparison A: 0x0C0008C

Timer 2 comparison B: 0x0C00098

Timer 2 comparison A: 0x0C0009C

Timer 3 comparison B: 0x0C000A8

Timer 3 comparison A: 0x0C000AC

Timer 4 comparison B: 0x0C000B8

Timer 4 comparison A: 0x0C000BC

Timer 5 comparison B: 0x0C000C8

Timer 5 comparison A: 0x0C000CC

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

## I/O Memory of 16-Bit Programmable Timers

Table III.5.7 shows the control bits of the 16-bit programmable timers.

For details on the I/O memory of the prescaler used to set a clock, refer to "Prescaler".

Table III.5.7 Control Bits of 16-Bit Programmable Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
16-bit timer 0/1 interrupt priority register	0040266 (B)	D7	–	reserved	–	–	–	0 when being read.				
		D6	P16T12	16-bit timer 1 interrupt level	0 to 7	X	R/W					
		D5	P16T11			X						
		D4	P16T10			X						
		D3	–			reserved	–		–	0 when being read.		
		D2	P16T02	16-bit timer 0 interrupt level	0 to 7	X	R/W					
		D1	P16T01			X						
		D0	P16T00			X						
16-bit timer 2/3 interrupt priority register	0040267 (B)	D7	–	reserved	–	–	–	0 when being read.				
		D6	P16T32	16-bit timer 3 interrupt level	0 to 7	X	R/W					
		D5	P16T31			X						
		D4	P16T30			X						
		D3	–			reserved	–		–	0 when being read.		
		D2	P16T22	16-bit timer 2 interrupt level	0 to 7	X	R/W					
		D1	P16T21			X						
		D0	P16T20			X						
16-bit timer 4/5 interrupt priority register	0040268 (B)	D7	–	reserved	–	–	–	0 when being read.				
		D6	P16T52	16-bit timer 5 interrupt level	0 to 7	X	R/W					
		D5	P16T51			X						
		D4	P16T50			X						
		D3	–			reserved	–		–	0 when being read.		
		D2	P16T42	16-bit timer 4 interrupt level	0 to 7	X	R/W					
		D1	P16T41			X						
		D0	P16T40			X						
16-bit timer 0/1 interrupt enable register	0040272 (B)	D7	E16TC1	16-bit timer 1 comparison A	1	Enabled	0	Disabled	0	R/W		
		D6	E16TU1	16-bit timer 1 comparison B					0	R/W		
		D5–4	–	reserved					–	–		0 when being read.
		D3	E16TC0	16-bit timer 0 comparison A	1	Enabled	0	Disabled	0	R/W		
		D2	E16TU0	16-bit timer 0 comparison B					0	R/W		
		D1–0	–	reserved					–	–		0 when being read.
16-bit timer 2/3 interrupt enable register	0040273 (B)	D7	E16TC3	16-bit timer 3 comparison A	1	Enabled	0	Disabled	0	R/W		
		D6	E16TU3	16-bit timer 3 comparison B					0	R/W		
		D5–4	–	reserved					–	–		0 when being read.
		D3	E16TC2	16-bit timer 2 comparison A	1	Enabled	0	Disabled	0	R/W		
		D2	E16TU2	16-bit timer 2 comparison B					0	R/W		
		D1–0	–	reserved					–	–		0 when being read.
16-bit timer 4/5 interrupt enable register	0040274 (B)	D7	E16TC5	16-bit timer 5 comparison A	1	Enabled	0	Disabled	0	R/W		
		D6	E16TU5	16-bit timer 5 comparison B					0	R/W		
		D5–4	–	reserved					–	–		0 when being read.
		D3	E16TC4	16-bit timer 4 comparison A	1	Enabled	0	Disabled	0	R/W		
		D2	E16TU4	16-bit timer 4 comparison B					0	R/W		
		D1–0	–	reserved					–	–		0 when being read.
16-bit timer 0/1 interrupt factor flag register	0040282 (B)	D7	F16TC1	16-bit timer 1 comparison A	1	Factor is generated	0	No factor is generated	X	R/W		
		D6	F16TU1	16-bit timer 1 comparison B					X	R/W		
		D5–4	–	reserved					–	–		0 when being read.
		D3	F16TC0	16-bit timer 0 comparison A	1	Factor is generated	0	No factor is generated	X	R/W		
		D2	F16TU0	16-bit timer 0 comparison B					X	R/W		
		D1–0	–	reserved					–	–		0 when being read.
16-bit timer 2/3 interrupt factor flag register	0040283 (B)	D7	F16TC3	16-bit timer 3 comparison A	1	Factor is generated	0	No factor is generated	X	R/W		
		D6	F16TU3	16-bit timer 3 comparison B					X	R/W		
		D5–4	–	reserved					–	–		0 when being read.
		D3	F16TC2	16-bit timer 2 comparison A	1	Factor is generated	0	No factor is generated	X	R/W		
		D2	F16TU2	16-bit timer 2 comparison B					X	R/W		
		D1–0	–	reserved					–	–		0 when being read.
16-bit timer 4/5 interrupt factor flag register	0040284 (B)	D7	F16TC5	16-bit timer 5 comparison A	1	Factor is generated	0	No factor is generated	X	R/W		
		D6	F16TU5	16-bit timer 5 comparison B					X	R/W		
		D5–4	–	reserved					–	–		0 when being read.
		D3	F16TC4	16-bit timer 4 comparison A	1	Factor is generated	0	No factor is generated	X	R/W		
		D2	F16TU4	16-bit timer 4 comparison B					X	R/W		
		D1–0	–	reserved					–	–		0 when being read.



Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Port input 0–3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA request register	0040290 (B)	D7	R16TC0	16-bit timer 0 comparison A	1	IDMA request	0	Interrupt request	0	R/W	
		D6	R16TU0	16-bit timer 0 comparison B					0	R/W	
		D5	RHDM1	High-speed DMA Ch.1					0	R/W	
		D4	RHDM0	High-speed DMA Ch.0					0	R/W	
		D3	RP3	Port input 3					0	R/W	
		D2	RP2	Port input 2					0	R/W	
		D1	RP1	Port input 1					0	R/W	
		D0	RP0	Port input 0					0	R/W	
16-bit timer 1–4 IDMA request register	0040291 (B)	D7	R16TC4	16-bit timer 4 comparison A	1	IDMA request	0	Interrupt request	0	R/W	
		D6	R16TU4	16-bit timer 4 comparison B					0	R/W	
		D5	R16TC3	16-bit timer 3 comparison A					0	R/W	
		D4	R16TU3	16-bit timer 3 comparison B					0	R/W	
		D3	R16TC2	16-bit timer 2 comparison A					0	R/W	
		D2	R16TU2	16-bit timer 2 comparison B					0	R/W	
		D1	R16TC1	16-bit timer 1 comparison A					0	R/W	
		D0	R16TU1	16-bit timer 1 comparison B					0	R/W	
16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA request register	0040292 (B)	D7	RSTX0	SIF Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	0	R/W	
		D6	RSRX0	SIF Ch.0 receive buffer full					0	R/W	
		D5	R8TU3	8-bit timer 3 underflow					0	R/W	
		D4	R8TU2	8-bit timer 2 underflow					0	R/W	
		D3	R8TU1	8-bit timer 1 underflow					0	R/W	
		D2	R8TU0	8-bit timer 0 underflow					0	R/W	
		D1	R16TC5	16-bit timer 5 comparison A					0	R/W	
		D0	R16TU5	16-bit timer 5 comparison B					0	R/W	
Port input 0–3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA enable register	0040294 (B)	D7	DE16TC0	16-bit timer 0 comparison A	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DE16TU0	16-bit timer 0 comparison B					0	R/W	
		D5	DEHDM1	High-speed DMA Ch.1					0	R/W	
		D4	DEHDM0	High-speed DMA Ch.0					0	R/W	
		D3	DEP3	Port input 3					0	R/W	
		D2	DEP2	Port input 2					0	R/W	
		D1	DEP1	Port input 1					0	R/W	
		D0	DEP0	Port input 0					0	R/W	
16-bit timer 1–4 IDMA enable register	0040295 (B)	D7	DE16TC4	16-bit timer 4 comparison A	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DE16TU4	16-bit timer 4 comparison B					0	R/W	
		D5	DE16TC3	16-bit timer 3 comparison A					0	R/W	
		D4	DE16TU3	16-bit timer 3 comparison B					0	R/W	
		D3	DE16TC2	16-bit timer 2 comparison A					0	R/W	
		D2	DE16TU2	16-bit timer 2 comparison B					0	R/W	
		D1	DE16TC1	16-bit timer 1 comparison A					0	R/W	
		D0	DE16TU1	16-bit timer 1 comparison B					0	R/W	
16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA enable register	0040296 (B)	D7	DESTX0	SIF Ch.0 transmit buffer empty	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DESRX0	SIF Ch.0 receive buffer full					0	R/W	
		D5	DE8TU3	8-bit timer 3 underflow					0	R/W	
		D4	DE8TU2	8-bit timer 2 underflow					0	R/W	
		D3	DE8TU1	8-bit timer 1 underflow					0	R/W	
		D2	DE8TU0	8-bit timer 0 underflow					0	R/W	
		D1	DE16TC5	16-bit timer 5 comparison A					0	R/W	
		D0	DE16TU5	16-bit timer 5 comparison B					0	R/W	
P1 function select register	00402D4 (B)	D7	–	reserved		–	–	–	–	0 when being read.	
		D6	CFP16	P16 function selection	1	EXCL5 #DMAEND1	0	P16	0	R/W	Extended functions (0x402D7)
		D5	CFP15	P15 function selection	1	EXCL4 #DMAEND0	0	P15	0	R/W	
		D4	CFP14	P14 function selection	1	FOSC1	0	P14	0	R/W	Extended functions (0x402DF)
		D3	CFP13	P13 function selection	1	EXCL3 T8UF3	0	P13	0	R/W	
		D2	CFP12	P12 function selection	1	EXCL2 T8UF2	0	P12	0	R/W	
		D1	CFP11	P11 function selection	1	EXCL1 T8UF1	0	P11	0	R/W	
D0	CFP10	P10 function selection	1	EXCL0 T8UF0	0	P10	0	R/W			
P1 I/O control register	00402D6 (B)	D7	–	reserved		–	–	–	–	0 when being read.	
		D6	IOC16	P16 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D5	IOC15	P15 I/O control					0	R/W	
		D4	IOC14	P14 I/O control					0	R/W	
		D3	IOC13	P13 I/O control					0	R/W	
		D2	IOC12	P12 I/O control					0	R/W	
		D1	IOC11	P11 I/O control					0	R/W	
D0	IOC10	P10 I/O control	0	R/W							

### III-5 PERIPHERAL BLOCK: 16-BIT PROGRAMMABLE TIMERS

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
P2 function select register	00402D8 (B)	D7	CFP27	P27 function selection	1	TM5	0	P27	0	R/W	Ext. func.(0x402DB)
		D6	CFP26	P26 function selection	1	TM4	0	P26	0	R/W	
		D5	CFP25	P25 function selection	1	TM3	0	P25	0	R/W	
		D4	CFP24	P24 function selection	1	TM2	0	P24	0	R/W	
		D3	CFP23	P23 function selection	1	TM1	0	P23	0	R/W	
		D2	CFP22	P22 function selection	1	TM0	0	P22	0	R/W	
		D1	CFP21	P21 function selection	1	#DWE	0	P21	0	R/W	Ext. func.(0x402DF)
		D0	CFP20	P20 function selection	1	#DRD	0	P20	0	R/W	
Port function extension register	00402DF (B)	D7	CFEX7	P07 port extended function	1	#DMAEND3	0	P07, etc.	0	R/W	
		D6	CFEX6	P06 port extended function	1	#DMAACK3	0	P06, etc.	0	R/W	
		D5	CFEX5	P05 port extended function	1	#DMAEND2	0	P05, etc.	0	R/W	
		D4	CFEX4	P04 port extended function	1	#DMAACK2	0	P04, etc.	0	R/W	
		D3	CFEX3	P34 port extended function	1	#GARD	0	P31, etc.	0	R/W	
		D2	CFEX2	P21 port extended function	1	#GAAS	0	P21, etc.	0	R/W	
		D1	CFEX1	P10, P11, P13 port extended function	1	DST0 DST1 DPCO	0	P10, etc. P11, etc. P13, etc.	1	R/W	
		D0	CFEX0	P12, P14 port extended function	1	DST2 DCLK	0	P12, etc. P14, etc.	1	R/W	
16-bit timer 0 comparison data A set-up register	0048180 (HW)	DF	CR0A15	16-bit timer 0 comparison data A CR0A15 = MSB CR0A0 = LSB	0 to 65535			X	R/W		
		DE	CR0A14		X						
		DD	CR0A13		X						
		DC	CR0A12		X						
		DB	CR0A11		X						
		DA	CR0A10		X						
		D9	CR0A9		X						
		D8	CR0A8		X						
		D7	CR0A7		X						
		D6	CR0A6		X						
		D5	CR0A5		X						
		D4	CR0A4		X						
		D3	CR0A3		X						
		D2	CR0A2		X						
		D1	CR0A1		X						
		D0	CR0A0		X						
16-bit timer 0 comparison data B set-up register	0048182 (HW)	DF	CR0B15	16-bit timer 0 comparison data B CR0B15 = MSB CR0B0 = LSB	0 to 65535			X	R/W		
		DE	CR0B14		X						
		DD	CR0B13		X						
		DC	CR0B12		X						
		DB	CR0B11		X						
		DA	CR0B10		X						
		D9	CR0B9		X						
		D8	CR0B8		X						
		D7	CR0B7		X						
		D6	CR0B6		X						
		D5	CR0B5		X						
		D4	CR0B4		X						
		D3	CR0B3		X						
		D2	CR0B2		X						
		D1	CR0B1		X						
		D0	CR0B0		X						
16-bit timer 0 counter data register	0048184 (HW)	DF	TC015	16-bit timer 0 counter data TC015 = MSB TC00 = LSB	0 to 65535			X	R		
		DE	TC014		X						
		DD	TC013		X						
		DC	TC012		X						
		DB	TC011		X						
		DA	TC010		X						
		D9	TC09		X						
		D8	TC08		X						
		D7	TC07		X						
		D6	TC06		X						
		D5	TC05		X						
		D4	TC04		X						
		D3	TC03		X						
		D2	TC02		X						
		D1	TC01		X						
		D0	TC00		X						
16-bit timer 0 control register	0048186 (B)	D7	–	reserved	–			0	–	0 when being read.	
		D6	SELF0	16-bit timer 0 fine mode selection	1	Fine mode	0	Normal	0	R/W	
		D5	SELCRB0	16-bit timer 0 comparison buffer	1	Enabled	0	Disabled	0	R/W	
		D4	OUTINV0	16-bit timer 0 output inversion	1	Invert	0	Normal	0	R/W	
		D3	CKSL0	16-bit timer 0 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	PTM0	16-bit timer 0 clock output control	1	On	0	Off	0	R/W	
		D1	PRESET0	16-bit timer 0 reset	1	Reset	0	Invalid	0	W	0 when being read.
		D0	PRUN0	16-bit timer 0 Run/Stop control	1	Run	0	Stop	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 1 comparison data A set-up register	0048188 (HW)	DF	CR1A15	16-bit timer 1 comparison data A CR1A15 = MSB CR1A0 = LSB	0 to 65535	X	R/W		
		DE	CR1A14						
		DD	CR1A13						
		DC	CR1A12						
		DB	CR1A11						
		DA	CR1A10						
		D9	CR1A9						
		D8	CR1A8						
		D7	CR1A7						
		D6	CR1A6						
		D5	CR1A5						
		D4	CR1A4						
		D3	CR1A3						
		D2	CR1A2						
		D1	CR1A1						
		D0	CR1A0						
16-bit timer 1 comparison data B set-up register	004818A (HW)	DF	CR1B15	16-bit timer 1 comparison data B CR1B15 = MSB CR1B0 = LSB	0 to 65535	X	R/W		
		DE	CR1B14						
		DD	CR1B13						
		DC	CR1B12						
		DB	CR1B11						
		DA	CR1B10						
		D9	CR1B9						
		D8	CR1B8						
		D7	CR1B7						
		D6	CR1B6						
		D5	CR1B5						
		D4	CR1B4						
		D3	CR1B3						
		D2	CR1B2						
		D1	CR1B1						
		D0	CR1B0						
16-bit timer 1 counter data register	004818C (HW)	DF	TC115	16-bit timer 1 counter data TC115 = MSB TC10 = LSB	0 to 65535	X	R		
		DE	TC114						
		DD	TC113						
		DC	TC112						
		DB	TC111						
		DA	TC110						
		D9	TC19						
		D8	TC18						
		D7	TC17						
		D6	TC16						
		D5	TC15						
		D4	TC14						
		D3	TC13						
		D2	TC12						
		D1	TC11						
		D0	TC10						
16-bit timer 1 control register	004818E (B)	D7	–	reserved	–	0	–	0 when being read.	
		D6	SELFM1	16-bit timer 1 fine mode selection	1 Fine mode	0 Normal	0	R/W	
		D5	SELCRB1	16-bit timer 1 comparison buffer	1 Enabled	0 Disabled	0	R/W	
		D4	OUTINV1	16-bit timer 1 output inversion	1 Invert	0 Normal	0	R/W	
		D3	CKSL1	16-bit timer 1 input clock selection	1 External clock	0 Internal clock	0	R/W	
		D2	PTM1	16-bit timer 1 clock output control	1 On	0 Off	0	R/W	
		D1	PRESET1	16-bit timer 1 reset	1 Reset	0 Invalid	0	W	0 when being read.
		D0	PRUN1	16-bit timer 1 Run/Stop control	1 Run	0 Stop	0	R/W	
16-bit timer 2 comparison data A set-up register	0048190 (HW)	DF	CR2A15	16-bit timer 2 comparison data A CR2A15 = MSB CR2A0 = LSB	0 to 65535	X	R/W		
		DE	CR2A14						
		DD	CR2A13						
		DC	CR2A12						
		DB	CR2A11						
		DA	CR2A10						
		D9	CR2A9						
		D8	CR2A8						
		D7	CR2A7						
		D6	CR2A6						
		D5	CR2A5						
		D4	CR2A4						
		D3	CR2A3						
		D2	CR2A2						
		D1	CR2A1						
		D0	CR2A0						

### III-5 PERIPHERAL BLOCK: 16-BIT PROGRAMMABLE TIMERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 2 comparison data B set-up register	0048192 (HW)	DF	CR2B15	16-bit timer 2 comparison data B CR2B15 = MSB CR2B0 = LSB	0 to 65535	X	R/W		
		DE	CR2B14						
		DD	CR2B13						
		DC	CR2B12						
		DB	CR2B11						
		DA	CR2B10						
		D9	CR2B9						
		D8	CR2B8						
		D7	CR2B7						
		D6	CR2B6						
		D5	CR2B5						
		D4	CR2B4						
		D3	CR2B3						
		D2	CR2B2						
		D1	CR2B1						
		D0	CR2B0						
		16-bit timer 2 counter data register	0048194 (HW)			DF			TC215
DE	TC214								
DD	TC213								
DC	TC212								
DB	TC211								
DA	TC210								
D9	TC29								
D8	TC28								
D7	TC27								
D6	TC26								
D5	TC25								
D4	TC24								
D3	TC23								
D2	TC22								
D1	TC21								
D0	TC20								
16-bit timer 2 control register	0048196 (B)			D7	–	reserved	–	0	–
		D6	SELF2	16-bit timer 2 fine mode selection	1 Fine mode	0 Normal	0	R/W	
		D5	SELCRB2	16-bit timer 2 comparison buffer	1 Enabled	0 Disabled	0	R/W	
		D4	OUTINV2	16-bit timer 2 output inversion	1 Invert	0 Normal	0	R/W	
		D3	CKSL2	16-bit timer 2 input clock selection	1 External clock	0 Internal clock	0	R/W	
		D2	PTM2	16-bit timer 2 clock output control	1 On	0 Off	0	R/W	
		D1	PRESET2	16-bit timer 2 reset	1 Reset	0 Invalid	0	W	0 when being read.
		D0	PRUN2	16-bit timer 2 Run/Stop control	1 Run	0 Stop	0	R/W	
16-bit timer 3 comparison data A set-up register	0048198 (HW)	DF	CR3A15	16-bit timer 3 comparison data A CR3A15 = MSB CR3A0 = LSB	0 to 65535	X	R/W		
		DE	CR3A14						
		DD	CR3A13						
		DC	CR3A12						
		DB	CR3A11						
		DA	CR3A10						
		D9	CR3A9						
		D8	CR3A8						
		D7	CR3A7						
		D6	CR3A6						
		D5	CR3A5						
		D4	CR3A4						
		D3	CR3A3						
		D2	CR3A2						
		D1	CR3A1						
		D0	CR3A0						
		16-bit timer 3 comparison data B set-up register	004819A (HW)			DF			CR3B15
DE	CR3B14								
DD	CR3B13								
DC	CR3B12								
DB	CR3B11								
DA	CR3B10								
D9	CR3B9								
D8	CR3B8								
D7	CR3B7								
D6	CR3B6								
D5	CR3B5								
D4	CR3B4								
D3	CR3B3								
D2	CR3B2								
D1	CR3B1								
D0	CR3B0								

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 3 counter data register	004819C (HW)	DF	TC315	16-bit timer 3 counter data	0 to 65535	X	R		
		DE	TC314	TC315 = MSB					
		DD	TC313	TC30 = LSB					
		DC	TC312						
		DB	TC311						
		DA	TC310						
		D9	TC39						
		D8	TC38						
		D7	TC37						
		D6	TC36						
		D5	TC35						
		D4	TC34						
		D3	TC33						
		D2	TC32						
		D1	TC31						
		D0	TC30						
16-bit timer 3 control register	004819E (B)	D7	–	reserved	–	0	–	0 when being read.	
		D6	SELFM3	16-bit timer 3 fine mode selection	1 Fine mode	0 Normal	0	R/W	
		D5	SELCRB3	16-bit timer 3 comparison buffer	1 Enabled	0 Disabled	0	R/W	
		D4	OUTINV3	16-bit timer 3 output inversion	1 Invert	0 Normal	0	R/W	
		D3	CKSL3	16-bit timer 3 input clock selection	1 External clock	0 Internal clock	0	R/W	
		D2	PTM3	16-bit timer 3 clock output control	1 On	0 Off	0	R/W	
		D1	PRESET3	16-bit timer 3 reset	1 Reset	0 Invalid	0	W	0 when being read.
		D0	PRUN3	16-bit timer 3 Run/Stop control	1 Run	0 Stop	0	R/W	
16-bit timer 4 comparison data A set-up register	00481A0 (HW)	DF	CR4A15	16-bit timer 4 comparison data A	0 to 65535	X	R/W		
		DE	CR4A14	CR4A15 = MSB					
		DD	CR4A13	CR4A0 = LSB					
		DC	CR4A12						
		DB	CR4A11						
		DA	CR4A10						
		D9	CR4A9						
		D8	CR4A8						
		D7	CR4A7						
		D6	CR4A6						
		D5	CR4A5						
		D4	CR4A4						
		D3	CR4A3						
		D2	CR4A2						
		D1	CR4A1						
		D0	CR4A0						
16-bit timer 4 comparison data B set-up register	00481A2 (HW)	DF	CR4B15	16-bit timer 4 comparison data B	0 to 65535	X	R/W		
		DE	CR4B14	CR4B15 = MSB					
		DD	CR4B13	CR4B0 = LSB					
		DC	CR4B12						
		DB	CR4B11						
		DA	CR4B10						
		D9	CR4B9						
		D8	CR4B8						
		D7	CR4B7						
		D6	CR4B6						
		D5	CR4B5						
		D4	CR4B4						
		D3	CR4B3						
		D2	CR4B2						
		D1	CR4B1						
		D0	CR4B0						
16-bit timer 4 counter data register	00481A4 (HW)	DF	TC415	16-bit timer 4 counter data	0 to 65535	X	R		
		DE	TC414	TC415 = MSB					
		DD	TC413	TC40 = LSB					
		DC	TC412						
		DB	TC411						
		DA	TC410						
		D9	TC49						
		D8	TC48						
		D7	TC47						
		D6	TC46						
		D5	TC45						
		D4	TC44						
		D3	TC43						
		D2	TC42						
		D1	TC41						
		D0	TC40						

### III-5 PERIPHERAL BLOCK: 16-BIT PROGRAMMABLE TIMERS

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 4 control register	00481A6 (B)	D7	–	reserved	–			0	–	0 when being read.	
		D6	<b>SELFM4</b>	16-bit timer 4 fine mode selection	1	Fine mode	0	Normal	0	R/W	
		D5	<b>SELCRB4</b>	16-bit timer 4 comparison buffer	1	Enabled	0	Disabled	0	R/W	
		D4	<b>OUTINV4</b>	16-bit timer 4 output inversion	1	Invert	0	Normal	0	R/W	
		D3	<b>CKSL4</b>	16-bit timer 4 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	<b>PTM4</b>	16-bit timer 4 clock output control	1	On	0	Off	0	R/W	
		D1	<b>PRESET4</b>	16-bit timer 4 reset	1	Reset	0	Invalid	0	W	0 when being read.
D0	<b>PRUN4</b>	16-bit timer 4 Run/Stop control	1	Run	0	Stop	0	R/W			
16-bit timer 5 comparison data A set-up register	00481A8 (HW)	DF	<b>CR5A15</b>	16-bit timer 5 comparison data A	0 to 65535			X	R/W		
		DE	<b>CR5A14</b>	CR5A15 = MSB				X			
		DD	<b>CR5A13</b>	CR5A0 = LSB				X			
		DC	<b>CR5A12</b>					X			
		DB	<b>CR5A11</b>					X			
		DA	<b>CR5A10</b>					X			
		D9	<b>CR5A9</b>					X			
		D8	<b>CR5A8</b>					X			
		D7	<b>CR5A7</b>					X			
		D6	<b>CR5A6</b>					X			
		D5	<b>CR5A5</b>					X			
		D4	<b>CR5A4</b>					X			
		D3	<b>CR5A3</b>					X			
		D2	<b>CR5A2</b>					X			
		D1	<b>CR5A1</b>					X			
D0	<b>CR5A0</b>					X					
16-bit timer 5 comparison data B set-up register	00481AA (HW)	DF	<b>CR5B15</b>	16-bit timer 5 comparison data B	0 to 65535			X	R/W		
		DE	<b>CR5B14</b>	CR5B15 = MSB				X			
		DD	<b>CR5B13</b>	CR5B0 = LSB				X			
		DC	<b>CR5B12</b>					X			
		DB	<b>CR5B11</b>					X			
		DA	<b>CR5B10</b>					X			
		D9	<b>CR5B9</b>					X			
		D8	<b>CR5B8</b>					X			
		D7	<b>CR5B7</b>					X			
		D6	<b>CR5B6</b>					X			
		D5	<b>CR5B5</b>					X			
		D4	<b>CR5B4</b>					X			
		D3	<b>CR5B3</b>					X			
		D2	<b>CR5B2</b>					X			
		D1	<b>CR5B1</b>					X			
D0	<b>CR5B0</b>					X					
16-bit timer 5 counter data register	00481AC (HW)	DF	<b>TC515</b>	16-bit timer 5 counter data	0 to 65535			X	R		
		DE	<b>TC514</b>	TC515 = MSB				X			
		DD	<b>TC513</b>	TC50 = LSB				X			
		DC	<b>TC512</b>					X			
		DB	<b>TC511</b>					X			
		DA	<b>TC510</b>					X			
		D9	<b>TC59</b>					X			
		D8	<b>TC58</b>					X			
		D7	<b>TC57</b>					X			
		D6	<b>TC56</b>					X			
		D5	<b>TC55</b>					X			
		D4	<b>TC54</b>					X			
		D3	<b>TC53</b>					X			
		D2	<b>TC52</b>					X			
		D1	<b>TC51</b>					X			
D0	<b>TC50</b>					X					
16-bit timer 5 control register	00481AE (B)	D7	–	reserved	–			0	–	0 when being read.	
		D6	<b>SELFM5</b>	16-bit timer 5 fine mode selection	1	Fine mode	0	Normal	0	R/W	
		D5	<b>SELCRB5</b>	16-bit timer 5 comparison buffer	1	Enabled	0	Disabled	0	R/W	
		D4	<b>OUTINV5</b>	16-bit timer 5 output inversion	1	Invert	0	Normal	0	R/W	
		D3	<b>CKSL5</b>	16-bit timer 5 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	<b>PTM5</b>	16-bit timer 5 clock output control	1	On	0	Off	0	R/W	
		D1	<b>PRESET5</b>	16-bit timer 5 reset	1	Reset	0	Invalid	0	W	0 when being read.
D0	<b>PRUN5</b>	16-bit timer 5 Run/Stop control	1	Run	0	Stop	0	R/W			

**CFP16–CFP10: P1[6:0] pin function selection (D[6:0]) / P1 function select register (0x402D4)**

Selects the pin to be used for input of an external count clock to the timer.

Write "1": Clock input pin

Write "0": I/O port pin

Read: Valid

Select clock input pins for the timers that are used as an event counter from among P10 through P16, by writing "1" to CFP10–CFP16. For the relationship between each pin and timer, refer to Table III.5.1. The pin is set for an I/O port by writing "0" to CFP1x.

In addition to pin selection here, the pin to be used for clock input to the 16-bit programmable timer must be set to input mode using the I/O control register.

At cold start, CFP1x is set to "0" (I/O port). At hot start, CFP1x retains its status from prior to the initial reset.

**CFP27–CFP22: P2[7:2] pin function selection (D[7:2]) / P2 function select register (0x402D8)**

Selects the pin used for clock output.

Write "1": Clock output pin

Write "0": I/O port pin

Read: Valid

Select the pin to be used to output a timer-generated clock to external devices from among P22 through P27, by writing "1" to CFP22–CFP27. For the relationship between each pin and timer, refer to Table III.5.1. The pin is set for an I/O port by writing "0" to CFP2x.

At cold start, CFP2x is set to "0" (I/O port). At hot start, CFP2x retains its status from prior to the initial reset.

**CFEX1: P10, P11, P13 port extended function (D1) / Port function extension register (0x402DF)****CFEX0: P12, P14 port extended function (D0) / Port function extension register (0x402DF)**

Sets whether the function of an I/O-port pin is to be extended.

Write "1": Function-extended pin

Write "0": I/O-port/peripheral-circuit pin

Read: Valid

When CFEX[1:0] is set to "1", the P14–P10 ports function as debug signal output ports. When CFEX[1:0] = "0", the CFP1[4:0] bit becomes effective, so the settings of these bits determine whether the P14–P10 ports function as I/O ports or external clock input ports.

At cold start, CFEX[1:0] is set to "1" (function-extended pins). At hot start, CFEX[1:0] retains its state from prior to the initial reset.

**IOC16–IOC10: P1[6:0] port I/O control (D[6:0]) / P1 I/O control register (0x402D6)**

Directs P10 through P16 for input or output and indicates the I/O control signal value of the port.

**When writing data**

Write "1": Output mode

Write "0": Input mode

For the pin selected from among P10 through P16 for use for external clock input, write "0" to the corresponding I/O control bit to set it to input mode. If the pin is set to output mode, even though its CFP1x may be set to "1", it functions as the output pin of an 8-bit programmable timer and cannot be used to receive an external clock.

**When reading data**

Read "1": I/O control signal (output)

Read "0": I/O control signal (input)

The I/O control signal value for the port pin is read from this register. When I/O port function is selected using the CFEX and CFP1x registers, the value written to the IOC register is read out as is. When peripheral function is selected, the read value depends on the peripheral circuit status and may not indicate the value written to the IOC register.

At cold start, IOC1x is set to "0" (input mode). At hot start, the bit retains its state from prior to the initial reset.

---

**SELF0:** Timer 0 fine mode selection (D6) / 16-bit timer 0 control register (0x48186)  
**SELF1:** Timer 1 fine mode selection (D6) / 16-bit timer 1 control register (0x4818E)  
**SELF2:** Timer 2 fine mode selection (D6) / 16-bit timer 2 control register (0x48196)  
**SELF3:** Timer 3 fine mode selection (D6) / 16-bit timer 3 control register (0x4819E)  
**SELF4:** Timer 4 fine mode selection (D6) / 16-bit timer 4 control register (0x481A6)  
**SELF5:** Timer 5 fine mode selection (D6) / 16-bit timer 5 control register (0x481AE)

---

Sets fine mode for clock output.

Write "1": Fine mode  
 Write "0": Normal output  
 Read: Valid

When SELFMx is set to "1", clock output is set in fine mode which allows adjustment of the output signal duty ratio in units of a half cycle for the input clock.

When SELFMx is set to "0", normal clock output will be performed.

At initial reset, SELFMx is set to "0" (normal output).

---

**SELCRB0:** Timer 0 comparison register buffer enable (D5) / 16-bit timer 0 control register (0x48186)  
**SELCRB1:** Timer 1 comparison register buffer enable (D5) / 16-bit timer 1 control register (0x4818E)  
**SELCRB2:** Timer 2 comparison register buffer enable (D5) / 16-bit timer 2 control register (0x48196)  
**SELCRB3:** Timer 3 comparison register buffer enable (D5) / 16-bit timer 3 control register (0x4819E)  
**SELCRB4:** Timer 4 comparison register buffer enable (D5) / 16-bit timer 4 control register (0x481A6)  
**SELCRB5:** Timer 5 comparison register buffer enable (D5) / 16-bit timer 5 control register (0x481AE)

---

Enables or disables writing to the comparison register buffer.

Write "1": Enabled  
 Write "0": Disabled  
 Read: Valid

When SELCRBx is set to "1", comparison data is read and written from/to the comparison register buffer. The content of the buffer is loaded to the comparison data register when the counter is reset by the software or the comparison B signal.

When SELCRBx is set to "0", comparison data is read and written from/to the comparison data register.

At initial reset, SELCRBx is set to "0" (disabled).

---

**OUTINV0:** Timer 0 output inversion (D4) / 16-bit timer 0 control register (0x48186)  
**OUTINV1:** Timer 1 output inversion (D4) / 16-bit timer 1 control register (0x4818E)  
**OUTINV2:** Timer 2 output inversion (D4) / 16-bit timer 2 control register (0x48196)  
**OUTINV3:** Timer 3 output inversion (D4) / 16-bit timer 3 control register (0x4819E)  
**OUTINV4:** Timer 4 output inversion (D4) / 16-bit timer 4 control register (0x481A6)  
**OUTINV5:** Timer 5 output inversion (D4) / 16-bit timer 5 control register (0x481AE)

---

Selects a logic of the output signal.

Write "1": Inverted (active low)  
 Write "0": Normal (active high)  
 Read: Valid

By writing "1" to OUTINVx, an active-low signal (off level = high) is generated for the TMx output. When OUTINVx is set to "0", an active-high signal (off level = low) is generated.

At initial reset, OUTINVx is set to "0" (active high).



---

**CKSL0:** Timer 0 input clock selection (D3) / 16-bit timer 0 control register (0x48186)  
**CKSL1:** Timer 1 input clock selection (D3) / 16-bit timer 1 control register (0x4818E)  
**CKSL2:** Timer 2 input clock selection (D3) / 16-bit timer 2 control register (0x48196)  
**CKSL3:** Timer 3 input clock selection (D3) / 16-bit timer 3 control register (0x4819E)  
**CKSL4:** Timer 4 input clock selection (D3) / 16-bit timer 4 control register (0x481A6)  
**CKSL5:** Timer 5 input clock selection (D3) / 16-bit timer 5 control register (0x481AE)

---

Selects the input clock of each timer.

Write "1": External clock  
 Write "0": Internal clock  
 Read: Valid

The internal clock (prescaler output) is selected for the input clock of each timer by writing "0" to CKSLx. An external clock (one that is fed from the clock input pin) is selected by writing "1", and the timer functions as an event counter. In this case, the clock input pin must be set using CFP1x before an external clock is selected here. At initial reset, CKSLx is set to "0" (internal clock).

---

**PTM0:** Timer 0 clock output control (D2) / 16-bit timer 0 control register (0x48186)  
**PTM1:** Timer 1 clock output control (D2) / 16-bit timer 1 control register (0x4818E)  
**PTM2:** Timer 2 clock output control (D2) / 16-bit timer 2 control register (0x48196)  
**PTM3:** Timer 3 clock output control (D2) / 16-bit timer 3 control register (0x4819E)  
**PTM4:** Timer 4 clock output control (D2) / 16-bit timer 4 control register (0x481A6)  
**PTM5:** Timer 5 clock output control (D2) / 16-bit timer 5 control register (0x481AE)

---

Controls the output of the TMx signal (timer output clock).

Write "1": On  
 Write "0": Off  
 Read: Valid

The TMx signal is output from the clock output pin by writing "1" to PTMx. Clock output is stopped by writing "0" to PTMx and goes to the off level according to the OUTINVx setting (low when OUTINVx = "0" or high when OUTINVx = "1"). In this case, the clock output pin must be set using CFP2x before outputting the TMx signal here. At initial reset, PTMx is set to "0" (off).

---

**PRESET0:** Timer 0 reset (D1) / 16-bit timer 0 control register (0x48186)  
**PRESET1:** Timer 1 reset (D1) / 16-bit timer 1 control register (0x4818E)  
**PRESET2:** Timer 2 reset (D1) / 16-bit timer 2 control register (0x48196)  
**PRESET3:** Timer 3 reset (D1) / 16-bit timer 3 control register (0x4819E)  
**PRESET4:** Timer 4 reset (D1) / 16-bit timer 4 control register (0x481A6)  
**PRESET5:** Timer 5 reset (D1) / 16-bit timer 5 control register (0x481AE)

---

Resets the counter.

Write "1": Reset  
 Write "0": Invalid  
 Read: Always "0"

The counter of timer x is reset by writing "1" to PRESETx.

Writing "0" results in No Operation.

Since PRESETx is a write-only bit, its content when read is always "0".

---

**PRUN0:** Timer 0 RUN/STOP control (D0) / 16-bit timer 0 control register (0x48186)  
**PRUN1:** Timer 1 RUN/STOP control (D0) / 16-bit timer 1 control register (0x4818E)  
**PRUN2:** Timer 2 RUN/STOP control (D0) / 16-bit timer 2 control register (0x48196)  
**PRUN3:** Timer 3 RUN/STOP control (D0) / 16-bit timer 3 control register (0x4819E)  
**PRUN4:** Timer 4 RUN/STOP control (D0) / 16-bit timer 4 control register (0x481A6)  
**PRUN5:** Timer 5 RUN/STOP control (D0) / 16-bit timer 5 control register (0x481AE)

---

Controls the timer's RUN/STOP state.

Write "1": RUN  
 Write "0": STOP  
 Read: Valid

Each timer is made to start counting up by writing "1" to PRUNx and made to stop counting by writing "0". In the STOP state, the counter data is retained until the timer is reset or placed in a RUN state. By changing states from STOP to RUN, the timer can restart counting beginning at the retained count. At initial reset, PRUNx is set to "0" (STOP).

---

**CR0A15–CR0A0:** Timer 0 comparison data A (D[F:0]) / 16-bit timer 0 comparison data A set-up register (0x48180)  
**CR1A15–CR1A0:** Timer 1 comparison data A (D[F:0]) / 16-bit timer 1 comparison data A set-up register (0x48188)  
**CR2A15–CR2A0:** Timer 2 comparison data A (D[F:0]) / 16-bit timer 2 comparison data A set-up register (0x48190)  
**CR3A15–CR3A0:** Timer 3 comparison data A (D[F:0]) / 16-bit timer 3 comparison data A set-up register (0x48198)  
**CR4A15–CR4A0:** Timer 4 comparison data A (D[F:0]) / 16-bit timer 4 comparison data A set-up register (0x481A0)  
**CR5A15–CR5A0:** Timer 5 comparison data A (D[F:0]) / 16-bit timer 5 comparison data A set-up register (0x481A8)

---

Sets the comparison data A of each timer.

When SELCRBx is set to "0", comparison data is directly read or writing from/to the comparison data register A. When SELCRBx is set to "1", comparison data is read or written from/to the comparison register buffer A. The content of the buffer is loaded to the comparison data register A when the counter is reset. The data set in this register is compared with each corresponding counter data. When the contents match, a comparison A interrupt is generated and the output signal rises (OUTINVx = "0") or falls (OUTINVx = "1"). This does not affect the counter value and count-up operation. At initial reset, CRxA is not initialized.

---

**CR0B15–CR0B0:** Timer 0 comparison data B (D[F:0]) / 16-bit timer 0 comparison data B set-up register (0x48182)  
**CR1B15–CR1B0:** Timer 1 comparison data B (D[F:0]) / 16-bit timer 1 comparison data B set-up register (0x4818A)  
**CR2B15–CR2B0:** Timer 2 comparison data B (D[F:0]) / 16-bit timer 2 comparison data B set-up register (0x48192)  
**CR3B15–CR3B0:** Timer 3 comparison data B (D[F:0]) / 16-bit timer 3 comparison data B set-up register (0x4819A)  
**CR4B15–CR4B0:** Timer 4 comparison data B (D[F:0]) / 16-bit timer 4 comparison data B set-up register (0x481A2)  
**CR5B15–CR5B0:** Timer 5 comparison data B (D[F:0]) / 16-bit timer 5 comparison data B set-up register (0x481AA)

---

Sets the comparison data B of each timer.

When SELCRBx is set to "0", comparison data is directly read or writing from/to the comparison data register B. When SELCRBx is set to "1", comparison data is read or written from/to the comparison register buffer B. The content of the buffer is loaded to the comparison data register B when the counter is reset. The data set in this register is compared with each corresponding counter data. When the contents match, a comparison B interrupt is generated and the output signal falls (OUTINVx = "0") or rises (OUTINVx = "1"). Furthermore, the counter is reset to "0". At initial reset, CRxB is not initialized.

---

**TC015–TC00:** Timer 0 counter data (D[F:0]) / 16-bit timer 0 counter data register (0x48184)

**TC115–TC10:** Timer 1 counter data (D[F:0]) / 16-bit timer 1 counter data register (0x4818C)

**TC215–TC20:** Timer 2 counter data (D[F:0]) / 16-bit timer 2 counter data register (0x48194)

**TC315–TC30:** Timer 3 counter data (D[F:0]) / 16-bit timer 3 counter data register (0x4819C)

**TC415–TC40:** Timer 4 counter data (D[F:0]) / 16-bit timer 4 counter data register (0x481A4)

**TC515–TC50:** Timer 5 counter data (D[F:0]) / 16-bit timer 5 counter data register (0x481AC)

---

The counter data of each timer can be read from this register.

The data can be read out at any time.

Since TCx is a read-only register, writing to this register is ignored.

At initial reset, TCx is not initialized.

---

**P16T02–P16T00:** Timer 0 interrupt level (D[2:0]) / 16-bit timer 0/1 interrupt priority register (0x40266)

**P16T12–P16T10:** Timer 1 interrupt level (D[6:4]) / 16-bit timer 0/1 interrupt priority register (0x40266)

**P16T22–P16T20:** Timer 2 interrupt level (D[2:0]) / 16-bit timer 2/3 interrupt priority register (0x40267)

**P16T32–P16T30:** Timer 3 interrupt level (D[6:4]) / 16-bit timer 2/3 interrupt priority register (0x40267)

**P16T42–P16T40:** Timer 4 interrupt level (D[2:0]) / 16-bit timer 4/5 interrupt priority register (0x40268)

**P16T52–P16T50:** Timer 5 interrupt level (D[6:4]) / 16-bit timer 4/5 interrupt priority register (0x40268)

---

Sets the priority levels of 16-bit programmable timer interrupts.

The priority level can be set in the range of 0 to 7.

At initial reset, P16Tx becomes indeterminate.

---

**E16TU0, E16TC0:** Timer 0 interrupt enable (D2, D3) / 16-bit timer 0/1 interrupt enable register (0x40272)

**E16TU1, E16TC1:** Timer 1 interrupt enable (D6, D7) / 16-bit timer 0/1 interrupt enable register (0x40272)

**E16TU2, E16TC2:** Timer 2 interrupt enable (D2, D3) / 16-bit timer 2/3 interrupt enable register (0x40273)

**E16TU3, E16TC3:** Timer 3 interrupt enable (D6, D7) / 16-bit timer 2/3 interrupt enable register (0x40273)

**E16TU4, E16TC4:** Timer 4 interrupt enable (D2, D3) / 16-bit timer 4/5 interrupt enable register (0x40274)

**E16TU5, E16TC5:** Timer 5 interrupt enable (D6, D7) / 16-bit timer 4/5 interrupt enable register (0x40274)

---

Enables or disables the generation of an interrupt to the CPU.

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

The E16TUx and E16TCx are provided for the comparison B and comparison A interrupt factors, respectively. The interrupt for which the bit is set to "1" is enabled, and the interrupt for which the bit is set to "0" is disabled.

At initial reset, these bits are set to "0" (interrupt disabled).

---

**F16TU0, F16TC0:** Timer 0 interrupt factor flag (D2, D3) / 16-bit timer 0/1 interrupt factor flag register (0x40282)

**F16TU1, F16TC1:** Timer 1 interrupt factor flag (D6, D7) / 16-bit timer 0/1 interrupt factor flag register (0x40282)

**F16TU2, F16TC2:** Timer 2 interrupt factor flag (D2, D3) / 16-bit timer 2/3 interrupt factor flag register (0x40283)

**F16TU3, F16TC3:** Timer 3 interrupt factor flag (D6, D7) / 16-bit timer 2/3 interrupt factor flag register (0x40283)

**F16TU4, F16TC4:** Timer 4 interrupt factor flag (D2, D3) / 16-bit timer 4/5 interrupt factor flag register (0x40284)

**F16TU5, F16TC5:** Timer 5 interrupt factor flag (D6, D7) / 16-bit timer 4/5 interrupt factor flag register (0x40284)

---

Indicates the status of 16-bit programmable timer interrupt generation.

#### When read

Read "1": Interrupt factor has occurred

Read "0": No interrupt factor has occurred

#### When written using the reset-only method (default)

Write "1": Interrupt factor flag is reset

Write "0": Invalid

#### When written using the read/write method

Write "1": Interrupt flag is set

Write "0": Interrupt flag is reset

### III-5 PERIPHERAL BLOCK: 16-BIT PROGRAMMABLE TIMERS

F16TUx and F16TCx are the interrupt factor flags corresponding to the comparison B and comparison A interrupts, respectively. The flag is set to "1" when each interrupt factor occurs.

At this time, if the following conditions are met, an interrupt to the CPU is generated:

1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The PSR's IE bit is set to "1" (interrupts enabled).
4. The value set in the corresponding interrupt priority register is higher than the CPU's interrupt level (IL).

When using the interrupt factor of the 16-bit programmable timer to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of how the interrupt enable and interrupt priority registers are set.

If the next interrupt is to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).

The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.

At initial reset, all these flags become indeterminate, so be sure to reset them in the software.

---

**R16TU0, R16TC0:** Timer 0 IDMA request

(D6, D7) / Port input 0–3, HSDMA, 16-bit timer 0 IDMA request register (0x40290)

**R16TU1, R16TC1:** Timer 1 IDMA request (D0, D1) / 16-bit timer 1–4 IDMA request register (0x40291)

**R16TU2, R16TC2:** Timer 2 IDMA request (D2, D3) / 16-bit timer 1–4 IDMA request register (0x40291)

**R16TU3, R16TC3:** Timer 3 IDMA request (D4, D5) / 16-bit timer 1–4 IDMA request register (0x40291)

**R16TU4, R16TC4:** Timer 4 IDMA request (D6, D7) / 16-bit timer 1–4 IDMA request register (0x40291)

**R16TU5, R16TC5:** Timer 5 IDMA request

(D0, D1) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA request register (0x40292)

---

Specifies whether to invoke IDMA when an interrupt factor occurs.

#### When using the set-only method (default)

Write "1": IDMA request

Write "0": Not changed

Read: Valid

#### When using the read/write method

Write "1": IDMA request

Write "0": Interrupt request

Read: Valid

R16TUx and R16TCx are IDMA request bits corresponding to the comparison B and comparison A interrupt factors, respectively. When the bit is set to "1", IDMA is invoked when the interrupt factor occurs, thereby performing programmed data transfers. When the register is set to "0", normal interrupt processing is performed and IDMA is not invoked. For details on IDMA, refer to "IDMA (Intelligent DMA)".

At initial reset, these bits are set to "0" (interrupt request).

---

**DE16TU0, DE16TC0:** Timer 0 IDMA enable

(D6, D7) / Port input 0–3, HSDMA, 16-bit timer 0 IDMA enable register (0x40294)

**DE16TU1, DE16TC1:** Timer 1 IDMA enable (D0, D1) / 16-bit timer 1–4 IDMA enable register (0x40295)

**DE16TU2, DE16TC2:** Timer 2 IDMA enable (D2, D3) / 16-bit timer 1–4 IDMA enable register (0x40295)

**DE16TU3, DE16TC3:** Timer 3 IDMA enable (D4, D5) / 16-bit timer 1–4 IDMA enable register (0x40295)

**DE16TU4, DE16TC4:** Timer 4 IDMA enable (D6, D7) / 16-bit timer 1–4 IDMA enable register (0x40295)

**DE16TU5, DE16TC5:** Timer 5 IDMA enable

(D0, D1) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA enable register (0x40296)

---

Enables IDMA transfer by means of an interrupt factor.

**When using the set-only method (default)**

Write "1": IDMA enabled

Write "0": Not changed

Read: Valid

**When using the read/write method**

Write "1": IDMA enabled

Write "0": IDMA disabled

Read: Valid

DE16TUx and DE16TCx are IDMA enable bits corresponding to the comparison B and comparison A interrupt factors, respectively. If the bit is set to "1", the IDMA request by the interrupt factor is enabled. If the bit is set to "0", the IDMA request is disabled.

After an initial reset, these bits are set to "0" (IDMA disabled).

## Programming Notes

---

- (1) The 16-bit programmable timers operate only when the prescaler is operating.
- (2) When setting the input clock or operation mode, make sure the 16-bit programmable timer is turned off.
- (3) If a same value is set to the comparison data A and B registers, a hazard may be generated in the output signal. Therefore, do not set the comparison registers as  $A = B$ .  
There is no problem when the interrupt function only is used.
- (4) When using the output clock, set the comparison data registers as  $A \geq 0$  and  $B \geq 1$ . The minimum settings are  $A = 0$  and  $B = 1$ . In this case, the timer output clock cycle is the input clock  $\times 1/2$ .
- (5) When the comparison data registers are set as  $A > B$  in normal mode, no comparison A interrupt is generated. In this case, the output signal is fixed at the off level.  
In fine mode, no comparison A interrupt is generated when the comparison data registers are set as  $A > 2 \times B + 1$ .
- (6) After an initial reset, the interrupt factor flag becomes indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in the software.
- (7) To prevent another interrupt from being generated by the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.
- (8) Be aware that unnecessary pulse may be generated according to the control of the clock output and port configuration when a 16-bit programmable timer is used to output the TMx clock.  
For example, when TMx is set as inverted output ( $OUTINVx = "1"$ ), the output waveform falls with the comparison B signal and it rises with the comparison A signal. Furthermore, the output pin is fixed at high level when PTMx is set to "0" to stop the clock output. When switching the output pin to the I/O port pin and then setting the port to low after the TMx signal falls with the comparison A signal, a high level pulse will be generated if "0" is written to PTMx before setting the port to low. It can be prevented by writing "0" to PTMx after setting the port to low.

## III-6 WATCHDOG TIMER

### Configuration of Watchdog Timer

The Peripheral Block incorporates a watchdog timer function to detect the CPU's crash.

This function is implemented through the use of the 16-bit programmable timer 0. When this function is enabled, an NMI (nonmaskable interrupt) is generated by the comparison B signal from the 16-bit programmable timer 0 (generating intervals can be set through the use of software). The 16-bit programmable timer 0 set in the software so as not to generate the NMI, making it possible to detect a program crash that may not pass through this processing routine.

Figure III.6.1 shows the block diagram of the watchdog timer.

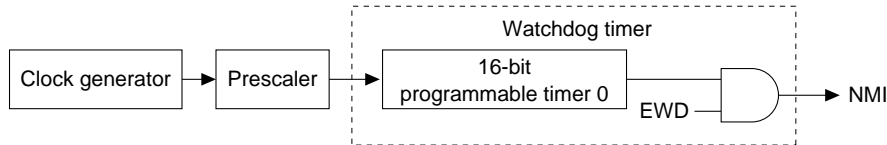


Figure III.6.1 Watchdog Timer Block Diagram

### Control of Watchdog Timer

#### Setting the operating clock and NMI generating interval

The watchdog timer is operated by the prescaler's output clock. Therefore, the watchdog timer function cannot be used when the prescaler is inactive.

The NMI is generated every time the 16-bit programmable timer 0 is reset by the comparison B setting. Therefore, this interval is determined by the prescaler's P16TS0[2:0] (D[2:0]) / 16-bit timer 0 clock control register (0x40147), and the comparison data B set in CR0B[15:0] (D[F:0]) / 16-bit timer 0 comparison register B (0x48182).

The NMI generating interval is calculated using the following equation:

$$\text{NMI generating interval} = \frac{\text{CR0B} + 1}{\text{fPSCIN} \times \text{pdr}} \quad [\text{sec.}]$$

fPSCIN: Prescaler input clock frequency [Hz]

pdr: Prescaler's division ratio set by the P16TS0 register (1/4096, 1/1024, 1/256, 1/64, 1/16, 1/4, 1/2, 1/1)

CR0B: Set value of the CR0B register (0 to 65,535)

For details on how to control the prescaler and the 16-bit programmable timer 0, refer to "Prescaler" and "16-Bit Programmable Timers".

#### Setting the watchdog timer function

To use the watchdog timer function, enable the NMI that is generated by the comparison B signal from the 16-bit programmable timer 0. For this purpose, use EWD (D1) / Watchdog timer enable register (0x40171).

The NMI is enabled by writing "1" to EWD. At initial reset, EWD is set to "0", so generation of the NMI is disabled.

To prevent an unwanted NMI from being generated by erroneous writing to EWD, this register is normally write-protected. To write-enable EWD, write "1" to WRWD (D7) / Watchdog timer write-protect register (0x40170). Only one writing to EWD is enabled in this way by the WRWD bit. When data is written to EWD after it is write-enabled, the WRWD bit is reset back to "0", thus making EWD write-protected again.

For the 16-bit programmable timer 0, set an appropriate comparison B value to make it start operating.

If the watchdog timer function is not to be used, set EWD to "0" and do not change it.

## Resetting the watchdog timer

When using the watchdog timer, prepare a routine to reset the 16-bit programmable timer 0 before an NMI is generated in a location where it will be periodically processed. Make sure this routine is processed within the NMI generation interval described above.

The 16-bit programmable timer 0 is reset by writing "1" to PRESET0 (D1) / 16-bit timer 0 control register (0x48186). At this point, the timer counter is set to 0, and the timer starts counting the NMI generation interval over again from that point.

If the watchdog timer is not reset within the set interval for any reason, the CPU is made to enter trap processing by an NMI and starts executing the processing routine indicated by the NMI vector.

The NMI trap vector address is set to 0x0C0001C by default.

The trap table base address can be changed using the TTBR registers (0x48134 to 0x48137).

## Operation in Standby Modes

---

### During HALT mode

In HALT mode (basic mode or HALT2 mode), the prescaler and watchdog timer are operating. Consequently, if HALT mode continues beyond the NMI generation interval, HALT mode is cleared by the NMI.

To disable the watchdog timer in HALT mode, set EWD to "0" before executing the halt instruction or turn off the 16-bit programmable timer 0.

If the NMI is disabled by EWD, the 16-bit programmable timer 0 continues counting even in HALT mode. To reenble the NMI after clearing HALT mode, reset the 16-bit programmable timer 0 in advance.

If HALT mode was entered after the 16-bit programmable timer 0 was turned off, reset the timer before restarting it.

### During SLEEP mode

In SLEEP mode, the prescaler is turned off. Therefore, the watchdog timer also stops operating. To prevent generation of an unwanted NMI after clearing SLEEP mode, reset the 16-bit programmable timer 0 before executing the slp instruction. In addition, disable generation of the NMI by EWD as necessary.



## I/O Memory of Watchdog Timer

Table III.6.1 shows the control bits of the watchdog timer.

Table III.6.1 Control Bits of Watchdog Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog timer write-protect register	0040170 (B)	D7	WRWD	EWD write protection	1   Write enabled   0   Write-protect	0	R/W	
		D6-0	–	–	–	–	–	0 when being read.
Watchdog timer enable register	0040171 (B)	D7-2	–	–	–	–	–	0 when being read.
		D1	EWD	Watchdog timer enable	1   NMI enabled   0   NMI disabled	0	R/W	
		D0	–	–	–	–	–	0 when being read.

### WRWD: EWD write protection (D7) / Watchdog timer write-protect register (0x40170)

Enables writing to the EWD register.

Write "1": Writing enabled

Write "0": Write-protected

Read: Valid

The EWD bit is write-protected to prevent unwanted modifications. Writing to this bit is enabled for only one writing by setting WRWD to "1". WRWD is reset back to "0" by writing to EWD, so EWD is write-protected again. If WRWD is reset to "0" when EWD is write-enabled (WRWD = "1"), EWD becomes write-protected again. At initial reset, WRWD is set to "0" (write-protected).

### EWD: NMI enable (D1) / Watchdog timer enable register (0x40171)

Controls the generation of a nonmaskable interrupt (NMI) by the watchdog timer.

Write "1": NMI is enabled

Write "0": NMI is disabled

Read: Valid

The watchdog timer's interrupt signal is masked by writing "0" to EWD, so a nonmaskable interrupt (NMI) to the CPU is not generated. If EWD is set to "1", an NMI is generated by the 16-bit programmable timer 0 comparison B signal.

Writing to EWD is valid only when WRWD = "1".

Even when EWD is set to "0", the 16-bit programmable timer 0 does not stop counting. Therefore, if the NMI has been temporarily disabled, be sure to reset the 16-bit programmable timer 0 before setting the EWD register back to "1".

At initial reset, EWD is set to "0" (NMI disabled).

## Programming Notes

- (1) If the watchdog timer's NMI is enabled, the watchdog timer must be reset in the software before the 16-bit programmable timer 0 outputs the comparison B signal.
- (2) Even when EWD is set to "0", the 16-bit programmable timer 0 does not stop counting. Therefore, if the NMI has been temporarily disabled, be sure to reset the 16-bit programmable timer 0 before setting EWD back to "1".

THIS PAGE IS BLANK.

## III-7 LOW-SPEED (OSC1) OSCILLATION CIRCUIT

### Configuration of Low-Speed (OSC1) Oscillation Circuit

The Peripheral Block has a built-in low-speed (OSC1) oscillation circuit.

The low-speed (OSC1) oscillation circuit generates a 32.768-kHz (Typ.) subclock.

The OSC1 clock output by this circuit is delivered to the CLG (clock generator) in the Core Block and is used as the source clock for the clock timer. It can also be used as a sub-clock for the low-speed (low-power) operation of the CPU and peripheral circuits (switchable in a program).

Figure III.7.1 shows the configuration of the clock system.

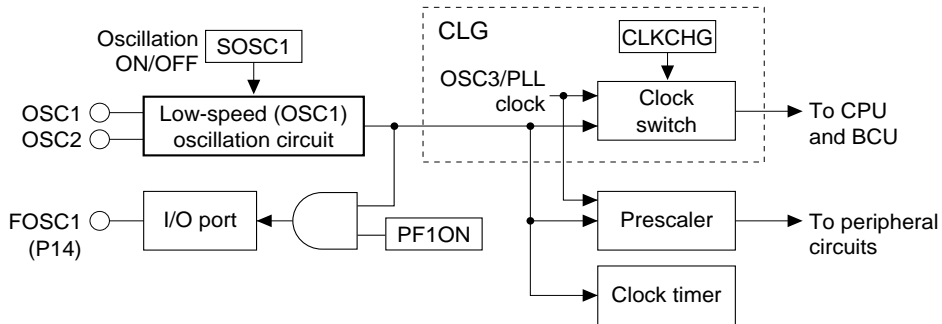


Figure III.7.1 Configuration of Clock System

The CPU operating clock can be switched to the output (OSC1 clock) of the low-speed (OSC1) oscillation circuit in a program. Furthermore, the oscillation circuit can be stopped in a program.

If the OSC3 clock is unnecessary such as when performing clock processing only, set the OSC1 clock for operation of the CPU/peripheral circuits and turn off the high-speed (OSC3) oscillation circuit in order to reduce current consumption.

The low-speed (OSC1) oscillation circuit does not stop in SLEEP mode.

For the control method when using the OSC1 clock for the operating clock of the peripheral circuits, refer to "Prescaler".

### I/O Pins of Low-Speed (OSC1) Oscillation Circuit

Table III.7.1 lists the I/O pins of the low-speed (OSC1) oscillation circuit.

Table III.7.1 I/O Pins of Low-Speed (OSC1) Oscillation Circuit

Pin name	I/O	Function
OSC1	I	Low-speed (OSC1) oscillation input pin Crystal oscillation or external clock input
OSC2	O	Low-speed (OSC1) oscillation output pin Crystal oscillation (open when external clock is used)
DCLK(P14/FOSC1)	I/O	DCLK signal output / I/O port / Low-speed (OSC1) oscillation clock output

## Oscillator Types

In the low-speed (OSC1) oscillation circuit, either a crystal oscillation or an external clock input can be selected as the type of oscillation circuit.

Figure III.7.2 shows the structure of the low-speed (OSC1) oscillation circuit.

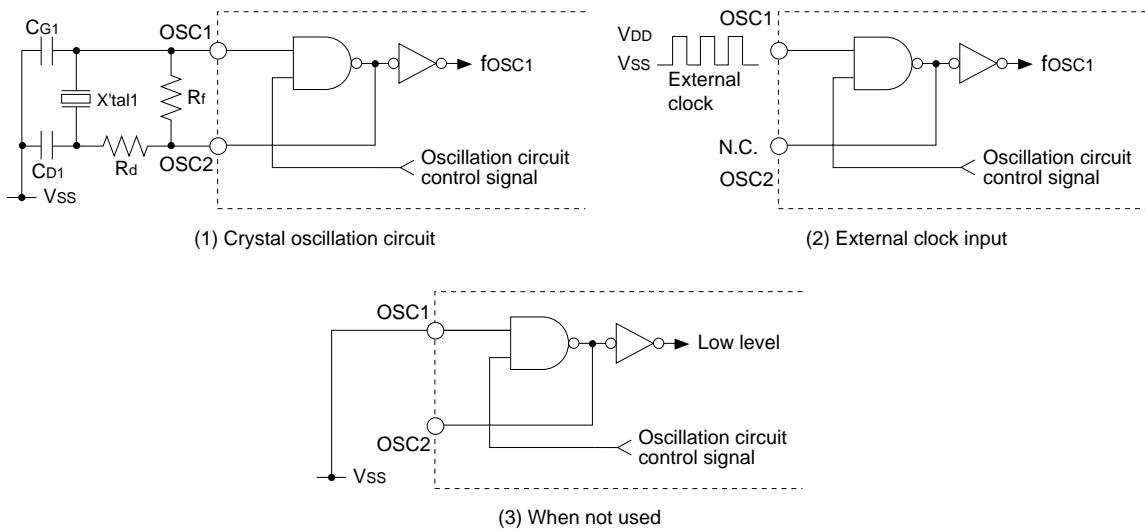


Figure III.7.2 Low-Speed (OSC1) Oscillation Circuit

When using a crystal oscillation for this circuit, connect a crystal resonator X'tal1 (32.768 kHz, Typ.), a feedback resistor (Rf), two capacitors (CG1, CD1), and a drain resistor (Rd), if necessary, to the OSC1 and OSC2 pins and VSS as shown in Figure III.7.2 (1).

When an external clock source is used, leave the OSC2 pin open and input a square-wave clock to the OSC1 pin. If the low-speed (OSC1) oscillation circuit is not used, connect the OSC1 pin to VSS and leave the OSC2 pin open.

The oscillation frequency is 32.768 kHz (Typ.). Use a crystal resonator or external clock that oscillates at this frequency. No other frequency can be used for clock applications.

For details on oscillation characteristics and the external clock input characteristics, refer to "Electrical Characteristics".

## Controlling Oscillation

The low-speed (OSC1) oscillation circuit can be turned on or off using SOSC1 (D0) / Power control register (0x40180).

The oscillation circuit is turned off by writing "0" to SOSC1 and turned back on again by writing "1". SOSC1 is set to "1" at initial reset, so the oscillation circuit is turned on.

- Notes:**
- When the low-speed (OSC1) oscillation circuit is used as the clock source for the CPU operating clock, it cannot be turned off. In this case, writing "0" to SOSC1 is ignored. Note also that writing to SOSC1 is allowed only when the power-control register protection flag is set to "0b10010110".
  - Immediately after the oscillation circuit is turned on, a certain period of time is required for oscillation to stabilize (3 sec max.). To prevent the device from operating erratically, do not use the clock until its oscillation has stabilized.

The low-speed (OSC1) oscillation circuit does not stop when the CPU is set in SLEEP mode.

## Switching Over the CPU Operating Clock

After an initial reset, the CPU starts operating using the OSC3 clock.

In cases in which some peripheral circuits (e.g., programmable timer, serial interface, A/D converter, and ports) do not need to be operate or processing in low-speed operation is possible, and the CPU can process its jobs at a low clock speed, the CPU operating clock can be switched to the OSC1 clock, thereby reducing current consumption. Use CLKCHG (D2) / Power control register (0x40180) to switch over the operating clock.

### Procedure for switching over from the OSC3 clock to the OSC1 clock

1. Turn on the low-speed (OSC1) oscillation circuit (by writing "1" to SOSC1).
  2. Wait until the OSC1 oscillation stabilizes.
  3. Change the CPU operating clock (by writing "0" to CLKCHG).
  4. Turn off the high-speed (OSC3) oscillation circuit (by writing "0" to SOSC3).
- \* Steps 1 and 2 are required only when the low-speed (OSC1) oscillation circuit is inactive.

- Notes:**
- Use separate instructions to switch from OSC3 to OSC1 and turn the OSC3 oscillation off. If these operations are processed simultaneously using one instruction, the CPU may operate erratically.
  - Make sure the operation of the peripheral circuits, such as the programmable timer and serial interface is terminated before the OSC3 oscillation is turned off in order to prevent them from operating erratically or the prescaler clock is set as OSC1. In addition, in order to prevent incorrect operation, a setup of prescaler must be performed before changing the CPU clock.

### Procedure for switching over from the OSC1 clock to the OSC3 clock

1. Turn on the high-speed (OSC3) oscillation circuit (by writing "1" to SOSC3).
2. Wait until the OSC3 oscillation stabilizes.
3. Switch over the CPU operating clock (by writing "1" to CLKCHG).

**Note:** The operating clock switchover by CLKCHG is effective only when both oscillation circuits are on and the power-control register protection flag is set to "0b10010110".

Refer to "Electrical Characteristics" for the oscillation stabilization times.

## Power-Control Register Protection Flag

The power-control register (SOSC1, SOSC3, CLKCHG, CLKDT[1:0]) at address 0x40180, which is used to control the oscillation circuits and the CPU operating clock, is normally disabled against writing in order to prevent it from malfunctioning due to unnecessary writing.

To enable this register for writing, the power-control register protection flag CLGP[7:0] (D[7:0]) / Power-control protection register (0x4019E) must be set to "0b10010110". Note that this setting allows for the power-control register (0x40180) to be written to only once, so all bits of CLGP[7:0] are cleared to "0" when this address is written to. Therefore, CLGP[7:0] must be set to "0b10010110" each time the power-control register (0x40180) is written to. The flag CLGP[7:0] does not affect the readout from the power-control register (0x40180).

## Operation in Standby Mode

In HALT mode, which is entered by executing the halt instruction, the low-speed (OSC1) oscillation circuits retains its status before HALT mode is entered. Under normal conditions, therefore, there is no need to control the oscillation circuit before entering or after exiting HALT mode.

The low-speed (OSC1) oscillation circuit does not stop operating in SLEEP mode set by executing the slp (sleep) instruction. Therefore, if the CPU was operating using the OSC1 clock before SLEEP mode was entered, the CPU keeps operating using the OSC1 clock in SLEEP mode.

## OSC1 Clock Output to External Devices

The low-speed (OSC1) oscillation clock can be output from the FOSC1 (P14) pin to external devices.

Table III.7.2 OSC1 Clock Output Pin

Pin name	I/O	Function	Function select bit
DCLK(P14/FOSC1)	I/O	DCLK signal output / I/O port /	CFP14 (D4) / P1 function select register (0x402D4)
		Low-speed (OSC1) oscillation clock output	CFEX0 (D0) / Port function extension register (0x402DF)

### Setting the clock output pin

The pin used to output the OSC1 clock to external devices is shared with the P14 I/O port and the debug clock signal DCLK.

At cold start, it is set for the DCLK signal output (CFP14 = "0" and CFEX0 = "1"). When using the FOSC1 clock output function, write "1" to CFP14 and "0" to CFEX0 (refer to "I/O Ports"), and also write "1" to IOC14 (0x402D6/D4).

At hot start, the pin retains its pre-reset status.

### Output control

To start clock output, write "1" to PF1ON (D0) / Clock option register (0x40190). The clock output is stopped by writing "0".

At initial reset, PF1ON is set to "0" (output disabled).

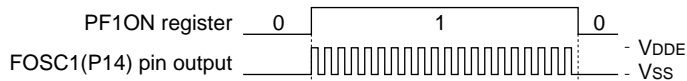


Figure III.7.3 OSC1 Clock Output

## I/O Memory of Low-Speed (OSC1) Oscillation Circuit

Table III.7.3 lists the control bits of the low-speed (OSC1) oscillation circuit.

Table III.7.3 Control Bits of Low-Speed (OSC1) Oscillation Circuit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Power control register	0040180 (B)	D7	CLKDT1	System clock division ratio selection	CLKDT[1:0]		0	R/W			
		D6	CLKDT0		1	1				1/8	
					1	0				1/4	
					0	1				1/2	
					0	0				1/1	
		D5	PSCON	Prescaler On/Off control	1	On	0	Off	1	R/W	
		D4-3	–	reserved					0	–	Writing 1 not allowed.
D2	CLKCHG	CPU operating clock switch	1	OSC3	0	OSC1	1	R/W			
D1	SOSC3	High-speed (OSC3) oscillation On/Off	1	On	0	Off	1	R/W			
D0	SOSC1	Low-speed (OSC1) oscillation On/Off	1	On	0	Off	1	R/W			
Clock option register	0040190 (B)	D7-4	–	–		–	–	–	0 when being read.		
		D3	HLT2OP	HALT clock option	1	On	0	Off	0	R/W	
		D2	8T1ON	OSC3-stabilize waiting function	1	Off	0	On	1	R/W	
		D1	–	reserved					0	–	Do not write 1.
		D0	PF1ON	OSC1 external output control	1	On	0	Off	0	R/W	
Power control protect register	004019E (B)	D7	CLGP7	Power control register protect flag	Writing 10010110 (0x96) removes the write protection of the power control register (0x40180) and the clock option register (0x40190). Writing another value set the write protection.		0	R/W			
		D6	CLGP6				0				
		D5	CLGP5				0				
		D4	CLGP4				0				
		D3	CLGP3				0				
		D2	CLGP2				0				
		D1	CLGP1				0				
		D0	CLGP0				0				
		P1 function select register	00402D4 (B)				D7			–	reserved
D6	CFP16			P16 function selection	1	EXCL5 #DMAEND1	0	P16	0	R/W	Extended functions (0x402D7)
D5	CFP15			P15 function selection	1	EXCL4 #DMAEND0	0	P15	0	R/W	
D4	CFP14			P14 function selection	1	FOSC1	0	P14	0	R/W	Extended functions (0x402DF)
D3	CFP13			P13 function selection	1	EXCL3 T8UF3	0	P13	0	R/W	
D2	CFP12			P12 function selection	1	EXCL2 T8UF2	0	P12	0	R/W	
D1	CFP11			P11 function selection	1	EXCL1 T8UF1	0	P11	0	R/W	
D0	CFP10			P10 function selection	1	EXCL0 T8UF0	0	P10	0	R/W	
Port function extension register	00402DF (B)	D7	CFEX7	P07 port extended function	1	#DMAEND3	0	P07, etc.	0	R/W	
		D6	CFEX6	P06 port extended function	1	#DMAACK3	0	P06, etc.	0	R/W	
		D5	CFEX5	P05 port extended function	1	#DMAEND2	0	P05, etc.	0	R/W	
		D4	CFEX4	P04 port extended function	1	#DMAACK2	0	P04, etc.	0	R/W	
		D3	CFEX3	P31 port extended function	1	#GARD	0	P31, etc.	0	R/W	
		D2	CFEX2	P21 port extended function	1	#GAAS	0	P21, etc.	0	R/W	
		D1	CFEX1	P10, P11, P13 port extended function	1	DST0 DST1 DPCO	0	P10, etc. P11, etc. P13, etc.	1	R/W	
		D0	CFEX0	P12, P14 port extended function	1	DST2 DCLK	0	P12, etc. P14, etc.	1	R/W	

III

OSC1

### III-7 PERIPHERAL BLOCK: LOW-SPEED (OSC1) OSCILLATION CIRCUIT

---

**SOSC1: Low-speed (OSC1) oscillation control (D0) / Power control register (0x40180)**

---

Turns the low-speed (OSC1) oscillation on or off.

Write "1": OSC1 oscillation turned on  
Write "0": OSC1 oscillation turned off  
Read: Valid

The oscillation of the low-speed (OSC1) oscillation circuit is stopped by writing "0" to SOSC1, and started again by writing "1".

After the oscillation has been restarted, an oscillation stabilization time (see "I-6 Electrical Characteristics") is required before the OSC1 clock can be used.

Writing to SOSC1 is allowed only when CLGP[7:0] is set to "0b10010110". Note also that if the CPU is operating using the OSC1 clock, writing "0" to SOSC1 is ignored and the oscillation is not turned off.

At initial reset, SOSC1 is set to "1" (OSC1 oscillation turned on).

---

**CLKCHG: CPU operating clock switch (D2) / Power control register (0x40180)**

---

Selects the CPU operating clock.

Write "1": OSC3 clock  
Write "0": OSC1 clock  
Read: Valid

The OSC3 clock is selected as the CPU operating clock by writing "1" to CLKCHG, and OSC1 is selected by writing "0". The operating clock can be switched over in this way only when both the high-speed (OSC3) and low-speed (OSC1) oscillation circuits are on. In addition, writing to CLKCHG is effective only when CLGP[7:0] is set to "0b10010110". Immediately after the oscillation circuit has started oscillating, wait for the oscillation to stabilize before switching over the CPU operating clock.

At initial reset, CLKCHG is set to "1" (OSC3 clock).

For controlling the high-speed (OSC3) oscillation circuit, refer to "CLG (Clock Generator)" in the Core Block.

---

**PF10N: OSC1 external output control (D0) / Clock option register (0x40190)**

---

Turns the low-speed (OSC1) clock output to external devices on or off.

Write "1": On  
Write "0": Off  
Read: Valid

The low-speed (OSC1) clock is output from the FOSC1 pin to an external device by writing "1" to PF10N. However, for this setting to be effective, the P14 pin must be set for the FOSC1 pin by CFP14 and CFEX0, and output mode must be set by setting IOC14 (D4/0x402D6 <P1 I/O control register>) to "1".

The clock output is disabled by writing "0".

Writing to PF10N is allowed only when CLGP[7:0] is set to "0b10010110".

At initial reset, PF10N is set to "0" (Off).

---

**CLGP7-CLGP0: Power-control register protection flag ([D7:0]) / Power control protection register (0x4019E)**

---

These bits remove the protection against writing to addresses 0x40180 and 0x40190.

Write "0b10010110": Write protection removed  
Write other than the above: No operation (write-protected)  
Read: Valid

Before writing to address 0x40180 or 0x40190, set CLGP[7:0] to "0b10010110" to remove the protection against writing to that address. This clearing of write protection is effective for only one writing, so the bits are cleared to "0b00000000" by one writing. Therefore, CLGP[7:0] must be set each time the protected address is written to.

At initial reset, CLGP is set to "0b00000000" (write-protected).



**HLT2OP: HALT clock option (D3) / Clock option register (0x40190)**

Select a HALT condition (basic mode or HALT2 mode).

Write "1": HALT2 mode

Write "0": Basic mode

Read: Valid

When "1" is written to HLT2OP, the CPU will enter HALT2 mode when the HALT instruction is executed. When "0" is written, the CPU will enter basic mode.

Writing to HLT2OP is allowed only when CLGP[7:0] is set to "0b10010110".

At initial reset, HLT2OP is set to "0" (basic mode).

The following shows the operating status in HALT mode (basic mode and HALT2 mode) and SLEEP mode.

Table III.7.4 Operating Status in Standby Mode

Standby mode		Operating status	Reactivating factor
HALT mode	Basic mode	<ul style="list-style-type: none"> <li>The CPU clock is stopped. (CPU stop status)</li> <li>BCU clock is supplied. (BCU run status)</li> <li>DMA clock is not stopped. (DMA run status)</li> <li>Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop)</li> <li>The high-speed oscillation circuit maintains the status before entering HALT mode.</li> <li>The low-speed oscillation circuit maintains the status before entering HALT mode.</li> </ul>	<ul style="list-style-type: none"> <li>Reset, NMI</li> <li>Enabled (not masked) interrupt factors</li> </ul>
	HALT2 mode	<ul style="list-style-type: none"> <li>The CPU clock is stopped. (CPU stop status)</li> <li>BCU clock is stopped. (BCU stop status)</li> <li>DMA clock is stopped. (DMA stop status)</li> <li>Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop)</li> <li>The high-speed oscillation circuit maintains the status before entering HALT mode.</li> <li>The low-speed oscillation circuit maintains the status before entering HALT mode.</li> </ul>	A restart is possible only in the case of: <ul style="list-style-type: none"> <li>Reset, NMI</li> <li>Enabled (not masked) interrupt factors</li> </ul> Note, however, that an interrupt from a peripheral circuit can restart the CPU only when the operating clock is supplied to the peripheral circuit.
SLEEP mode		<ul style="list-style-type: none"> <li>The CPU clock is stopped. (CPU stop status)</li> <li>BCU clock is stopped. (BCU stop status)</li> <li>Clocks for the peripheral circuits are stopped.</li> <li>The high-speed oscillation circuit is stopped.</li> <li>The low-speed oscillation circuit maintains the status before entering SLEEP mode.</li> </ul>	<ul style="list-style-type: none"> <li>Reset, NMI</li> <li>Enabled (not masked) input port interrupt factors</li> <li>Clock timer interrupt when the low-speed oscillation circuit is being operated</li> </ul>

**CFP14: P14 function selection (D4) / P1 function select register (0x402D4)**

Selects the pin function of the P14 I/O port.

Write "1": OSC1 clock output pin

Write "0": I/O port pin

Read: Invalid

The P14 pin is set for OSC1 clock output (FOSC1) by writing "1" to CFP14.

When this pin is used as the FOSC1 output pin, also set IOC14 (D4/0x402D6 <P1 I/O control register>) to "1" (output).

At cold start, CFP14 is set to "0" (I/O port pin). At hot start, CFP14 retains its status from before the initial reset.

**CFEX0: P12, P14 extended function (D0) / Port function extension register (0x402DF)**

Sets whether the function of the P14 pin is to be extended.

Write "1": DCLK output pin

Write "0": P14/FOSC1 output pin

Read: Invalid

When CFEX0 is set to "1", the P14 pin functions as a debug clock DCLK output pin. When CFEX0 = "0", the CFP14 register becomes effective, so the settings of this register determine whether the P14 pin functions as an P14 I/O port or a FOSC1 output pin.

At cold start, CFEX0 is set to "1" (DCLK output pin). At hot start, CFEX0 retains its state from prior to the initial reset.

## ***Programming Notes***

---

- (1) Immediately after the low-speed (OSC1) oscillation circuit is turned on, a certain period of time is required for oscillation to stabilize (see "I-6 Electrical Characteristics"). To prevent the device from operating erratically, do not use the clock until its oscillation has stabilized.
- (2) The oscillation circuit used for the CPU operating clock cannot be turned off.
- (3) The CPU operating clock can only be switched over when both the OSC3 and OSC1 oscillation circuits are on. Furthermore, when turning off an oscillation circuit that has become unnecessary as a result of the CPU operating clock switchover, be sure to use separate instructions for switchover and oscillation turnoff. If these two operations are processed simultaneously using one instruction, the CPU may operate erratically.
- (4) If the low-speed (OSC1) oscillation circuit is turned off, all peripheral circuits operated using the OSC1 clock will be inactive.
- (5) If the OSC3 clock is unnecessary, use the OSC1 clock to operate the CPU and turn the high-speed (OSC3) oscillation circuit off. This helps reduce current consumption.
- (6) When the P14/FOSC1/DCLK pin is used as the FOSC1 output pin, set IOC14 (D4/0x402D6) to "1" (output) in addition to the CFP14 (D4/0x402D4) and CFEX0 (D0/0x402DF) settings.

# III-8 CLOCK TIMER

## Configuration of Clock Timer

The clock timer consists of an 8-bit binary counter that is clocked by a 256-Hz signal derived from the low-speed (OSC1) oscillation clock  $f_{osc1}$ , and second, minute, hour, and day counters, allowing all data (128 Hz to 1 Hz, seconds, minutes, hours, and day) to be read out in a software. It can also generate an interrupt using a 32-Hz, 8-Hz, 2-Hz, or 1-Hz (1-second) signal or when a one-minute, one-hour, or one-day count is up, in addition to generating an alarm at a specified time (minute or hour) or day.

The low-speed (OSC1) oscillation circuit and the clock timer can be kept operating even when the CPU and other internal peripheral circuits are placed in standby mode (HALT or SLEEP).

Normally, this clock timer should be used for a clock and various other clocking functions.

Figure III.8.1 shows the structure of the clock timer.

**Note:** Since the clock timer is driven by a clock originating from the low-speed (OSC1) oscillation circuit, this timer cannot be used unless the low-speed (OSC1) oscillation circuit (32.768 kHz) is used.

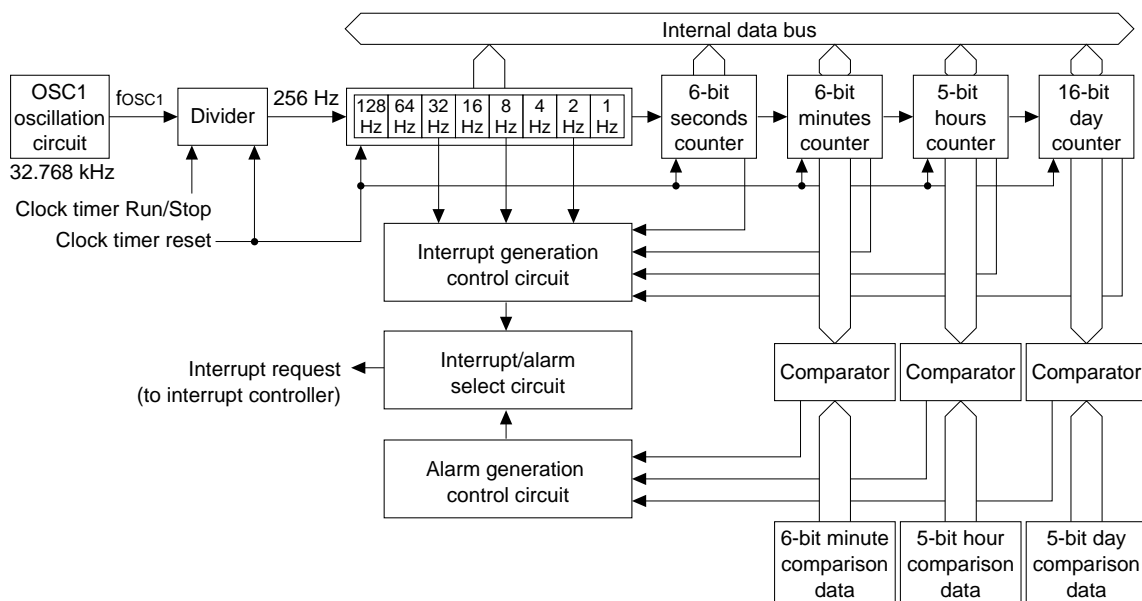


Figure III.8.1 Structure of Clock Timer

## Control and Operation of the Clock Timer

### Initial setting

At initial reset, the clock timer's counter data, setup contents of alarms, and control bits including RUN/STOP, are not initialized. (This does not include the CPU core power on/off flag TCHVOF or OSC1 auto-off flag TCAOFF.)

Therefore, when using the clock timer, initialize it as follows:

1. Before you start setting up, stop the clock timer and disable the clock timer interrupt.
2. Reset the counters.
3. Preset the minute, hour, and day data (only when necessary).
4. Select an interrupt factor.
5. Select the alarm function.
6. Enable the interrupt.
7. Start the clock timer.

The following shows how to set and control each of the above. For details on interrupt control, refer to "Interrupt Function".

### Resetting the counters

Each counter of the clock timer can only be reset to "0" in the software. Note that they are not reset by an initial reset or the auto-off function.

To reset the clock timer, write "1" to TCRST (D1) / Clock timer Run/Stop register (0x40151). Note, however, that this reset input is accepted only when the clock timer is inactive, and is ignored when the timer is operating.

- Notes:**
- The clock timer reset bit TCRST and the clock timer RUN/STOP control bit TCRUN are located at the same address (0x40151). However, the clock timer cannot be reset at the same time it is set to RUN by writing "1" to both. In this case, the reset input is ignored and the timer starts counting up from the counter values then in effect. Always make sure TCRUN = "0" before resetting the timer.
  - When the counters are cleared as the clock timer is reset, an interrupt may be generated depending on the timer settings. Therefore, first disable the clock timer interrupt before resetting the clock timer, and after resetting the clock timer, reset the interrupt factor flag, interrupt factor generation flag, and alarm factor generation flag.

### Presetting minute, hour, and day data

The clock timer's minute, hour, and day counters have a data preset function, enabling the desired time and day to be set.

Table III.8.1 Presetting the Counters

Counter	Data register	Preset value
Minute counter	TCHD[5:0] (D[5:0]) / Clock timer minute register (0x40155)	0 to 59
Hour counter	TCDD[4:0] (D[4:0]) / Clock timer hour register (0x40156)	0 to 23
Day counter	TCND[15:0] (D[7:0]) / Clock timer day (high-order) register (0x40158) (D[7:0]) / Clock timer day (low-order) register (0x40157)	0 to 65535

When using the clock timer as an RTC, be sure to set these counter values before starting operating of the clock timer. For the day counter, set a number of days starting from the reference day (e.g., January 1, 1990).

## RUN/STOP the clock timer

The clock timer starts counting when "1" is written to TCRUN (D0) / Clock timer Run/Stop register (0x40151) and stops counting when "0" is written.

When the clock timer is made to RUN, the 256-Hz clock input is enabled at a falling edge of the low-speed (OSC1) oscillation clock pulse, and the 8-bit binary counter counts up at each falling edge of this 256-Hz clock. Figure III.8.2 shows the operation of the 8-bit binary counter.

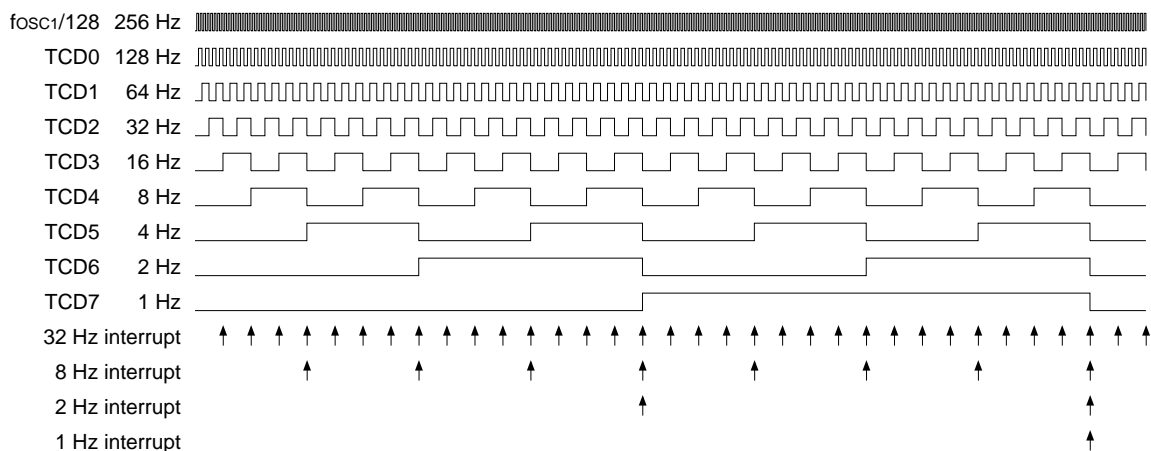


Figure III.8.2 Timing Chart of 8-Bit Binary Counter

The 8-bit binary counter outputs a 1-Hz signal in its final stage.

The second counter counts the 1-Hz signal thus output. When it counts 60 seconds, the counter outputs a 60-second signal and is reset to 0 seconds.

Similarly, the minute and hour counters count 60 minutes and 24 hours, respectively, using the signals output by each preceding counter.

The day counter is a 16-bit binary counter and can count up to 65,536 days using the 24-hour signal output by the hour counter.

One of the following signals output by each counter can be selected to generate an interrupt:

32 Hz, 8 Hz, 2 Hz, 1 Hz (1 second), 1 minute, 1 hour, 1 day

If "0" is written to TCRUN, the clock timer is stopped at a rising edge of the low-speed (OSC1) oscillation clock to prevent device malfunction caused by the concurrent termination of counting (falling edge of the 256-Hz clock).

Even when the clock timer is stopped, each counter retains the data set at that point. When the timer is made to RUN again while in that state, each counter restarts counting from the retained value.

## Reading out counter data

The data in each counter can be read out in a software as binary data.

Table III.8.2 Reading Out Counter Data

Counter	Data register
1 Hz to 128 Hz	TCD[7:0] (D[7:0]) / Clock timer divider register (0x40153)
Second counter	TCMD[5:0] (D[5:0]) / Clock timer second counter (0x40154)
Minute counter	TCHD[5:0] (D[5:0]) / Clock timer minute counter (0x40155)
Hour counter	TCDD[4:0] (D[4:0]) / Clock timer hour counter (0x40156)
Day counter	TCND[15:0] (D[7:0]) / Clock timer day (high-order) counter (0x40158) (D[7:0]) / Clock timer day (low-order) counter (0x40157)

Data is read directly from the counter during operation. For this reason, a counter can overflow while reading data from each counter, so the data thus read may not be exact. For example, if the 8-bit binary counter is read at 0xFF and then overflows before reading the next seconds counter, the value of the seconds counter is its count plus the one second that has elapsed since the 8-bit binary counter was read. To prevent this problem, try reading out each counter several times and make sure data has not been modified.

## Setting alarm function

The clock timer has an alarm function, enabling an interrupt to be generated at a specified time and day. This specification can be made in minutes, hours, and days for each alarm or a combination of multiple alarms. Use TCASE[2:0] (D[4:2]) / Clock timer interrupt control register (0x40152) for this specification.

Table III.8.3 Alarm Factor Selection

TCASE2	TCASE1	TCASE0	Alarm factor
X	X	1	Minutes alarm
X	1	X	Hours alarm
1	X	X	Day alarm
0	0	0	None

For example, if TCASE is set to "001", only a minutes alarm is enabled and an alarm is generated at a specified minute every hour. If TCASE is set to "111", an alarm is generated on each specified day at each specified hour and minute. If alarms are not to be used, set TCASE to "000".

An interrupt can be generated every minute, every hour, and every day through the use of the counter's interrupt function instead of the alarm function.

To specify a day, hours, and minutes, use the registers shown below:

To specify minutes: TCCH[5:0] (D[5:0]) / Minute-comparison data register (0x40159) 0 to 59 minutes\*

To specify hours: TCCD[4:0] (D[4:0]) / Hour-comparison data register (0x4015A) 0 to 23 hours\*

To specify day: TCCN[4:0] (D[4:0]) / Day-comparison data register (0x4015B) 0 to 31 days after

\* The minute-comparison data register (6 bits) and hour-comparison data register (5 bits) can be set for up to 63 minutes and 31 hours, respectively. Note that even when the data set in these registers exceeds 59 minutes or 23 hours, the data is not considered invalid.

The values set in these registers are compared with those of each counter, and when they match, the alarm factor generation flag TCAF (D0) / Clock timer interrupt control register (0x40152) is set to "1". If clock timer interrupts have been enabled using the interrupt controller, an interrupt is generated when the flag is set.

The day-comparison data register is a 5-bit register, and its value is compared with the five low-order bits of the day counter. Therefore, an alarm can be generated for up to 31 days after the register is set.

## Interrupt Function

### Clock timer interrupt factors

The clock timer can generate an interrupt using a 32-Hz, 8-Hz, 2-Hz, 1-Hz (1-second), 1-minute, 1-hour, or 1-day signal. The interrupt factor to be used from among these signals can be selected using the interrupt factor selection bit TCISE[2:0] (D[7:5]) / Clock timer interrupt control register (0x40152).

Table III.8.4 Selecting Interrupt Factor

TCISE2	TCISE1	TCISE0	Interrupt factor
1	1	1	None
1	1	0	1 day
1	0	1	1 hour
1	0	0	1 minute
0	1	1	1 Hz
0	1	0	2 Hz
0	0	1	8 Hz
0	0	0	32 Hz

An interrupt factor is generated at intervals of a selected signal (each falling edge of the signal). If interrupts based on these signals are not to be used, set TCISE to "111".

When a selected interrupt factor is generated, the interrupt factor generation flag TCIF (D1) / Clock timer interrupt control register (0x40152) is set to "1". At the same time, the clock timer interrupt factor flag FCTM (D1) / Port input 4–7, clock timer, A/D interrupt factor flag register (0x40287) also is set to "1". At this time, if the interrupt conditions set by the interrupt control registers are met, an interrupt to the CPU is generated.

An interrupt can be generated on a specified alarm day at a specified time as described in the preceding section. Interrupts generated by a signal and those generated by an alarm can both be used. However, since the clock timer has only one interrupt factor flag, it is the same interrupt that is generated by the timer. Therefore, if both types of interrupts are used, when an interrupt occurs, read the interrupt factor generation flag TCIF and alarm factor generation flag TCAF to determine which factor has generated the interrupt. Once the factor generation flag is set to "1", it remains set until it is reset by writing "1" in the software. After confirming that the flag is set, write "1" to reset it. The interrupt factor generation flag TCIF and alarm factor generation flag TCAF should be reset after at least 4 ms have passed from generation of an interrupt or an alarm.

**Note:** To prevent generation of an unwanted interrupt, disable the clock timer interrupt before selecting the interrupt and alarm factors. Then, before reenabling the interrupt, reset each factor generation flag and the interrupt factor flag.

### Control registers of the interrupt controller

The following lists the clock timer interrupt control registers:

Interrupt factor flag: FCTM (D1) / Port input 4–7, clock timer, A/D interrupt factor flag register (0x40287)

Interrupt enable: ECTM (D1) / Port input 4–7, clock timer, A/D interrupt enable register (0x40277)

Interrupt level: PCTM[2:0] (D[2:0]) / Clock timer interrupt priority register (0x4026B)

When an interrupt factor occurs, the clock timer sets the interrupt factor flag to "1" as described above. At this time, if the interrupt enable register bit is set to "1", an interrupt request is generated.

Interrupts can be disabled by leaving the interrupt enable register bit reset to "0". The interrupt factor flag is always set to "1" when an interrupt factor is generated, regardless of the setting of the interrupt enable register (even when it is set to "0").

The interrupt priority register sets the priority levels (0 to 7) of interrupts. An interrupt request to the CPU is accepted on the condition that no other interrupt request has been generated that is of a higher priority.

It is only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the clock timer interrupt level set by the interrupt priority register that a clock timer interrupt request is actually accepted by the CPU. For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to "ITC (Interrupt Controller)".

Note that the clock timer interrupt factor does not have a function to invoke an intelligent DMA.

### Trap vectors

The trap vector addresses for the clock-timer interrupt by default are set to 0x0C00104.

The trap table base address can be changed using the TTBR registers (0x48134 to 0x48137).

## Examples of Use of Clock Timer

---

The following shows examples of use of the clock timer and how to control the timer in each case.

### To use the clock timer as a timer/counter

Example in which while the CPU is inactive, the clock timer is kept operating in order to start again the CPU after a specified length of time has elapsed (e.g., three days):

1. Make sure the low-speed (OSC1) oscillation circuit is oscillating stably (SOSC1 = "1").  
Wait for approximately three seconds after the oscillation starts for its oscillation to stabilize.
2. Disable the clock timer interrupt using the interrupt controller (ECTM = "0").
3. Stop the clock timer and set "3 days" in the day-comparison register (TCRUN = "0", TCCN = "3").
4. Choose a "day-specified alarm" using the alarm-factor select bit and set "none" in the interrupt-factor select bit (TCASE = "100", TCISE = "111").
5. Reset the interrupt factor and alarm factor generation flags (FCTM = "0", TCAF = "0").
6. Reenable the clock timer interrupt using the interrupt controller (ECTM = "1").
7. Switch the CPU operating clock to the low-speed (OSC1) clock (CLKCHG = "0").
8. Turn off the high-speed (OSC3) oscillation circuit (SOSC3 = "0").
9. Reset the clock timer (TCRST = "0").
10. Start the clock timer (TCRUN = "1").
11. Execute the halt instruction to stop the CPU.

:

Wait until an interrupt is generated by a day-specified alarm from the clock timer. When an interrupt occurs, the CPU starts up using the OSC1 clock.

:

12. If necessary, turn on the high-speed (OSC3) oscillation circuit and change the CPU operating clock back to the OSC3 clock.

In the above example, if the device is reset before a three-day period has elapsed, the device operates as follows:

- The CPU starts up using the OSC3 clock.
- The clock timer counters are not reset. They remain in the RUN state.

The time during which the CPU has been idle can be checked by reading out the clock timer counters.

### For using the clock timer as RTC

Example in which the clock timer is kept operating and an alarm is generated at 10:00 A.M. every day:

1. Disable the clock timer interrupt using the interrupt controller (ECTM = "0").
2. Stop the clock timer (TCRUN = "0").
3. Reset the clock timer (TCRST = "1").
4. Set the current day and time in the minute (TCHD), hour (TCDD), and day (TCND) counters. For the day counter, set a number of days starting from the reference day (e.g., January 1, 1990). When the count is read, it is converted into the current date by the software.
5. Set "10:00" in the hour-compare register (TCCD = "0x0A").
6. Select an a "hour-specified alarm" using the alarm factor select bit, and set "none" in the interrupt factor select bit (TCASE = "010", TCISE = "111").
7. Reset the interrupt factor and alarm-factor generation flags (FCTM = "1", TCAF = "0").
8. Reenable the clock timer interrupt using the interrupt controller (ECTM = "1").
9. Start the clock timer (TCRUN = "1").

:

The clock timer is made to generate an interrupt at 10:00 every day by an hour-specified alarm.

:

In the above example, if any interrupt factor other than an alarm is selected, an interrupt is also generated by that interrupt factor. To determine which factor caused the interrupt generated, read the interrupt factor generation flag TCIF and alarm factor generation flag TCAF. If TCAF is set to 1, the interrupt has been caused by an alarm. If you select an interrupt factor (other than a 1-day factor) along with the hour-specified alarm, the selected interrupt factor occurs at the same time as the alarm factor.



## I/O Memory of Clock Timer

Table III.8.5 shows the control bits of the clock timer.

Table III.8.5 Control Bits of Clock Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
Clock timer Run/Stop register	0040151 (B)	D7-2	–	reserved	–	–	–	0 when being read.				
		D1	TCRST	Clock timer reset	1 Reset	0 Invalid	X	W				
		D0	TCRUN	Clock timer Run/Stop control	1 Run	0 Stop	X	R/W				
Clock timer interrupt control register	0040152 (B)	D7	TCISE2	Clock timer interrupt factor selection	TCISE[2:0]		Interrupt factor		X	R/W		
		D6	TCISE1		1	1	1	None				X
		D5	TCISE0		1	1	0	Day				X
					1	0	1	Hour				
					1	0	0	Minute				
					0	1	1	1 Hz				
					0	1	0	2 Hz				
				0	0	1	8 Hz					
				0	0	0	32 Hz					
		D4	TCASE2	Clock timer alarm factor selection	TCASE[2:0]		Alarm factor		X	R/W		
D3	TCASE1	1	X		X	Day	X					
D2	TCASE0	X	1		X	Hour	X					
		X	X		1	Minute						
		0	0	0	None							
D1	TCIF	Interrupt factor generation flag	1	Generated	0	Not generated	X	R/W	Reset by writing 1.			
D0	TCAF	Alarm factor generation flag	1	Generated	0	Not generated	X	R/W	Reset by writing 1.			
Clock timer divider register	0040153 (B)	D7	TCD7	Clock timer data 1 Hz	1 High	0 Low	X	R				
		D6	TCD6	Clock timer data 2 Hz	1 High	0 Low	X	R				
		D5	TCD5	Clock timer data 4 Hz	1 High	0 Low	X	R				
		D4	TCD4	Clock timer data 8 Hz	1 High	0 Low	X	R				
		D3	TCD3	Clock timer data 16 Hz	1 High	0 Low	X	R				
		D2	TCD2	Clock timer data 32 Hz	1 High	0 Low	X	R				
		D1	TCD1	Clock timer data 64 Hz	1 High	0 Low	X	R				
		D0	TCD0	Clock timer data 128 Hz	1 High	0 Low	X	R				
Clock timer second register	0040154 (B)	D7-6	–	reserved	–	–	–	0 when being read.				
		D5	TCMD5	Clock timer second counter data	0 to 59 seconds		X	R				
		D4	TCMD4	TCMD5 = MSB			X					
		D3	TCMD3	TCMD0 = LSB			X					
		D2	TCMD2				X					
		D1	TCMD1				X					
		D0	TCMD0				X					
Clock timer minute register	0040155 (B)	D7-6	–	reserved	–	–	–	0 when being read.				
		D5	TCHD5	Clock timer minute counter data	0 to 59 minutes		X	R/W				
		D4	TCHD4	TCHD5 = MSB			X					
		D3	TCHD3	TCHD0 = LSB			X					
		D2	TCHD2				X					
		D1	TCHD1				X					
		D0	TCHD0				X					
Clock timer hour register	0040156 (B)	D7-5	–	reserved	–	–	–	0 when being read.				
		D4	TCDD4	Clock timer hour counter data	0 to 23 hours		X	R/W				
		D3	TCDD3	TCDD4 = MSB			X					
		D2	TCDD2	TCDD0 = LSB			X					
		D1	TCDD1				X					
		D0	TCDD0				X					
Clock timer day (low-order) register	0040157 (B)	D7	TCND7	Clock timer day counter data	0 to 65535 days		X	R/W				
		D6	TCND6	(low-order 8 bits)	(low-order 8 bits)		X					
		D5	TCND5	TCND0 = LSB			X					
		D4	TCND4				X					
		D3	TCND3				X					
		D2	TCND2				X					
		D1	TCND1				X					
		D0	TCND0				X					
Clock timer day (high-order) register	0040158 (B)	D7	TCND15	Clock timer day counter data	0 to 65535 days		X	R/W				
		D6	TCND14	(high-order 8 bits)	(high-order 8 bits)		X					
		D5	TCND13	TCND15 = MSB			X					
		D4	TCND12				X					
		D3	TCND11				X					
		D2	TCND10				X					
		D1	TCND9				X					
		D0	TCND8				X					

### III-8 PERIPHERAL BLOCK: CLOCK TIMER

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Clock timer minute comparison register	0040159 (B)	D7-6	–	reserved	–	–	–	0 when being read.	
		D5	TCCH5	Clock timer minute comparison data TCCH5 = MSB TCCH0 = LSB	0 to 59 minutes (Note) Can be set within 0–63.	X	R/W		
		D4	TCCH4			X			
		D3	TCCH3			X			
		D2	TCCH2			X			
		D1	TCCH1			X			
D0	TCCH0	X							
Clock timer hour comparison register	004015A (B)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4	TCCD4	Clock timer hour comparison data TCCD4 = MSB TCCD0 = LSB	0 to 23 hours (Note) Can be set within 0–31.	X	R/W		
		D3	TCCD3			X			
		D2	TCCD2			X			
		D1	TCCD1			X			
		D0	TCCD0			X			
Clock timer day comparison register	004015B (B)	D7-5	–			reserved			–
		D4	TCCN4	Clock timer day comparison data TCCN4 = MSB TCCN0 = LSB	0 to 31 days	X	R/W	Compared with TCND[4:0].	
		D3	TCCN3			X			
		D2	TCCN2			X			
		D1	TCCN1			X			
		D0	TCCN0			X			
Clock timer interrupt priority register	004026B (B)	D7-3	–			reserved			–
		D2	PCTM2	Clock timer interrupt level	0 to 7	X	R/W		
		D1	PCTM1			X			
		D0	PCTM0			X			
Port input 4–7, clock timer, A/D interrupt enable register	0040277 (B)	D7-6	–	reserved	–		–	–	0 when being read.
		D5	EP7	Port input 7	1 Enabled	0 Disabled	0	R/W	
		D4	EP6	Port input 6			0	R/W	
		D3	EP5	Port input 5			0	R/W	
		D2	EP4	Port input 4			0	R/W	
		D1	ECTM	Clock timer			0	R/W	
		D0	EADE	A/D converter			0	R/W	
Port input 4–7, clock timer, A/D interrupt factor flag register	0040287 (B)	D7-6	–	reserved	–		–	–	0 when being read.
		D5	FP7	Port input 7	1 Factor is generated	0 No factor is generated	X	R/W	
		D4	FP6	Port input 6			X	R/W	
		D3	FP5	Port input 5			X	R/W	
		D2	FP4	Port input 4			X	R/W	
		D1	FCTM	Clock timer			X	R/W	
		D0	FADE	A/D converter			X	R/W	

#### TCRST: Clock timer reset (D1) / Clock timer Run/Stop register (0x40151)

Resets the clock timer.

Write "1": The clock timer is reset

Write "0": Invalid

Read: Always "0"

The clock timer is reset by writing "1" to TCRST when the timer is inactive. All timer counters are cleared to "0". The clock timer cannot be reset when in the RUN state, nor can it be reset at the same time it is made to RUN through the execution of one write to address 0x40151. (The clock timer is started, but not reset.) In this case, first reset the clock timer and then use another instruction to RUN the clock timer. When the counters are cleared as the clock timer is reset, an interrupt may be generated, depending on the register settings. Therefore, before resetting the clock timer, first disable the clock timer interrupt, and after resetting the clock timer, reset the interrupt factor flag and the interrupt factor and alarm factor generation flags.

Writing "0" to TCRST results in No Operation. Since this TCRST is a write-only bit, its value when read is always "0".

The clock timer is not reset by an initial reset.

**TCRUN:** Clock timer RUN/STOP control (D0) / Clock timer Run/Stop register (0x40151)

Controls the RUN/STOP of the clock timer.

Write "1": RUN  
 Write "0": STOP  
 Read: Valid

The clock timer is made to start counting by writing "1" to the TCRUN register and made to stop by writing "0". The timer data is retained even in the STOP state. The timer can also be made to start counting from the retained data by changing its state from STOP to RUN.

The TCRUN register is not initialized at initial reset.

**TCD7–TCDD0:** 1–128 Hz counter data (D[7:0]) / Clock timer divider register (0x40153)

**TCMD5–TCMD0:** Second counter data (D[5:0]) / Clock timer second register (0x40154)

**TCHD5–TCHD0:** Minute counter data (D[5:0]) / Clock timer minute register (0x40155)

**TCDD4–TCDD0:** Hour counter data (D[4:0]) / Clock timer hour register (0x40156)

**TCND15–TCND0:** Day counter data (D[7:0]) / Clock timer day (high-order) register (0x40158)  
 (D[7:0]) / Clock timer day (low-order) register (0x40157)

Data can be read out from each counter.

The minute, hour, and day counters allow data to be written to, in addition to being read out.

The 1–128 Hz counter and seconds counter are read-only, so writing to these registers is ignored.

The unused high-order bits at each address of the second, minute, and hour counter data are always "0" when read out.

The counter data is not initialized at initial reset.

**TCCH5–TCCH0:** Minute-comparison data (D[5:0]) / Clock timer minute-comparison register (0x40159)

**TCCD4–TCCD0:** Hour-comparison data (D[4:0]) / Clock timer hour-comparison register (0x4015A)

**TCCN4–TCCN0:** Day-comparison data (D[4:0]) / Clock timer day-comparison register (0x4015B)

Set a day on which and a time at which an alarm is to be generated.

The comparison data register corresponding to the alarm factor selected using the TCASE register is compared with the counter data, and when the data matches, an alarm interrupt request is generated.

The day-comparison data is compared with the 5 low-order bits of the day counter.

Each register can be read out.

These registers are not initialized at initial reset.

**TCISE2–TCISE0:** Interrupt factor selection (D[7:5]) / Clock timer interrupt control register (0x40152)

Selects the factor for which the clock timer interrupt is to be generated.

Table III.8.6 Selecting Interrupt Factor

TCISE2	TCISE1	TCISE0	Interrupt factor
1	1	1	None
1	1	0	1 day
1	0	1	1 hour
1	0	0	1 minute
0	1	1	1 Hz
0	1	0	2 Hz
0	0	1	8 Hz
0	0	0	32 Hz

When the clock timer interrupt is enabled, an interrupt is generated cyclically at each falling edge of the selected signal. If you the interrupt caused by these factors is not be used set TCISE to "111".

TCISE is not initialized at initial reset.

---

**TCASE2–TCASE0:** Alarm factor select register (D[4:2]) / Clock timer interrupt control register (0x40152)

---

Selects the factor for which an alarm is to be generated.

Table III.8.7 Selecting Alarm Factor

TCASE2	TCASE1	TCASE0	Alarm factor
X	X	1	Minutes alarm
X	1	X	Hours alarm
1	X	X	Day alarm
0	0	0	None

Use the TCASE2, TCASE1, and TCASE0 bits to select a day, hour, and minute alarm, respectively. It is therefore possible to select multiple alarm factors. When one of these bits is set to "1", the contents of the comparison data register that corresponds to the selected alarm factor is compared with the counter. If the comparison data of all selected alarm factors matches the counter data, an alarm interrupt request is generated. The comparison data register from which the alarm factor is unselected by writing "0" is not compared with the counter data. TCASE is not initialized at initial reset.

---

**TCIF:** Interrupt factor generation flag (D1) / Clock timer interrupt control register (0x40152)

---

Indicates whether an interrupt factor has occurred.

- Read "1": Interrupt factor has occurred
- Read "0": No interrupt factor has occurred
- Write "1": Flag is reset
- Write "0": Invalid

TCIF is set to "1" when an interrupt factor selected using TCISE occurs. Since there is only one source for the clock timer interrupt, use this flag to differentiate it from interrupts caused by an alarm.

Once set to "1", TCIF remains set until it is reset by writing "1".

TCIF is not initialized at initial reset.

This bit does not affect generation of an interrupt even if it is set to "1" or "0".

---

**TCAF:** Alarm factor generation flag (D0) / Clock timer interrupt control register (0x40152)

---

Indicates whether an alarm factor has occurred.

- Read "1": Alarm factor has occurred
- Read "0": No alarm factor has occurred
- Write "1": Flag is reset
- Write "0": Invalid

TCAF is set to "1" when all alarm factors selected using the TCASE register occur. Since there is only one source for the clock timer interrupt, use this flag to differentiate it from interrupts due to other interrupt factors.

Once set to "1", TCAF remains set until it is reset by writing "1".

TCAF is not initialized at initial reset.

This bit does not affect generation of an alarm even if it is set to "1" or "0".

---

**PCTM2–PCTM0:** Clock timer interrupt level (D[2:0]) / Clock timer interrupt priority register (0x4026B)

---

Sets the priority level of the clock timer interrupt between 0 and 7.

At initial reset, PCTM becomes indeterminate.

---

**ECTM:** Clock timer interrupt enable (D1) / Port input 4–7, clock timer, A/D interrupt enable register (0x40277)

---

Enables or disables generation of an interrupt to the CPU.

- Write "1": Interrupt enabled
- Write "0": Interrupt disabled
- Read: Valid

This bit controls the clock timer interrupt. The interrupt is enabled by setting ECTM to "1" and is disabled by setting it to "0".

At initial reset, ECTM is set to "0" (interrupt disabled).

---

**FCTM:** Clock timer interrupt factor flag (D1) / Port input 4–7, clock timer, A/D interrupt factor flag register (0x40287)

Indicates whether the clock timer interrupt factor has occurred.

**When read**

Read "1": Interrupt factor has occurred

Read "0": No interrupt factor has occurred

**When written using the reset-only method (default)**

Write "1": Interrupt factor flag is reset

Write "0": Invalid

**When written using the read/write method**

Write "1": Interrupt flag is set

Write "0": Interrupt flag is reset

FCTM is set to "1" when the selected interrupt factor or alarm factor occurs.

At this time, if the following conditions are met, an interrupt to the CPU is generated:

1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher interrupt priority is generated.
3. The IE bit of the PSR is set to "1" (interrupt enabled).
4. The corresponding interrupt priority register is set to a value higher than the CPU interrupt level (IL).

The interrupt factor flag is always set to "1" when an interrupt factor occurs, no matter how the interrupt enable and interrupt priority registers are set.

For the next interrupt to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).

The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept generated interrupts (or if the reti instruction is executed) without the interrupt factor flag being reset, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.

The FCTM flag becomes indeterminate at initial reset, so be sure to reset it in the software.

## ***Programming Notes***

---

- (1) The low-speed (OSC1) oscillation circuit, which is the clock source for the clock timer, requires a certain period of time for oscillation to stabilize (see "Electrical Characteristics") after it is started up. Therefore, immediately after power-on, wait until the oscillation stabilizes before starting the clock timer.
- (2) At initial reset, the clock timer counter data, the setup contents of alarms, and control bits, including RUN/STOP, are not initialized. Therefore, always initialize the clock timer in the software following power-on.
- (3) The clock timer reset bit TCRST and the clock timer RUN/STOP control bit TCRUN are located at the same address (0x40151). However, the clock timer cannot be reset at the same time it is set to RUN by writing "1" to both. In this case, the reset input is ignored and the timer starts counting up from the counter values then in effect. When resetting the timer, always make sure TCRUN = "0" (timer stopped).
- (4) When the counters are cleared as the clock timer is reset, an interrupt may be generated depending on the register settings. Therefore, before resetting the clock timer, first disable the clock timer interrupt and, after resetting the clock timer, reset the interrupt factor flag and the interrupt factor generation and alarm factor generation flags.
- (5) To prevent generation of an unwanted interrupt, disable the clock timer interrupt before selecting the interrupt and alarm factors. Then, before reenabling the interrupt, reset each factor generation flag and the interrupt factor flag.
- (6) The interrupt factor flag (FCTM) becomes indeterminate at initial reset. To prevent generation of an unwanted interrupt, be sure to reset the flag in a program.
- (7) To prevent regeneration of interrupts with the same factor after an interrupt has occurred, be sure to reset the interrupt factor flag (FCTM) before setting the PSR again or executing the reti instruction.

## III-9 SERIAL INTERFACE

### Configuration of Serial Interfaces

#### Features of Serial Interfaces

The Peripheral Block contains four channels (Ch.0, Ch.1, Ch.2 and Ch.3) of serial interfaces, the features of which are described below. The functions of these four serial interfaces are the same.

- A clock-synchronized or asynchronous mode can be selected for the transfer method.

Clock-synchronized mode

Data length: 8 bits, fixed (No start, stop, and parity bits)

Receive error: An overrun error can be detected.

Asynchronous mode

Data length: 7 or 8 bits, selectable

Receive error: Overrun, framing, or parity errors can be detected.

Start bit: 1 bit, fixed

Stop bit: 1 or 2 bits, selectable

Parity bit: Even, odd, or none; selectable

Since the transmit and receive units are independent, full-duplex communication is possible.

- Baud-rate setting: Any desired baud rate can be set by selecting the prescaler's division ratio, setting the 8-bit programmable timer, or using external clock input (asynchronous mode only).
- The receive and transmit units are constructed with a double-buffer structure, allowing for successive receive and transmit operations.
- Data transfers using IDMA or HSDMA are possible.
- Three types of interrupts (transmit data empty, receive data full, and receive error) can be generated.

Figure III.9.1 shows the configuration of the serial interface (one channel).

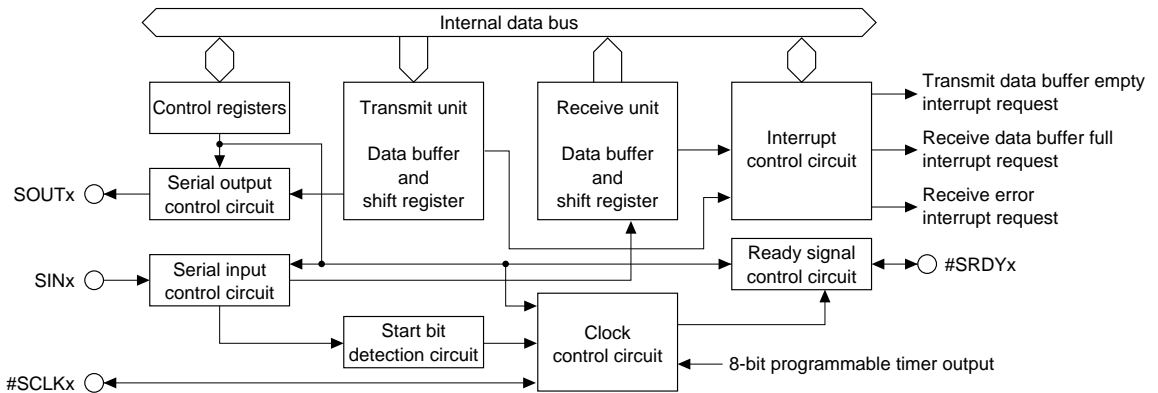


Figure III.9.1 Configuration of Serial Interface

**Note:** Ch.0 to Ch.3 have the same configuration and the same function. The signal and control bit names are suffixed by a 0, 1, 2, or 3 to indicate the channel number, enabling discrimination between channels 0 to 3. In this manual, however, channel numbers 0 to 3 are replaced with "x" unless discrimination is necessary, because explanations are common to all four channels.

## I/O Pins of Serial Interface

Table III.9.1 lists the I/O pins used by the serial interface.

Table III.9.1 Serial-Interface Pin Configuration

Pin name	I/O	Function	Function select bit
P00( <b>SINO</b> )	I/O	I/O port / Serial IF Ch.0 data input	CFP00(D0)/P0 function select register(0x402D0)
P01( <b>SOUT0</b> )	I/O	I/O port / Serial IF Ch.0 data output	CFP01(D1)/P0 function select register(0x402D0)
P02( <b>#SCLK0</b> )	I/O	I/O port / Serial IF Ch.0 clock input/output	CFP02(D2)/P0 function select register(0x402D0)
P03( <b>#SRDY0</b> )	I/O	I/O port / Serial IF Ch.0 ready input/output	CFP03(D3)/P0 function select register(0x402D0)
P04( <b>SIN1/#DMAACK2</b> )	I/O	I/O port / Serial IF Ch.1 data input / #DMAACK2 signal output	CFP04(D4)/P0 function select register(0x402D0) CFEX4(D4)/Port function extension register(0x402DF)
P05( <b>SOUT1/#DMAEND2</b> )	I/O	I/O port / Serial IF Ch.1 data output / #DMAEND2 signal output	CFP05(D5)/P0 function select register(0x402D0) CFEX5(D5)/Port function extension register(0x402DF)
P06( <b>#SCLK1/ #DMAACK3</b> )	I/O	I/O port / Serial IF Ch.1 clock input/output / #DMAACK3 signal output	CFP06(D6)/P0 function select register(0x402D0) CFEX6(D6)/Port function extension register(0x402DF)
P07( <b>#SRDY1/ #DMAEND3</b> )	I/O	I/O port / Serial IF Ch.1 ready input/output / #DMAEND3 signal output	CFP07(D7)/P0 function select register(0x402D0) CFEX7(D7)/Port function extension register(0x402DF)
P27(TM5/ <b>SIN2</b> )	I/O	I/O port / Serial IF Ch.2 data input	CFP27(D7)/Function select register(0x402D8) SSIN2(D0)/Port function extension register(0x402DB)
P26(TM4/ <b>SOUT2</b> )	I/O	I/O port / Serial IF Ch.2 data output	CFP26(D6)/Function select register(0x402D8) SSOUT2(D1)/Port function extension register(0x402DB)
P25(TM3/ <b>SCLK2</b> )	I/O	I/O port / Serial IF Ch.2 serial clock input/output	CFP25(D5)/Function select register(0x402D8) SSCLK2(D2)/Port function extension register(0x402DB)
P24(TM2/ <b>#SRDY2</b> )	I/O	I/O port / Serial IF Ch.2 ready input/output	CFP24(D4)/Function select register(0x402D8) SSRDY2(D3)/Port function extension register(0x402DB)
P33( <b>#DMAACK1/SIN3</b> )	I/O	I/O port / Serial IF Ch.3 data input	CFP33(D3)/Function select register(0x402DC) SSIN3(D0)/Port function extension register(0x402D7)
P16( <b>EXCL5/#DMAAND1/ SOUT3</b> )	I/O	I/O port / Serial IF Ch.3 data output	CFP16(D6)/Function select register(0x402D4) SSOUT3(D1)/Port function extension register(0x402D7)
P15( <b>EXCL4/#DMAAND0/ #SCLK3</b> )	I/O	I/O port / Serial IF Ch.3 serial clock input/output	CFP15(D5)/Function select register(0x402D4) SSCLK3(D2)/Port function extension register(0x402D7)
P32( <b>#DMAACK0/ #SRDY3</b> )	I/O	I/O port / Serial IF Ch.3 ready input/output	CFP32(D2)/Function select register(0x402DC) SSRDY3(D3)/Port function extension register(0x402D7)

### SINx (serial-data input pin)

This pin is used to input serial data to the device, regardless of the transfer mode.

### SOUTx (serial-data output pin)

This pin is used to output serial data from the device, regardless of the transfer mode.

### #SCLKx (clock input/output pin)

This pin is used to input or output a clock.

In the clock-synchronized slave mode, it is used as a clock input pin; in the clock-synchronized master mode, it is used as a clock output pin.

In the asynchronous mode, this pin is used as clock input when an external clock is used. This pin is not used when the internal clock is used, so it can be used as an I/O port.

### #SRDYx (ready-signal input/output pin)

This pin is used to input or output the ready signal that is used in the clock-synchronized mode.

In the clock-synchronized slave mode, it is used as a ready-signal output pin; in the clock-synchronized master mode, it is used as a ready-signal input pin.

This pin is not used in the asynchronous mode, so it can be used as an I/O port.

### Method for setting the serial-interface input/output pins

All of the pins used in the serial interface are shared with I/O ports. At cold start, they are all set for I/O port pins Pxx (function select bits CFPxx, CFEXx, SSxxx = "0"). When using the serial interface, make function select bit settings for the pins used, according to the channel and transfer mode to be used.

At hot start, the pins retain their status from prior to the reset.



## Setting Transfer Mode

The transfer mode of the serial interface can be set using SMDx[1:0] individually for each channel as shown in Table III.9.2 below.

Table III.9.2 Transfer Mode

SMDx1	SMDx0	Transfer mode
1	1	8-bit asynchronous mode
1	0	7-bit asynchronous mode
0	1	Clock-synchronized slave mode
0	0	Clock-synchronized master mode

At initial reset, SMDx becomes indeterminate, so be sure to initialize it in the software.

When using the IrDA interface, set the transfer mode for the asynchronous 7-bit or asynchronous 8-bit mode.

The input/output pins are configured differently, depending on the transfer mode. The pin configuration in each mode is shown in Table III.9.3.

Table III.9.3 Pin Configuration by Transfer Mode

Transfer mode	SINx	SOUTx	#SCLKx	#SRDYx
8-bit asynchronous mode	Data input	Data output	Clock input/P port	P port
7-bit asynchronous mode	Data input	Data output	Clock input/P port	P port
Clock-synchronized slave mode	Data input	Data output	Clock input	Ready output
Clock-synchronized master mode	Data input	Data output	Clock output	Ready input

All four pins are used in the clock-synchronized mode.

In the asynchronous mode, since #SRDYx is unused, P03 (or P07, P24, P23) can be used as an I/O (P) port. In addition, when an external clock is not used, P02 (or P06, P25, P15) can also be used as an I/O port.

The I/O control and data registers for the I/O ports used in the serial interface can be used as general-purpose read/write registers.

**Note:** To enable the IrDA interface to be set, IRMDx[1:0] (D[1:0]) / Serial I/F IrDA register (Ch.0: 0x401E4, Ch.1: 0x401E9, Ch.2: 0x401F4, Ch.3: 0x401F9) is provided. Since these bits become indeterminate at initial reset, be sure to initialize them by writing "00" when using as the normal interface or "10" when using as the IrDA interface.

## Clock-Synchronized Interface

### Outline of Clock-Synchronized Interface

In the clock-synchronized transfer mode, 8 bits of data are synchronized to the common clock on both the transmit and receive sides when the data is transferred. Since the transmit and receive units both have a double-buffer structure, successive transmit and receive operations are possible. Since the clock line is shared between the transmit and receive units, the communication mode is half-duplex.

#### Master and slave modes

Either the clock-synchronized master mode or the clock-synchronized slave mode can be selected using SMDx[1:0].

#### Clock-synchronized master mode (SMDx[1:0] = "00")

In this mode, clock-synchronized 8-bit serial transfers, in which the serial interface functions as the master, can be performed using the internal clock to synchronize the operation of the internal shift registers.

The synchronizing clock is output from the #SCLKx pin, enabling an external (slave side) serial input/output device to be controlled. The #SRDYx pin is also used to input a signal that indicates whether the external serial input/output device is ready to transmit or receive (when ready in a low level).

#### Clock-synchronized slave mode (SMDx[1:0] = "01")

In this mode, clock-synchronized 8-bit serial transfers, in which the serial interface functions as a slave, can be performed using the synchronizing clock that is supplied by an external (master side) serial input/output device. The synchronizing clock is input from the #SCLKx pin for use as the synchronizing clock of the serial interface. In addition, a #SRDYx signal indicating whether the serial interface is ready to transmit or receive (when ready in a low level) is output from the #SRDYx pin.

Figure III.9.2 shows an example of how the input/output pins are connected in the clock-synchronized mode.

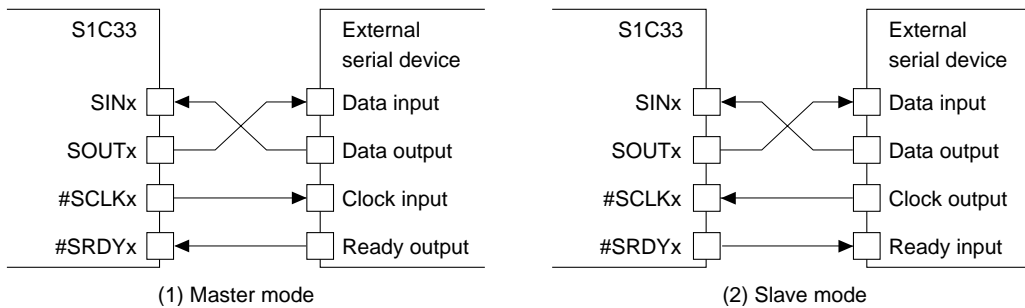


Figure III.9.2 Example of Connection in Clock-Synchronized Mode

#### Clock-synchronized transfer data format

In clock-synchronized transfers, the data format is fixed as shown below.

- Data length: 8 bits
- Start bit: None
- Stop bit: None
- Parity bit: None

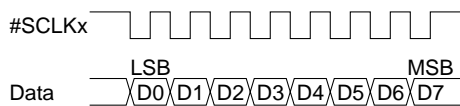


Figure III.9.3 Clock-Synchronized Transfer Data Format

Serial data is transmitted and received starting with the LSB.

## Setting Clock-Synchronized Interface

When performing clock-synchronized transfers via the serial interface, the following settings must be made before data transfer is actually begun:

1. Setting input/output pins
2. Setting the interface mode
3. Setting the transfer mode
4. Setting the input clock
5. Setting interrupts and IDMA/HSDMA

The following explains the content of each setting. For details on interrupt/DMA settings, refer to "Serial Interface Interrupts and DMA".

**Note:** Always make sure the serial interface is inactive (TXENx and RXENx = "0") before these settings are made. A change of settings during operation may cause a malfunction.

### Setting input/output pins

All four pins—SINx, SOUTx, #SCLKx, and #SRDYx—are used in the clock-synchronized mode. Configure the following registers according to the channel to be used (two or more channel can be used simultaneously).

Ch.0: CFP0[3:0] (D[3:0]) / P0 function select register (0x402D0) = "1111"

Ch.1: CFP0[7:4] (D[7:4]) / P0 function select register (0x402D0) = "1111"

CFEX[7:4] (D[7:4]) / Port function extension register (0x402DF) = "0000"

Ch.2: SSRDY2, SSCLK2, SSOUT2, SSIN2 (D[3:0]) / Port SIO function extension register (0x402DB) = "1111"

Ch.3: SSRDY3, SSCLK3, SSOUT3, SSIN3 (D[3:0]) / Port SIO function extension register (0x402D7) = "1111"

### Setting the interface mode

IRMDx[1:0] (D[1:0]) / Serial I/F Ch.0 IrDA register (0x401E4), Serial I/F Ch.1 IrDA register (0x401E9), Serial I/F Ch.2 IrDA register (0x401F4) or Serial I/F Ch.3 IrDA register (0x401F9) is used to set the interface mode (normal or IrDA interface). Write "00" to IRMDx[1:0] to choose the ordinary interface. Since IRMDx[1:0] becomes indeterminate at initial reset, it must be initialized.

### Setting the transfer mode

Use SMDx to set the transfer mode of the serial interface as described earlier. When using the serial interface as the master for clock-synchronized transfer, set SMDx[1:0] to "00"; when using the serial interface as a slave, set SMDx[1:0] to "01".

### Setting the input clock

#### • Clock-synchronized master mode

This mode operates using an internally derived clock. The clock source for each channel is as follows:

Ch.0: A clock output by 8-bit programmable timer 2

Ch.1: A clock output by 8-bit programmable timer 3

Ch.2: A clock output by 8-bit programmable timer 4

Ch.3: A clock output by 8-bit programmable timer 5

Therefore, in order for the serial interface to be used in the clock-synchronized master mode, the following conditions must be met:

1. The prescaler is feeding a clock to 8-bit programmable timer 2 (3).
2. The 8-bit programmable timer 2 (3) is generating a clock.

Any desired clock frequency can be selected by setting the division ratio of the prescaler and the reload data of the 8-bit programmable timer as necessary.

### III-9 PERIPHERAL BLOCK: SERIAL INTERFACE

The relationship between the contents of these settings and the transfer rate is expressed by Eq. 1 below.

To ensure that the duty ratio of the clock to be fed to the serial interface is 50%, the 8-bit programmable timer further divides the underflow signal frequency by 2 internally. This 1/2 frequency division is factored into Eq. 1.

$$\text{RLD} = \frac{\text{fPSCIN} \times \text{pdr}}{2 \times \text{bps}} - 1 \quad (\text{Eq. 1})$$

RLD: Reload data register setup value of the 8-bit programmable timer

fPSCIN: Prescaler input clock frequency (Hz)

bps: Transfer rate (bits/second)

pdr: Division ratio of the prescaler

**Note:** The division ratios selected by the prescaler differ between 8-bit programmable timers 2 and 3, so be careful when setting the ratio.

8-bit programmable timer 2, 4: 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/2048, 1/4096

8-bit programmable timer 3, 5: 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256

For details on how to control the prescaler and 8-bit programmable timers, refer to "Prescaler", and "8-Bit Programmable Timers".

The serial-interface control register contains an SSCKx bit to select the clock source used for the asynchronous mode. Although this bit does not affect the clock in the clock-synchronized mode, its content becomes indeterminate at initial reset. Therefore, be sure to initialize this bit by writing "0" (Internal clock), even when using the serial interface in the clock-synchronized master mode.

- **Clock-synchronized slave mode**

This mode operates using the clock that is output by the external master. This clock is input from the #SCLK pin.

Therefore, there is no need to control the prescaler or 8-bit programmable timer.

Initialize SSCKx by writing "1" (#SCLKx).

## Control and Operation of Clock-Synchronized Transfer

### Transmit control

#### (1) Enabling transmit operation

Use the transmit-enable bit TXENx for transmit control.

Ch.0 transmit-enable: TXEN0 (D7) / Serial I/F Ch.0 control register (0x401E3)

Ch.1 transmit-enable: TXEN1 (D7) / Serial I/F Ch.1 control register (0x401E8)

Ch.2 transmit-enable: TXEN2 (D7) / Serial I/F Ch.2 control register (0x401F3)

Ch.3 transmit-enable: TXEN3 (D7) / Serial I/F Ch.3 control register (0x401F8)

When transmit is enabled by writing "1" to this bit, the clock input to the shift register is enabled (ready for input), thus allowing for data to be transmitted. The synchronizing clock input/output of the #SCLKx pin is also enabled (ready for input/output).

Transmit is disabled by writing "0" to TXENx.

After the function select register is set for the serial interface, the I/O direction of the #SRDYx and #SCLKx pins are changed at follows:

#SRDYx: When slave mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

#SCLKx: When master mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

**Note:** In clock-synchronized transfers, the clock line is shared between the transmit and receive units, so the communication mode is half-duplex. Therefore, TXENx and receive-enable bit RXENx cannot be enabled simultaneously. When transmitting data, fix RXENx at "0" and do not change it during a transmit operation.

In addition, make sure TXENx is not set to "0" during a transmit operation.

#### (2) Transmit procedure

The serial interface contains a transmit shift register and a transmit data register (transmit data buffer), which are provided independently of those used for a receive operation.

Ch.0 transmit data: TXD0[7:0] (D[7:0]) / Serial I/F Ch.0 transmit data register (0x401E0)

Ch.1 transmit data: TXD1[7:0] (D[7:0]) / Serial I/F Ch.1 transmit data register (0x401E5)

Ch.2 transmit data: TXD2[7:0] (D[7:0]) / Serial I/F Ch.2 transmit data register (0x401F0)

Ch.3 transmit data: TXD3[7:0] (D[7:0]) / Serial I/F Ch.3 transmit data register (0x401F5)

The serial interface contains a status bit to indicate the status of the transmit data register.

Ch.0 transmit data buffer empty: TDBE0 (D1) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 transmit data buffer empty: TDBE1 (D1) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 transmit data buffer empty: TDBE2 (D1) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 transmit data buffer empty: TDBE3 (D1) / Serial I/F Ch.3 status register (0x401F7)

This bit is reset to "0" by writing data to the transmit-data register, and set to "1" again (buffer empty) when the data is transferred to the shift register.

The serial interface starts transmitting when data is written to the transmit data register.

The transfer status can be checked using the transmit-completion flag (TENDx).

Ch.0 transmit-completion flag: TEND0 (D5) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 transmit-completion flag: TEND1 (D5) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 transmit-completion flag: TEND2 (D5) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 transmit-completion flag: TEND3 (D5) / Serial I/F Ch.3 status register (0x401F7)

This bit goes "1" when data is being transmitted and goes "0" when the transmission has completed.

When data is transmitted successively in clock-synchronized master mode, TENDx maintains "1" until all data is transmitted (Figure III.9.4). In slave mode, TENDx goes "0" every time 1-byte data is transmitted (Figure III.9.5).

Following explains transmit operation in both the master and slave modes.

**• Clock-synchronized master mode**

The timing at which the device starts transmitting in the master mode is as follows:

When #SRDY is on a low level while TDBEx = "0" (the transmit-data register contains data written to it) or when TDBEx is set to "0" (data has been written to the transmit-data register) while #SRDY is on a low level.

Figure III.9.4 shows a transmit timing chart in the clock-synchronized master mode.

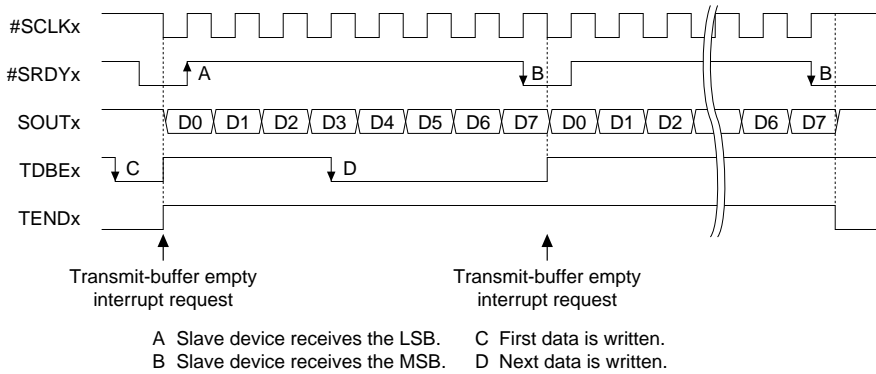


Figure III.9.4 Transmit Timing Chart in Clock-Synchronized Master Mode

1. If the #SRDYx signal from the slave is on a high level, the master waits until it is on a low level (ready to receive).
2. If #SRDYx is on a low level, the synchronizing clock input to the serial interface begins. The synchronizing clock is also output from the #SCLKx pin to the slave device.
3. The content of the data register is transferred to the shift register synchronously with the first falling edge of the clock. At the same time, the LSB of the data transferred to the shift register is output from the SOUTx pin.
4. The data in the shift register is shifted 1 bit by the next falling edge of the clock, and the bit following the LSB is output from SOUTx. This operation is repeated until all 8 bits of data are transmitted.

The slave device must take in each bit synchronously with the rising edges of the synchronizing clock.

**• Clock-synchronized slave mode**

Figure III.9.5 shows a transmit timing chart in the clock-synchronized slave mode.

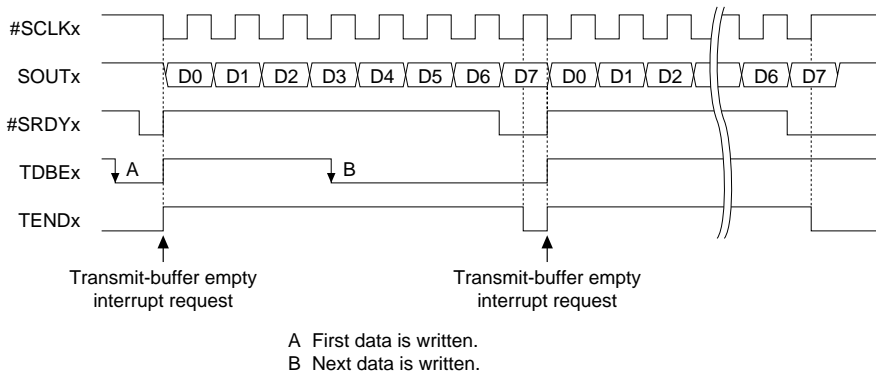


Figure III.9.5 Transmit Timing Chart in Clock-Synchronized Slave Mode

1. After setting the #SRDYx signal to a low level (ready to transmit), the slave waits for clock input from the master.
2. When the synchronizing clock is input from the #SCLKx pin, the content of the data register is transferred to the shift register synchronously with the first falling edge of the clock. At the same time, the LSB of the data transferred to the shift register is output from the SOUTx pin. The #SRDYx signal is returned to a high level at this point.
3. The data in the shift register is shifted 1 bit by the next falling edge of the clock, and the bit following the LSB is output from SOUTx. This operation is repeated until all 8 bits of data are transmitted.
4. The #SRDYx signal is set to a low level when the last bit (8th bit) is output from the SOUTx pin.

The master device must take in each bit synchronously with the rising edges of the synchronizing clock.

### • Successive transmit operations

When the data in the transmit data register is transferred to the shift register, TDBEx is reset to "1" (buffer empty). Once this occurs, the next transmit data can be written to the transmit data register, even during data transmission.

This allows data to be transmitted successively. The transmit procedure is described above.

When TDBEx is set to "1", a transmit-data empty interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the next piece of transmit data can be written using an interrupt processing routine. In addition, since this interrupt factor can be used to invoke DMA, the data prepared in memory can be transmitted successively to the transmit-data register through DMA transfers.

For details on how to control interrupts and DMA requests, refer to "Serial Interface Interrupts and DMA".

### (3) Terminating transmit operation

Upon completion of data transmission, write "0" to the transmit-enable bit TXENx to disable transmit operation.

## Receive control

### (1) Enabling receive operation

Use the receive-enable bit RXENx for receive control.

Ch.0 receive-enable: RXEN0 (D6) / Serial I/F Ch.0 control register (0x401E3)

Ch.1 receive-enable: RXEN1 (D6) / Serial I/F Ch.1 control register (0x401E8)

Ch.2 receive-enable: RXEN2 (D6) / Serial I/F Ch.2 control register (0x401F3)

Ch.3 receive-enable: RXEN3 (D6) / Serial I/F Ch.3 control register (0x401F8)

When receive operations are enabled by writing "1" to this bit, clock input to the shift register is enabled (ready for input), thereby starting a data-receive operation. The synchronizing clock input/output on the #SCLKx pin also is enabled (ready for input/output). Receive operations are disabled by writing "0" to RXENx.

After the function select register is set for the serial interface, the I/O direction of the #SRDYx and #SCLKx pins are changed at follows:

#SRDYx: When slave mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

#SCLKx: When master mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

**Note:** In clock-synchronized transfers, the clock line is shared between the transmit and receive units, so the communication mode is half-duplex. Therefore, RXENx and transmit-enable bit TXENx cannot be enabled simultaneously. When receiving data, fix TXENx at "0" and do not change it during a receive operation. In addition, make sure RXENx is not set to "0" during a receive operation.

### (2) Receive procedure

This serial interface has a receive shift register and a receive data register (receive data buffer) that are provided independently of those used for transmit operations.

Ch.0 receive data: RXD0[7:0] (D[7:0]) / Serial I/F Ch.0 receive data register (0x401E1)

Ch.1 receive data: RXD1[7:0] (D[7:0]) / Serial I/F Ch.1 receive data register (0x401E6)

Ch.2 receive data: RXD2[7:0] (D[7:0]) / Serial I/F Ch.2 receive data register (0x401F1)

Ch.3 receive data: RXD3[7:0] (D[7:0]) / Serial I/F Ch.3 receive data register (0x401F6)

The receive data can be read out from this register.

A status bit is also provided that indicates the status of the receive data register.

Ch.0 receive data buffer full: RDBF0 (D0) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 receive data buffer full: RDBF1 (D0) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 receive data buffer full: RDBF2 (D0) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 receive data buffer full: RDBF3 (D0) / Serial I/F Ch.3 status register (0x401F7)

This bit is set to "1" (buffer full) when the MSB of serial data is received and the data in the shift register is transferred to the receive data register, indicating that the received data can be read out. When the data is read out, the bit is reset to "0".

The following describes a receive operation in the master and slave modes.

**• Clock-synchronized master mode**

Figure III.9.6 shows a receive timing chart in the clock-synchronized master mode.

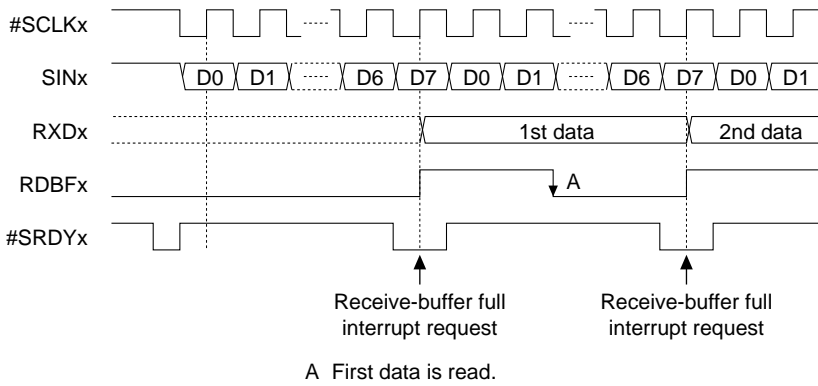
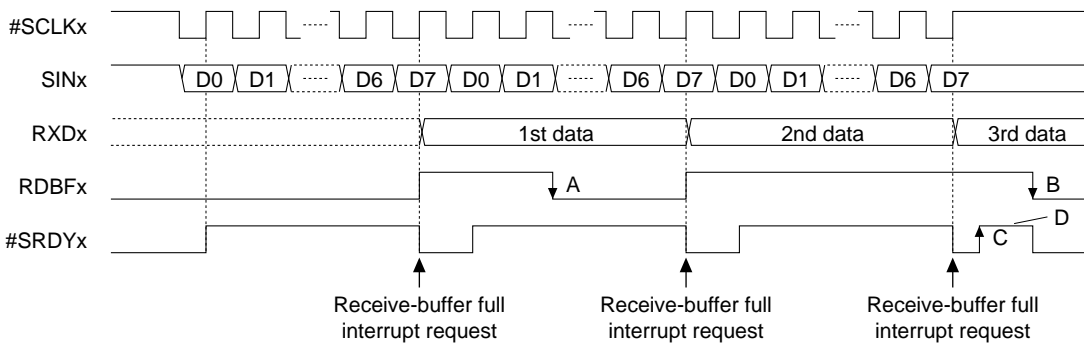


Figure III.9.6 Receive Timing Chart in Clock-Synchronized Master Mode

1. If the #SRDYx signal from the slave is on a high level, the master waits until it turns to a low level (ready to receive).
2. If #SRDYx is on a low level, synchronizing clock input to the serial interface begins. The synchronizing clock is also output from the #SCLKx pin to the slave device.
3. The slave device outputs each bit of data synchronously with the falling edges of the clock. The LSB is output first.
4. This serial interface takes the SINx input into the shift register at the rising edges of the clock. The data in the shift register is sequentially shifted as bits are taken in. This operation is repeated until the MSB of data is received.
5. When the MSB is taken in, the data in the shift register is transferred to the receive data register, enabling the data to be read out.

**• Clock-synchronized slave mode**

Figure III.9.7 shows a receive timing chart in the clock-synchronized slave mode.



- A First data is read. C An overrun error occurs because the receive operation has completed when RDBFx = "1".  
 B 3rd data is read. D Send the busy signal to the master device to stop the clock.

Figure III.9.7 Receive Timing Chart in Clock-Synchronized Slave Mode

1. After setting the #SRDYx signal to a low level (ready to receive), the slave waits for clock input from the master.
2. The master device outputs each bit of data synchronously with the falling edges of the clock. The LSB is output first.
3. This serial interface takes the SINx input into the shift register at the rising edges of the clock that is input from #SCLKx. The data in the shift register is sequentially shifted as bits are taken in. This operation is repeated until the MSB of data is received.
4. When the MSB is taken in, the data in the shift register is transferred to the receive data register, enabling the data to be read out.



### • Successive receive operations

When the data received in the shift register is transferred to the receive data register, RDBF<sub>x</sub> is set to "1" (buffer full), indicating that the received data can be read out.

Since the receive data register can be read out while receiving the next data, data can be received successively. The procedure for receiving is described above.

When RDBF<sub>x</sub> is set to "1", a receive-data full interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the received data can be read by an interrupt processing routine. In addition, since this interrupt factor can be used to invoke DMA, the received data can be received successively in locations prepared in memory through DMA transfers.

For details on how to control interrupts/DMA, refer to "Serial Interface Interrupts and DMA".

### (3) Overrun error

If, during successive receive operation, a receive operation for the next data is completed before the receive data register is read out, the receive data register is overwritten with the new data. Therefore, the receive data register must always be read out before a receive operation for the next data is completed.

When the receive data register is overwritten, an overrun error is generated and the overrun error flag is set to "1".

Ch.0 overrun error flag: OER0 (D2) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 overrun error flag: OER1 (D2) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 overrun error flag: OER2 (D2) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 overrun error flag: OER3 (D2) / Serial I/F Ch.3 status register (0x401F7)

Once the overrun error flag is set to "1", it remains set until it is reset by writing "0" to it in the software.

The overrun error is one of the receive-error interrupt factors in the serial interface. An interrupt can be generated for this error by setting the interrupt controller as necessary, so that the error can be processed by an interrupt processing routine.

### (4) #SRDY<sub>x</sub> in slave mode

When receive operations are enabled by writing "1" to RXEN<sub>x</sub>, the #SRDY<sub>x</sub> signal is turned to a low level, thereby indicating to the master device that the slave is ready to receive. When the LSB of serial data is received, #SRDY<sub>x</sub> is turned to a high level; when the MSB is received, #SRDY<sub>x</sub> is returned to a low level, in preparation for the next receive operation.

If an overrun error occurs, #SRDY<sub>x</sub> is turned to a high level (unable to receive) at that point, with receive operations for the following data thus suspended. In this case, #SRDY<sub>x</sub> is returned to a low by reading out the data overwritten in the receive data register, and if any receive data follows, the slave restarts receiving data.

### (5) Terminating receive operation

Upon completion of a data receive operation, write "0" to the receive-enable bit RXEN<sub>x</sub> to disable receive operations.

## Asynchronous Interface

### Outline of Asynchronous Interface

Asynchronous transfers are performed by adding a start bit and a stop bit to the start and end points of each serial-converted data. With this method, there is no need to use a clock that is fully synchronized on the transmit and receive sides; instead, transfer operations are timed by the start and stop bits added to the start and end points of each data.

In the 8-bit asynchronous mode (SMDx[1:0] = "11"), 8 bits of data can be transferred; in the 7-bit asynchronous mode (SMDx[1:0] = "10"), 7 bits of data can be transferred.

In either mode, it is possible to select the stop-bit length, add a parity bit, and choose between even and odd parity. The start bit is fixed at "1".

The operating clock can be selected between an internal clock generated by an 8-bit programmable timer or an external clock that is input from the #SCLKx pin.

Since the transmit and receive units are both constructed with a double-buffer structure, successive transmit and receive operations are possible. Furthermore, since the transmit and receive units are independent, full-duplex communication in which transmit and receive operations are performed simultaneously is also possible.

Figure III.9.8 shows an example of how input/output pins are connected for transfers in the asynchronous mode.

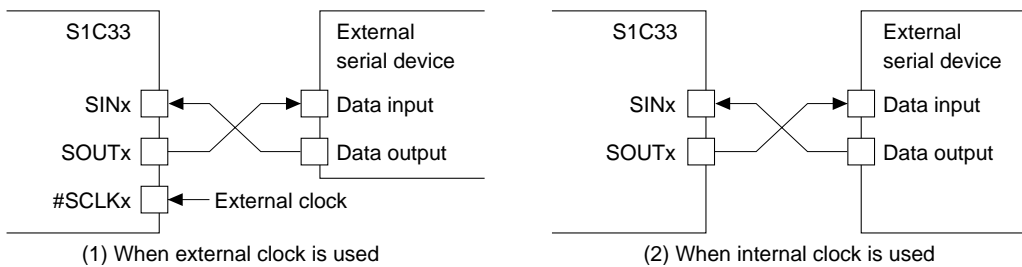


Figure III.9.8 Example of Connection in Asynchronous Mode

When the asynchronous mode is selected, it is possible to use the IrDA interface function.

### Asynchronous-transfer data format

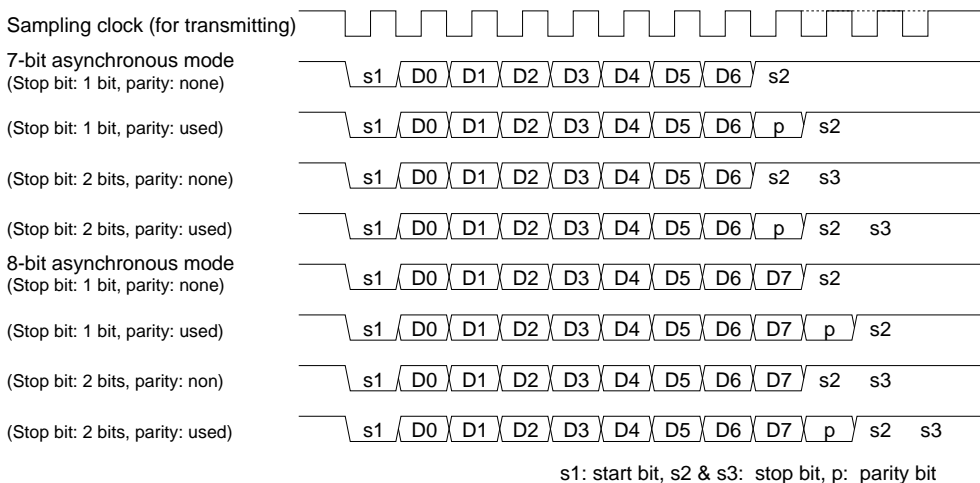
The data format for asynchronous transfer is shown below.

Data length: 7 or 8 bits (determined by the selected transfer mode)

Start bit: 1 bit, fixed

Stop bit: 1 or 2 bits

Parity bit: Even or odd parity, or none



s1: start bit, s2 & s3: stop bit, p: parity bit

Figure III.9.9 Data Format for Asynchronous Transfer

Serial data is transmitted and received, starting with the LSB.

## Setting Asynchronous Interface

When performing asynchronous transfer via the serial interface, the following must be done before data transfer can be started:

1. Setting input/output pins
2. Setting the interface mode
3. Setting the transfer mode
4. Setting the input clock
5. Setting the data format
6. Setting interrupt/IDMA/HSDMA

The following describes how to set each of the above. For details on interrupt/DMA settings, refer to "Serial Interface Interrupts and DMA".

**Note:** Always make sure the serial interface is inactive (TXENx and RXENx = "0") before making these settings. A change in settings during operation may result in a malfunction.

### Setting input/output pins

In the asynchronous mode, two pins—SINx and SOUTx—are used. When external clock input is used, one more pin, #SCLKx, is also used. Configure the following registers according to the channel to be used (two or more channel can be used simultaneously).

Ch.0: CFP0[3:0] (D[3:0]) / P0 function select register (0x402D0) = "0011"\*

Ch.1: CFP0[7:4] (D[7:4]) / P0 function select register (0x402D0) = "0011"\*

CFEX[7:4] (D[7:4]) / Port function extension register (0x402DF) = "0000" or "1100"

Ch.2: SSRDY2, SSCLK2, SSOUT2, SSIN2 (D[3:0]) / Port SIO function extension register (0x402DB) = "0011"\*

Ch.3: SSRDY3, SSCLK3, SSOUT3, SSIN3 (D[3:0]) / Port SIO function extension register (0x402D7) = "0011"\*

\* "0111" when #SCLKx is used.

### Setting the interface mode

IRMDx[1:0] (D[1:0]) / Serial I/F IrDA register (Ch.0: 0x401E4, Ch.1: 0x401E9, Ch.2: 0x401F4, Ch.3: 0x401F9) is used to set the IrDA interface. Since IRMDx[1:0] becomes indeterminate at initial reset, initialize it by writing "00" when using the serial interface as a normal interface, or "10" when using the serial interface as an IrDA interface. This setting must be made before a transfer mode is set.

### Setting the transfer mode

Use SMDx to set the transfer mode of the serial interface as described earlier. When using the serial interface in the 8-bit asynchronous mode, set SMDx[1:0] to "11", when using the serial interface in the 7-bit asynchronous mode, set SMDx[1:0] to "10".

### Setting the input clock

In the asynchronous mode, the operating clock can be selected between the internal clock and an external clock.

Ch.0 input clock selection: SSCK0 (D2) / Serial I/F Ch.0 control register (0x401E3)

Ch.1 input clock selection: SSCK1 (D2) / Serial I/F Ch.1 control register (0x401E8)

Ch.2 input clock selection: SSCK2 (D2) / Serial I/F Ch.2 control register (0x401F3)

Ch.3 input clock selection: SSCK3 (D2) / Serial I/F Ch.3 control register (0x401F8)

The external clock is selected (input from the #SCLKx pin) by writing "1" to SSCKx, and an internal clock is selected by writing "0".

**Note:** SSCKx becomes indeterminate at initial reset, so be sure to reset it in the software.

### III-9 PERIPHERAL BLOCK: SERIAL INTERFACE

- **Internal clock**

When the internal clock is selected, the serial interface is clocked by a clock generated using an 8-bit programmable timer. The clock source for each channel is as follows:

- Ch.0: Clock output by 8-bit programmable timer 2
- Ch.1: Clock output by 8-bit programmable timer 3
- Ch.2: Clock output by 8-bit programmable timer 4
- Ch.3: Clock output by 8-bit programmable timer 5

Therefore, before the internal clock can be used, the following conditions must be met:

1. The prescaler is outputting a clock to the 8-bit programmable timer 2 (or 3).
2. The 8-bit programmable timer 2 (or 3) is outputting a clock.

Any desired clock frequency can be obtained by setting the prescaler division ratio and the reload data of the 8-bit programmable timer as necessary. The relationship between the contents of these setting and the transfer rate is expressed by Eq. 2.

The 8-bit programmable timer has its underflow signal further divided by 2 internally, in order to ensure that the duty ratio of the clock supplied to the serial interface is 50%.

Furthermore, the clock output by the 8-bit programmable timer is divided by 16 or 8 internally in the serial interface, in order to create a sampling clock (refer to "Sampling clock"). This division ratio must also be considered when setting the transfer rate.

These division ratios are taken into account in Eq. 2.

$$RLD = \frac{fPSCIN \times pdr \times sdr}{2 \times bps} - 1 \quad (\text{Eq. 2})$$

RLD: Set value of the 8-bit programmable timer's reload data register

fPSCIN: Prescaler input clock frequency (Hz)

bps: Transfer rate (bits/second)

pdr: Division ratio of the prescaler

sdr: Internal division ratio of the serial interface (1/16 or 1/8)

**Note:** The division ratio selected using the prescaler differs between 8-bit programmable timers 2 and 3.

Take this into account when setting a division ratio.

8-bit programmable timer 2, 4: 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/2048, 1/4096

8-bit programmable timer 3, 5: 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256

Table III.9.4 shows examples of prescaler division ratios and the reload data settings of the programmable timer, in cases in which the internal division ratio of the serial interface is set to 1/16.

Table III.9.4 Example of Transfer Rate Settings

Transfer rate (bps)	fPSCIN = 20 MHz			fPSCIN = 25 MHz			fPSCIN = 33 MHz		
	RLD	pdr	Error (%)	RLD	pdr	Error (%)	RLD	pdr	Error (%)
300	129	1/16	0.16025	162	1/16	-0.14698	216	1/16	0.00640
1200	129	1/4	0.16025	162	1/4	-0.14698	216	1/4	0.00640
2400	129	1/2	0.16025	162	1/2	-0.14698	216	1/2	0.00640
4800	64	1/2	0.16025	80	1/2	-0.46939	108	1/2	-0.45234
9600	32	1/2	-1.35732	40	1/2	-0.75584	53	1/2	0.46939
14400	21	1/2	-1.35732	13	1/4	-3.11880	35	1/2	0.46939
28800	10	1/2	-1.35732	13	1/2	-3.11880	17	1/2	0.46939

Make sure the error is within 1%. Calculate the error using the following equation:

$$\text{Error} = \left\{ \frac{fPSCIN \times pdr}{(RLD + 1) \times 32 \times bps} - 1 \right\} \times 100 [\%]$$

For details on how to control the prescaler and 8-bit programmable timers, refer to "Prescaler" and "8-Bit Programmable Timers".

### • External clock

When an external clock is selected, the serial interface is clocked by a clock input from the #SCLKx pin. Therefore, there is no need to control the prescaler and 8-bit programmable timers.

Any desired clock frequency can be set. The clock input from the #SCLKx pin is internally divided by 16 or 8 in the serial interface, in order to create a sampling clock (refer to "Sampling clock"). This division ratio must also be considered when setting the transfer rate.

### • Sampling clock

In the asynchronous mode, TCLK (the clock output by the 8-bit programmable timer or input from the #SCLKx pin) is internally divided in the serial interface, in order to create a sampling clock.

A 1/16 division ratio is selected by writing "0" to DIVMDx, and a 1/8 ratio is selected by writing "1".

Ch.0 clock division ratio selection: DIVMD0 (D4) / Serial I/F Ch.0 IrDA register (0x401E4)

Ch.1 clock division ratio selection: DIVMD1 (D4) / Serial I/F Ch.1 IrDA register (0x401E9)

Ch.2 clock division ratio selection: DIVMD2 (D4) / Serial I/F Ch.2 IrDA register (0x401F4)

Ch.3 clock division ratio selection: DIVMD3 (D4) / Serial I/F Ch.3 IrDA register (0x401F9)

**Note:** The DIVMDx bit becomes indeterminate at initial reset, so be sure to reset it in the software. Settings of this bit are valid only in the asynchronous mode (and when using the IrDA interface).

#### For receiving

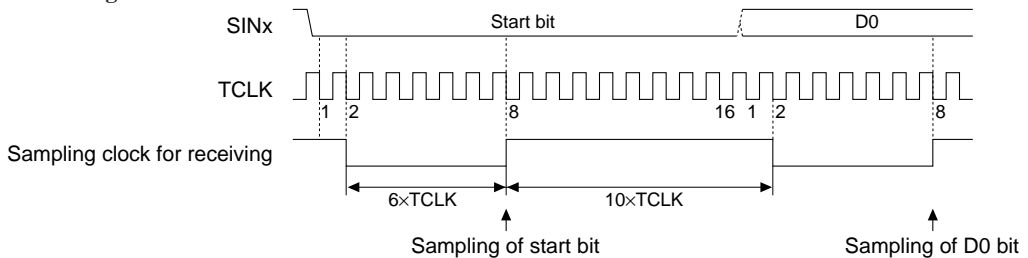


Figure III.9.10 Sampling Clock for Asynchronous Receive Operation (when 1/16 division is selected)

As shown in Figure III.9.10, the sampling clock is created by dividing TCLK by 16 (or 8). Its duty ratio (low: high ratio) is 6:10 (or 2:6 when divided by 8), and not 50%. Since the receive data is sampled in the middle point of each bit, the sampling clock recognizes the start bit first, and then changes the level from high to low at the second falling edge of TCLK. And at the 8th (4th for 1/8) falling edge of TCLK, it changes the level from low to high. This change in levels is repeated for the following bits of data:

Each bit of data is sampled at each rising edge of this sampling clock. When the stop bit is sampled, the sampling clock is fixed at high level until the next start bit is sampled.

If the SINx pin is returned to high level at the second falling edge of TCLK when it recognize the start bit, the data is assumed to be noise, and generation of the sampling clock is stopped.

If the SINx pin is not on a low level when the start bit is sampled at the 8th (4th for 1/8) clock, such as when the baud rate is not matched between the transmit and receive units, the serial interface stops sampling the following data and returns to a start-bit detection mode. In this case, no error is generated.

#### For transmitting

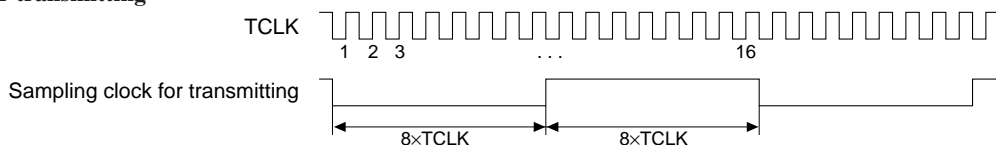


Figure III.9.11 Sampling Clock for Asynchronous Transmit Operation (when 1/16 division is selected)

When transmitting data, a sampling clock of a 50% duty cycle is generated from TCLK by dividing it by 16 (or 8), and each bit of data is output synchronously with this clock.

## Setting the data format

In the asynchronous mode, the data length is 7 or 8 bits as determined by the transfer mode set. The start bit is fixed at 1.

The stop and parity bits can be set as shown in the Table III.9.5 using the following control bits:

Table III.9.5 Serial I/F Control Bits

Item	Ch.0 (Serial I/F Ch.0 control register)	Ch.1 (Serial I/F Ch.1 control register)	Ch.2 (Serial I/F Ch.2 control register)	Ch.3 (Serial I/F Ch.3 control register)
Stop-bit select	STPB0(D3/0x401E3)	STPB1(D3/0x401E8)	STPB2(D3/0x401F3)	STPB3(D3/0x401F8)
Parity enable	EPR0(D5/0x401E3)	EPR1(D5/0x401E8)	EPR2(D5/0x401F3)	EPR3(D5/0x401F8)
Parity-mode select	PMD0(D4/0x401E3)	PMD1(D4/0x401E8)	PMD2(D4/0x401F3)	PMD3(D4/0x401F8)

Table III.9.6 Stop Bit and Parity Bit Settings

STPBx	EPRx	PMDx	Stop bit	Parity bit
1	1	1	2 bits	Odd
		0	2 bits	Even
0	1	*	2 bits	None
		1	1 bit	Odd
	0	0	1 bit	Even
0	0	1	1 bit	Odd
		0	1 bit	Non

\* Setting PMDx is invalid when EPRx = "0".

**Note:** These bits become indeterminate at initial reset, so be sure to initialize them in the software.

## Control and Operation of Asynchronous Transfer

### Transmit control

#### (1) Enabling transmit operation

Use the transmit-enable bit TXENx for transmit control.

Ch.0 transmit-enable: TXEN0 (D7) / Serial I/F Ch.0 control register (0x401E3)

Ch.1 transmit-enable: TXEN1 (D7) / Serial I/F Ch.1 control register (0x401E8)

Ch.2 transmit-enable: TXEN2 (D7) / Serial I/F Ch.2 control register (0x401F3)

Ch.3 transmit-enable: TXEN3 (D7) / Serial I/F Ch.3 control register (0x401F8)

When transmit is enabled by writing "1" to this bit, the clock input to the shift register is enabled (ready for input), thus allowing data to be transmitted.

Transmit is disabled by writing "0" to TXENx.

**Note:** Do not set TXENx to "0" during a transmit operation.

#### (2) Transmit procedure

The serial interface has a transmit shift register and a transmit data register (transmit data buffer) that are provided independently of those used for receive operations.

Ch.0 transmit data: TXD0[7:0] (D[7:0]) / Serial I/F Ch.0 transmit data register (0x401E0)

Ch.1 transmit data: TXD1[7:0] (D[7:0]) / Serial I/F Ch.1 transmit data register (0x401E5)

Ch.2 transmit data: TXD2[7:0] (D[7:0]) / Serial I/F Ch.2 transmit data register (0x401F0)

Ch.3 transmit data: TXD3[7:0] (D[7:0]) / Serial I/F Ch.3 transmit data register (0x401F5)

The serial interface starts a transmit operation by writing data to this register. In the 7-bit asynchronous mode, bit 7 (MSB) in each register is ignored.

The serial interface also contains a status bit to indicate the status of the transmit data register.

Ch.0 transmit data buffer empty: TDBE0 (D1) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 transmit data buffer empty: TDBE1 (D1) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 transmit data buffer empty: TDBE2 (D1) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 transmit data buffer empty: TDBE3 (D1) / Serial I/F Ch.3 status register (0x401F7)

This bit is reset to "0" by writing data to the transmit data register, and set back to "1" (buffer empty) when the data is transferred to the shift register. The transfer begins when the serial interface starts sending the start bit.

The transfer status can be checked using the transmit-completion flag (TENDx).

Ch.0 transmit-completion flag: TEND0 (D5) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 transmit-completion flag: TEND1 (D5) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 transmit-completion flag: TEND2 (D5) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 transmit-completion flag: TEND3 (D5) / Serial I/F Ch.3 status register (0x401F7)

This bit goes "1" when data is being transmitted and goes "0" when the transmission has completed.

When data is transmitted successively in asynchronous mode, TENDx maintains "1" until all data is transmitted.

Figure III.9.12 shows a transmit timing chart in the asynchronous mode.

Example: Data length 8 bits

Stop bit 1 bit

Parity bit Included

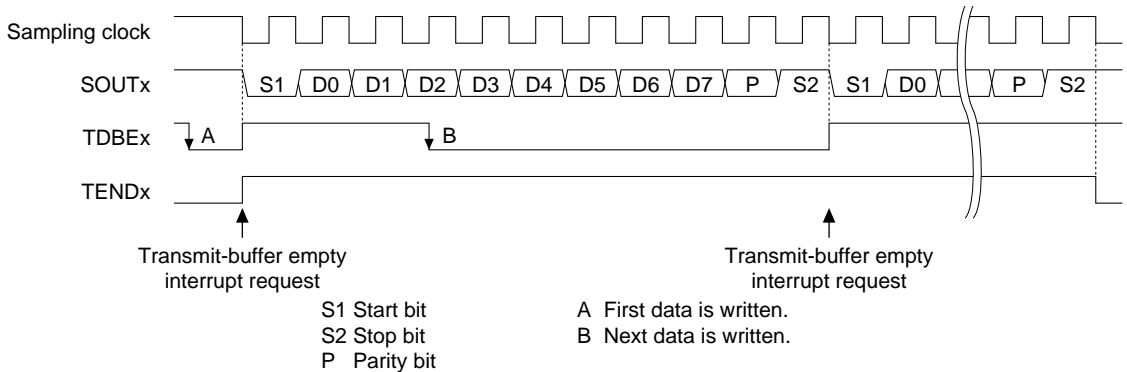


Figure III.9.12 Transmit Timing Chart in Asynchronous Mode

1. The contents of the data register are transferred to the shift register synchronously with the first falling edge of the sampling clock. At the same time, the SOUTx pin is setting to a low level to send the start bit.
2. Each bit of data in the shift register is transmitted beginning with the LSB at each falling edge of the subsequent sampling clock. This operation is repeated until all 8 (or 7) bits of data are transmitted.
3. After sending the MSB, the parity bit (if EPRx = "1") and the stop bit are transmitted in succession.

• **Successive transmit operation**

When the data in the transmit data register is transferred to the shift register, TDBEx is reset to "1" (buffer empty). Once this occurs, the next transmit data can be written to the transmit data register, even during data transmission.

This allows data to be transmitted successively. The transmit procedure is described above.

When TDBEx is set to "1", a transmit-data empty interrupt factor simultaneously occurs. Since an interrupt can be generated as set by the interrupt controller, the next transmit data can be written using an interrupt processing routine. In addition, since this interrupt factor can be used to invoke IDMA, the data prepared in memory can be transmitted successively to the transmit data register through DMA transfers.

For details on how to control interrupts and IDMA requests, refer to "Serial Interface Interrupts and DMA".

**(3) Terminating transmit operations**

When data transmission is completed, write "0" to the transmit-enable bit TXENx to disable transmit operations.

Receive control

(1) Enabling receive operations

Use the receive-enable bit RXENx for receive control.

Ch.0 receive-enable: RXEN0 (D6) / Serial I/F Ch.0 control register (0x401E3)

Ch.1 receive-enable: RXEN1 (D6) / Serial I/F Ch.1 control register (0x401E8)

Ch.2 receive-enable: RXEN2 (D6) / Serial I/F Ch.2 control register (0x401F3)

Ch.3 receive-enable: RXEN3 (D6) / Serial I/F Ch.3 control register (0x401F8)

When receiving enabled by writing "1" to this bit, clock input to the shift register is enabled (ready for input), meaning that it is ready to receive data. Receive operations are disabled by writing "0" to RXENx.

**Note:** Do not set RXENx to "0" during a receive operation.

(2) Receive procedure

This serial interface has a receive shift register and a receive data register (receive data buffer) that are provided independently of those used for transmit operations.

Ch.0 receive data: RXD0[7:0] (D[7:0]) / Serial I/F Ch.0 receive data register (0x401E1)

Ch.1 receive data: RXD1[7:0] (D[7:0]) / Serial I/F Ch.1 receive data register (0x401E6)

Ch.2 receive data: RXD2[7:0] (D[7:0]) / Serial I/F Ch.2 receive data register (0x401F1)

Ch.3 receive data: RXD3[7:0] (D[7:0]) / Serial I/F Ch.3 receive data register (0x401F6)

Receive data can be read out from this register.

A status bit is also provided to indicate the status of the receive data register.

Ch.0 receive data buffer full: RDBF0 (D0) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 receive data buffer full: RDBF1 (D0) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 receive data buffer full: RDBF2 (D0) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 receive data buffer full: RDBF3 (D0) / Serial I/F Ch.3 status register (0x401F7)

This bit is set to "1" (buffer full) when data is transferred from the shift register to the receive data register after the stop bit is sampled (the second bit if two stop bits are used), indicating that the received data can be read out. When the data is read out, the bit is reset to "0".

Figure III.9.13 shows a receive timing chart in the asynchronous mode.

Example: Data length 8 bits

Stop bit 1 bit

Parity bit Included

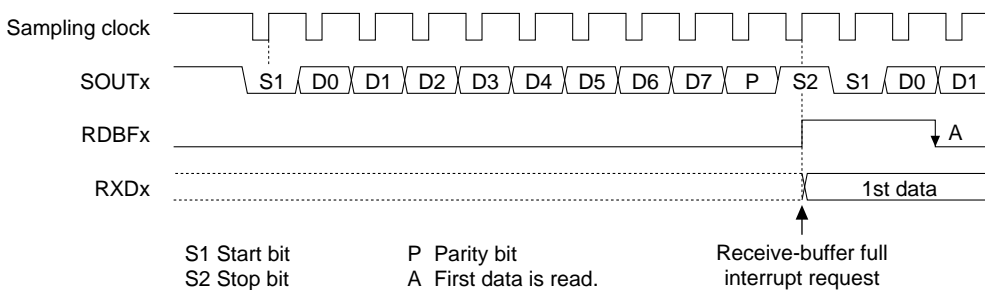


Figure III.9.13 Receive Timing Chart in Asynchronous Mode

1. The serial interface starts sampling when the start bit is input (SINx = low).
2. When the start bit is sampled at the first rising edge of the sampling clock, each bit of receive data is taken into the shift register, beginning with the LSB at each rising edge of the subsequent clock. This operation is repeated until the MSB of data is received.
3. When the MSB is taken in, the parity bit that follows is also taken in (if EPRx = "1").
4. When the stop bit is sampled, the data in the shift register is transferred to the receive data register, enabling the data to be read out.

The parity is checked when data is transferred to the receive data register (if EPRx = "1").

**Note:** The receive operation is terminated when the first stop bit is sampled even if the stop bit is configured with two bits.



### • Successive receive operations

When the data received in the shift register is transferred to the receive data register, RDBF<sub>x</sub> is set to "1" (buffer full), indicating that the received data can be read out. Thereafter, data can be received successively because the receive data register can be read out while the next data is received. The procedure for receiving is described above.

When RDBF<sub>x</sub> is set to "1", a receive-data full interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the received data can be read using an interrupt processing routine. In addition, since this interrupt factor can be used to invoke IDMA, the received data can be received successively in locations prepared in memory through DMA transfers.

For details on how to control interrupts and IDMA requests, refer to "Serial Interface Interrupts and DMA".

### (3) Receive errors

Three types of receive errors can be detected when receiving data in the asynchronous mode.

Since an interrupt can be generated by setting the interrupt controller, the error can be processed using an interrupt processing routine. For details on receive error interrupts, refer to "Serial Interface Interrupts and DMA".

#### • Parity error

If EPR<sub>x</sub> is set to "1" (parity added), the parity is checked when data is received.

This parity check is performed when the data received in the shift register is transferred to the receive data register in order to check conformity with PMD<sub>x</sub> settings (odd or even parity). If any nonconformity is found in this check, a parity error is assumed and the parity error flag is set to "1".

Ch.0 parity error flag: PER0 (D3) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 parity error flag: PER1 (D3) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 parity error flag: PER2 (D3) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 parity error flag: PER3 (D3) / Serial I/F Ch.3 status register (0x401F7)

Even when this error occurs, the received data in error is transferred to the receive data register and the receive operation is continued. However, the content of the received data for which a parity error is flagged cannot be guaranteed.

The PER<sub>x</sub> flag is reset to "0" by writing "0".

#### • Framing error

If data with a stop bit = "0" is received, the serial interface assumes that the data is out of synchronization and generates a framing error.

If two stop bits are used, only the first stop bit is checked.

When this error occurs, the framing-error flag is set to "1".

Ch.0 framing-error flag: FER0 (D4) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 framing-error flag: FER1 (D4) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 framing-error flag: FER2 (D4) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 framing-error flag: FER3 (D4) / Serial I/F Ch.3 status register (0x401F7)

Even when this error occurs, the received data in error is transferred to the receive data register and the receive operation is continued. However, the content of the received data for which a framing error is flagged cannot be guaranteed, even if no framing error is found in the following data received.

The FER<sub>x</sub> flag is reset to "0" by writing "0".

#### • Overrun error

If during successive receive operations, a receive operation for the next data is completed before the receive data register is read out, the receive data register is overwritten with the new data. Therefore, the receive data register must always be read out before a receive operation for the next data is completed.

When the receive data register is overwritten, an overrun error is generated and the overrun-error flag is set to "1".

### III-9 PERIPHERAL BLOCK: SERIAL INTERFACE

Ch.0 overrun-error flag: OER0 (D2) / Serial I/F Ch.0 status register (0x401E2)

Ch.1 overrun-error flag: OER1 (D2) / Serial I/F Ch.1 status register (0x401E7)

Ch.2 overrun-error flag: OER2 (D2) / Serial I/F Ch.2 status register (0x401F2)

Ch.3 overrun-error flag: OER3 (D2) / Serial I/F Ch.3 status register (0x401F7)

Even when this error occurs, the received data in error is transferred to the receive data register and the receive operation is continued.

The OERx flag is reset to "0" by writing "0".

#### (4) Terminating receive operation

When a data receive operation is completed, write "0" to the receive-enable bit RXENx to disable receive operations.

## IrDA Interface

### Outline of IrDA Interface

Each channel of the serial interface contains a RZI modulator circuit, allowing an infrared-ray communication circuit to be configured based on IrDA 1.0 simply by adding a simple external circuit.

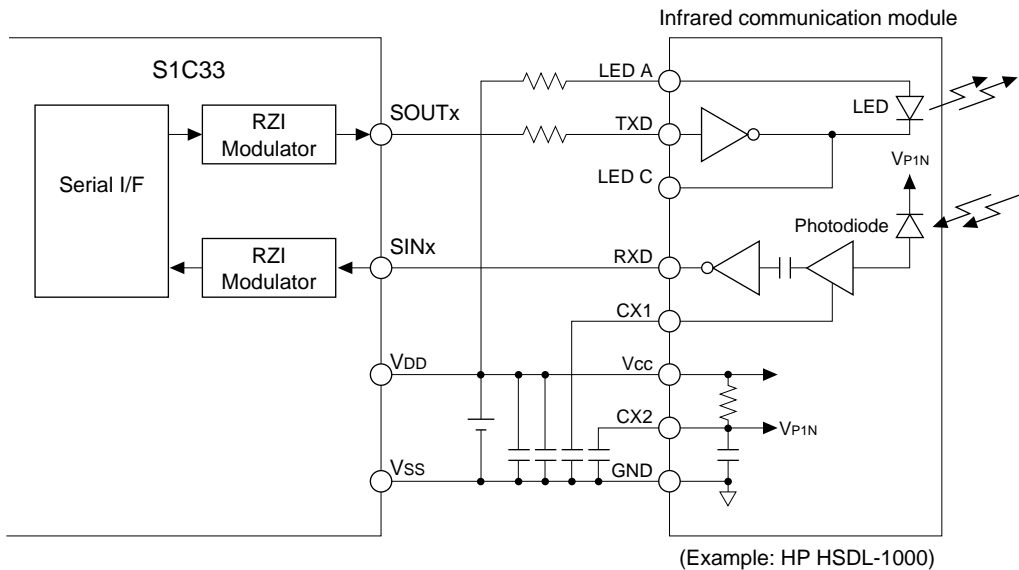


Figure III.9.14 Configuration Example of IrDA Interface

This IrDA interface function can be used only when the selected transfer mode is an asynchronous mode. Since the contents of the asynchronous mode are applied directly for the serial-interface functions other than the IrDA interface unit, refer to "Asynchronous Interface", for details on how to set and control the data formats and data transfers.

### Setting IrDA Interface

When performing infrared-ray communication, the following settings must be made before communication can be started:

1. Setting input/output pins
2. Selecting the interface mode (IrDA interface function)
3. Setting the transfer mode
4. Setting the input clock
5. Setting the data format
6. Setting the interrupt/IDMA/HSDMA
7. Setting the input/output logic

The contents for items 1 through 5 have been explained in connection with the asynchronous interface. For details, refer to "Asynchronous Interface". For details on item 6, refer to "Serial Interface Interrupts and DMA".

**Note:** Before making these settings, always make sure the serial interface is inactive (TXENx and RXENx are both set to "0"), as a change in settings during operation could cause a malfunction. In addition, be sure to set the transfer mode in (3) and the following items before selecting the IrDA interface function in (2).

### Selecting the IrDA interface function

To use the IrDA interface function, select it using the control bits shown below and then set the 8-bit (or 7-bit) asynchronous mode as the transfer mode.

Ch.0 IrDA interface-function selection: IRMD0[1:0] (D[1:0]) / Serial I/F Ch.0 IrDA register (0x401E4)

Ch.1 IrDA interface-function selection: IRMD1[1:0] (D[1:0]) / Serial I/F Ch.1 IrDA register (0x401E9)

Ch.2 IrDA interface-function selection: IRMD2[1:0] (D[1:0]) / Serial I/F Ch.2 IrDA register (0x401F4)

Ch.3 IrDA interface-function selection: IRMD3[1:0] (D[1:0]) / Serial I/F Ch.3 IrDA register (0x401F9)

Table III.9.7 Setting of IrDA Interface

IRMDx1	IRMDx0	Interface mode
1	1	Do not set. (reserved)
1	0	IrDA 1.0 interface
0	1	Do not set. (reserved)
0	0	Normal interface

**Note:** The IRMDx bit becomes indeterminate when initially reset, so be sure to initialize it in the software.

### Setting the input/output logic

When using the IrDA interface, the logic of the input/output signals of the RZI modulator circuit can be changed in accordance with the infrared-ray communication module or the circuit connected externally to the chip. The logic of the internal serial interface is "active-low". If the input/output signals are active-high, the logic of these signals must be inverted before they can be used. The input SINx and output SOUTx logic can be set individually through the use of the IRRLx and IRTLx bits, respectively.

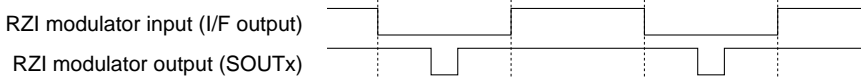
Table III.9.8 IrDA Input/Output Logic Inversion Bits

Item	Ch.0 (Serial I/F Ch.0 control register)	Ch.1 (Serial I/F Ch.1 control register)	Ch.2 (Serial I/F Ch.2 control register)	Ch.3 (Serial I/F Ch.3 control register)
IrDA input logic inversion	IRRL0(D2/0x401E4)	IRRL1(D2/0x401E9)	IRRL2(D2/0x401F4)	IRRL3(D2/0x401F9)
IrDA output logic inversion	IRTL0(D3/0x401E4)	IRTL1(D3/0x401E9)	IRTL2(D3/0x401F4)	IRTL3(D3/0x401F9)

The logic of the input/output signal is inverted by writing "1" to each corresponding bit. Logic is not inverted if the bit is set to "0".

When transmitting

(1) IRTLx = "0"

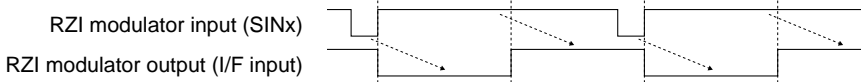


(2) IRTLx = "1"



When receiving

(1) IRRLx = "0"



(2) IRRLx = "1"

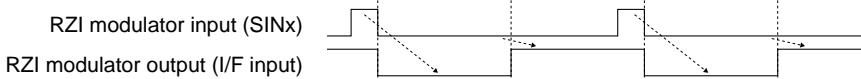


Figure III.9.15 IRRLx and IRTLx Settings

**Note:** The IRRLx and IRTLx bits become indeterminate at initial reset, so be sure to initialize them in the software.

### Control and Operation of IrDA Interface

The transmit/receive procedures have been explained in the section on the asynchronous interface, so refer to "Control and Operation of Asynchronous Transfer".

The following describes the data modulation and demodulation performed using the RZI modulator circuit:

#### When transmitting

During data transmission, the pulse width of the serial interface output signal is set to  $3/16$  before the signal is output from the SOUTx pin.

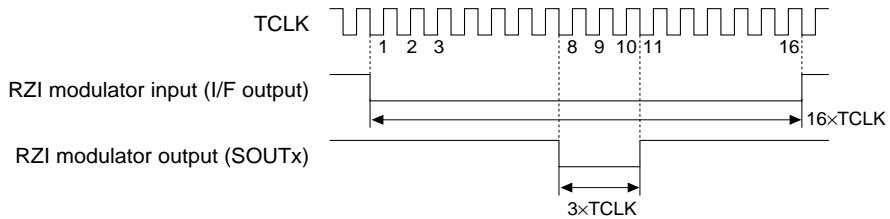


Figure III.9.16 Data Modulation by RZI Circuit

#### When receiving

During data reception, the pulse width of the input signal from SINx is set to  $16/3$  before the signal is transferred to the serial interface.

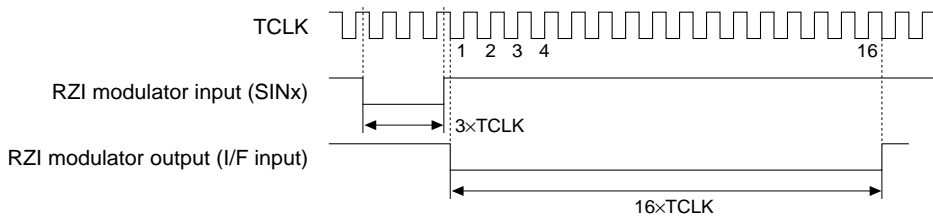


Figure III.9.17 Demodulation by RZI Circuit

- Notes:**
- When using the IrDA interface, set the internal division ratio of the serial interface  $1/16$  ( $DIVMDx = "1"$ ), rather than  $1/8$  ( $DIVMDx = "0"$ ).
  - Although Figure III.9.7 shows the input signal as a low pulse of a  $3xTCLK$  width, the RZI circuit recognizes low pulses by means of the signal edge (rising edge when  $IRRLx = "0"$ ; falling edge when  $IRRLx = "1"$ ). Note that noise may cause a malfunction.

## Serial Interface Interrupts and DMA

The serial interface can generate the following three types of interrupts in each channel:

- Transmit-buffer empty interrupt
- Receive-buffer full interrupt
- Receive-error interrupt

### Transmit-buffer empty interrupt factor

This interrupt factor occurs when the transmit data set in the transmit data register is transferred to the shift register, in which case the interrupt factor flag FSTXx is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated.

Occurrence of this interrupt factor indicates that the next transmit data can be written to the transmit data register.

This interrupt factor can also be used to invoke IDMA, enabling transmit data to be written to the register by means of a DMA transfer.

### Receive-completion interrupt

This interrupt factor occurs when a receive operation is completed and the receive data taken into the shift register is transferred to the receive data register, in which case the interrupt factor flag FSRXx is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated. Occurrence of this interrupt factor indicates that the received data can be read out.

This interrupt factor can also be used to invoke IDMA, enabling the received data to be written into specified memory locations by means of a DMA transfer.

### Receive-error interrupt

This interrupt factor occurs when a parity, framing, or overrun error is detected during data reception, in which case the interrupt factor flag FSERRx is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated.

Since all three types of errors generate the same interrupt factor, check the error flags PERx (parity error), OERx (overrun error), and FERx (framing error) to identify the type of error that has occurred. In the clock-synchronized mode, parity and framing errors do not occur.

**Note:** If a receive error (parity or framing error) occurs, the receive-error interrupt and receive-buffer full interrupt factors occur simultaneously. However, since the receive-error interrupt has priority over the receive-buffer full interrupt, the receive-error interrupt is processed first. It is therefore necessary for the receive-buffer full interrupt factor flag be cleared through the use of the receive-error interrupt processing routine.

### Control registers of the interrupt controller

Table III.9.9 shows the interrupt controller's control registers provided for each interrupt source (channel).

Table III.9.9 Control Register of Interrupt Controller

Channel	Interrupt factor	Interrupt factor flag	Interrupt enable register	Interrupt priority register
Ch.0	Receive-error interrupt	FSERR0(D0/0x40286)	ESERR0(D0/0x40276)	PSIO0[2:0](D[6:4]/0x40269)
	Receive-buffer full	FSRX0(D1/0x40286)	ESRX0(D1/0x40276)	
	Transmit-buffer empty	FSTX0(D2/0x40286)	ESTX0(D2/0x40276)	
Ch.1	Receive-error interrupt	FSERR1(D3/0x40286)	ESERR1(D3/0x40276)	PSIO1[2:0](D[2:0]/0x4026A)
	Receive-buffer full	FSRX1(D4/0x40286)	ESRX1(D4/0x40276)	
	Transmit-buffer empty	FSTX1(D5/0x40286)	ESTX1(D5/0x40276)	
Ch.2	Receive-error interrupt	FSERR2(D0/0x40289)	ESERR2(D0/0x40279)	PSIO2[2:0](D[2:0]/0x4026E)
	Receive-buffer full	FSRX2(D1/0x40289)	ESRX2(D1/0x40279)	
	Transmit-buffer empty	FSTX2(D2/0x40289)	ESTX2(D2/0x40279)	
Ch.3	Receive-error interrupt	FSERR3(D3/0x40289)	ESERR3(D3/0x40279)	PSIO3[2:0](D[6:4]/0x4026E)
	Receive-buffer full	FSRX3(D4/0x40289)	ESRX3(D4/0x40279)	
	Transmit-buffer empty	FSTX3(D5/0x40289)	ESTX3(D5/0x40279)	

When the interrupt factor described above occurs, the corresponding interrupt factor flag is set to "1". If the interrupt enable register bit for that interrupt factor has been set to "1", an interrupt request is generated.

Interrupts caused by an interrupt factor can be disabled by leaving the interrupt enable register bit for that factor set to "0". The interrupt factor flag is set to "1" whenever interrupt conditions are met, regardless of the setting of the interrupt enable register (even if it is set to "0").

The interrupt priority register sets the interrupt priority level of each interrupt source in a range between 0 and 7. An interrupt request to the CPU is accepted only when no other interrupt request of a higher priority has been generated. In addition, only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the input interrupt level set by the interrupt priority register, will the input interrupt request actually be accepted by the CPU.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to "ITC (Interrupt Controller)".

## Intelligent DMA

The receive-buffer full interrupt and transmit-buffer empty interrupt factors can be used to invoke intelligent DMA (IDMA). This enables successive transmit/receive operations between memory and the transmit/receive-buffer to be performed by means of a DAM transfer.

The following shows the IDMA channel numbers set for each interrupt factor:

IDMA Ch.

Ch.0 receive-buffer full interrupt: 0x17

Ch.0 transmit-buffer empty interrupt: 0x18

Ch.1 receive-buffer full interrupt: 0x19

Ch.1 transmit-buffer empty interrupt: 0x1A

Ch.2 receive-buffer full interrupt: 0x22

Ch.2 transmit-buffer empty interrupt: 0x23

Ch.3 receive-buffer full interrupt: 0x24

Ch.3 transmit-buffer empty interrupt: 0x25

The IDMA request and enable bits shown in Table III.9.10 must be set to "1" for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

Table III.9.10 Control Bits for IDMA Transfer

Channel	Interrupt factor	IDMA request bit	IDMA enable bit
Ch.0	Receive-buffer full	RSRX0(D6/0x40292)	DESRX0(D6/0x40296)
	Transmit-buffer empty	RSTX0(D7/0x40292)	DESTX0(D7/0x40296)
Ch.1	Receive-buffer full	RSRX1(D0/0x40293)	DESRX1(D0/0x40297)
	Transmit-buffer empty	RSTX1(D1/0x40293)	DESTX1(D1/0x40297)
Ch.2	Receive-buffer full	RSRX2(D2/0x4029B)	DESRX2(D2/0x4029C)
	Transmit-buffer empty	RSTX2(D3/0x4029B)	DESTX2(D3/0x4029C)
Ch.3	Receive-buffer full	RSRX3(D4/0x4029B)	DESRX3(D4/0x4029C)
	Transmit-buffer empty	RSTX3(D5/0x4029B)	DESTX3(D5/0x4029C)

If an interrupt factor occurs when the IDMA request and enable bits are set to "1", IDMA is invoked. No interrupt request is generated at that point. An interrupt request is generated upon completion of the DMA transfer. The bits can also be set so as not to generate an interrupt, with only a DAM transfer performed. For details on DMA transfer and how to control interrupts upon completion of DMA transfer, refer to "IDMA (Intelligent DMA)".

### High-speed DMA

The receive-buffer full interrupt and transmit-buffer empty interrupt factors for Ch.0 and Ch.1 can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit corresponding to each channel:

Table III.9.11 HSDMA Trigger Set-up Bits

SIF channel	HSDMA channel	Trigger set-up bits
0	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0/1 trigger set-up register (0x40298)
1	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298)
0	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299)
1	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299)

For HSDMA to be invoked by the receive-buffer full interrupt factor, the trigger set-up bits should be set to "1010". For HSDMA to be invoked by the transmit-buffer empty interrupt factor, the trigger set-up bits should be set to "1011". Transfer conditions, etc. must also be set on the HSDMA side.

The HSDMA channel is invoked through generation of the interrupt factor.

For details on HSDMA transfer, refer to "HSDMA (High-Speed DMA)".

### Trap vectors

The trap-vector address of each default interrupt factor is set as follows:

- Ch.0 receive-error interrupt: 0x0C000E0
- Ch.0 receive-buffer full interrupt: 0x0C000E4
- Ch.0 transmit-buffer empty interrupt: 0x0C000E8
- Ch.1 receive-error interrupt: 0x0C000F0
- Ch.1 receive-buffer full interrupt: 0x0C000F4
- Ch.1 transmit-buffer empty interrupt: 0x0C000F8
- Ch.2 receive-error interrupt: 0x0C00130
- Ch.2 receive-buffer full interrupt: 0x0C00134
- Ch.2 transmit-buffer empty interrupt: 0x0C00138
- Ch.3 receive-error interrupt: 0x0C00140
- Ch.3 receive-buffer full interrupt: 0x0C00144
- Ch.3 transmit-buffer empty interrupt: 0x0C00148

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).



## I/O Memory of Serial Interface

Table III.9.12 shows the control bits of the serial interface.

For details on the I/O memory of the prescaler that is used to set clocks, as well as that of 8-bit programmable timers, refer to "Prescaler" and "8-Bit Programmable Timers", respectively.

Table III.9.12 Control Bits of Serial Interface

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Serial I/F Ch.0 transmit data register	00401E0 (B)	D7	TXD07	Serial I/F Ch.0 transmit data TXD07(06) = MSB TXD00 = LSB	0x0 to 0xFF(0x7F)		X	R/W	7-bit asynchronous mode does not use TXD07.		
		D6	TXD06				X				
		D5	TXD05				X				
		D4	TXD04				X				
		D3	TXD03				X				
		D2	TXD02				X				
		D1	TXD01				X				
		D0	TXD00				X				
		Serial I/F Ch.0 receive data register	00401E1 (B)				D7	RXD07		Serial I/F Ch.0 receive data RXD07(06) = MSB RXD00 = LSB	0x0 to 0xFF(0x7F)
D6	RXD06			X							
D5	RXD05			X							
D4	RXD04			X							
D3	RXD03			X							
D2	RXD02			X							
D1	RXD01			X							
D0	RXD00			X							
Serial I/F Ch.0 status register	00401E2 (B)			D7-6	–	–	–		–		
		D5	TEND0	Ch.0 transmit-completion flag	1	Transmitting	0	End	0	R	
		D4	FER0	Ch.0 flaming error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	PER0	Ch.0 parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D2	OER0	Ch.0 overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D1	TDBE0	Ch.0 transmit data buffer empty	1	Empty	0	Buffer full	1	R	
		D0	RDBF0	Ch.0 receive data buffer full	1	Buffer full	0	Empty	0	R	
Serial I/F Ch.0 control register	00401E3 (B)	D7	TXEN0	Ch.0 transmit enable	1	Enabled	0	Disabled	0	R/W	
		D6	RXEN0	Ch.0 receive enable	1	Enabled	0	Disabled	0	R/W	
		D5	EPR0	Ch.0 parity enable	1	With parity	0	No parity	X	R/W	Valid only in asynchronous mode.
		D4	PMD0	Ch.0 parity mode selection	1	Odd	0	Even	X	R/W	
		D3	STPB0	Ch.0 stop bit selection	1	2 bits	0	1 bit	X	R/W	
		D2	SCK0	Ch.0 input clock selection	1	#SCLK0	0	Internal clock	X	R/W	
		D1	SMD01	Ch.0 transfer mode selection	SMD0[1:0]		Transfer mode		X	R/W	
		D0	SMD00		1	1	8-bit asynchronous	X			
		1	0		7-bit asynchronous						
		0	1		Clock sync. Slave						
			0	0	Clock sync. Master						
Serial I/F Ch.0 IrDA register	00401E4 (B)	D7-5	–	–	–		–	–	0 when being read.		
		D4	DIVMD0	Ch.0 async. clock division ratio	1	1/8	0	1/16	X	R/W	
		D3	IRTL0	Ch.0 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.
		D2	IRRL0	Ch.0 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W	
		D1	IRMD01	Ch.0 interface mode selection	IRMD0[1:0]		I/F mode		X	R/W	
		D0	IRMD00		1	1	reserved	X			
		1	0	IrDA 1.0							
		0	1	reserved							
		0	0	General I/F							
Serial I/F Ch.1 transmit data register	00401E5 (B)	D7	TXD17	Serial I/F Ch.1 transmit data TXD17(16) = MSB TXD10 = LSB	0x0 to 0xFF(0x7F)		X	R/W	7-bit asynchronous mode does not use TXD17.		
		D6	TXD16				X				
		D5	TXD15				X				
		D4	TXD14				X				
		D3	TXD13				X				
		D2	TXD12				X				
		D1	TXD11				X				
		D0	TXD10				X				
		Serial I/F Ch.1 receive data register	00401E6 (B)				D7	RXD17		Serial I/F Ch.1 receive data RXD17(16) = MSB RXD10 = LSB	0x0 to 0xFF(0x7F)
D6	RXD16			X							
D5	RXD15			X							
D4	RXD14			X							
D3	RXD13			X							
D2	RXD12			X							
D1	RXD11			X							
D0	RXD10			X							

III

SIF

### III-9 PERIPHERAL BLOCK: SERIAL INTERFACE

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Serial I/F Ch.1 status register	00401E7 (B)	D7-6	–	–	–	–	–	0 when being read.	
		D5	<b>TEND1</b>	Ch.1 transmit-completion flag	1 Transmitting 0 End	0	R		
		D4	<b>FER1</b>	Ch.1 flaming error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.	
		D3	<b>PER1</b>	Ch.1 parity error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.	
		D2	<b>OER1</b>	Ch.1 overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.	
		D1	<b>TDBE1</b>	Ch.1 transmit data buffer empty	1 Empty 0 Buffer full	1	R		
		D0	<b>RDBF1</b>	Ch.1 receive data buffer full	1 Buffer full 0 Empty	0	R		
Serial I/F Ch.1 control register	00401E8 (B)	D7	<b>TXEN1</b>	Ch.1 transmit enable	1 Enabled 0 Disabled	0	R/W		
		D6	<b>RXEN1</b>	Ch.1 receive enable	1 Enabled 0 Disabled	0	R/W		
		D5	<b>EPR1</b>	Ch.1 parity enable	1 With parity 0 No parity	X	R/W	Valid only in asynchronous mode.	
		D4	<b>PMD1</b>	Ch.1 parity mode selection	1 Odd 0 Even	X	R/W		
		D3	<b>STPB1</b>	Ch.1 stop bit selection	1 2 bits 0 1 bit	X	R/W		
		D2	<b>SSCK1</b>	Ch.1 input clock selection	1 #SCLK1 0 Internal clock	X	R/W		
		D1 D0	<b>SMD11</b> <b>SMD10</b>	Ch.1 transfer mode selection	SMD1[1:0] Transfer mode	X	R/W		
1 1	8-bit asynchronous	X							
1 0	7-bit asynchronous								
0 1	Clock sync. Slave								
0 0	Clock sync. Master								
Serial I/F Ch.1 IrDA register	00401E9 (B)	D7-5	–	–	–	–	–	0 when being read.	
		D4	<b>DIVMD1</b>	Ch.1 async. clock division ratio	1 1/8 0 1/16	X	R/W		
		D3	<b>IRTL1</b>	Ch.1 IrDA I/F output logic inversion	1 Inverted 0 Direct	X	R/W	Valid only in asynchronous mode.	
		D2	<b>IRRL1</b>	Ch.1 IrDA I/F input logic inversion	1 Inverted 0 Direct	X	R/W		
		D1 D0	<b>IRMD11</b> <b>IRMD10</b>	Ch.1 interface mode selection	IRMD1[1:0] I/F mode	X	R/W		
		1 1	reserved	X					
		1 0	IrDA 1.0						
0 1	reserved								
0 0	General I/F								
Serial I/F Ch.2 transmit data register	00401F0 (B)	D7	<b>TXD27</b>	Serial I/F Ch.2 transmit data	0x0 to 0xFF(0x7F)		X	R/W	
		D6	<b>TXD26</b>	TXD27(26) = MSB			X		
		D5	<b>TXD25</b>	TXD20 = LSB			X		
		D4	<b>TXD24</b>				X		
		D3	<b>TXD23</b>				X		
		D2	<b>TXD22</b>				X		
		D1	<b>TXD21</b>				X		
		D0	<b>TXD20</b>				X		
Serial I/F Ch.2 receive data register	00401F1 (B)	D7	<b>RXD27</b>	Serial I/F Ch.2 receive data	0x0 to 0xFF(0x7F)		X	R	
		D6	<b>RXD26</b>	RXD27(26) = MSB			X		
		D5	<b>RXD25</b>	RXD20 = LSB			X		
		D4	<b>RXD24</b>				X		
		D3	<b>RXD23</b>				X		
		D2	<b>RXD22</b>				X		
		D1	<b>RXD21</b>				X		
		D0	<b>RXD20</b>				X		
Serial I/F Ch.2 status register	00401F2 (B)	D7-6	–	reserved	–	–	–	0 when being read.	
		D5	<b>TEND2</b>	Ch.2 transmit-completion flag	1 Transmitting 0 End	0	R		
		D4	<b>FER2</b>	Ch.2 flaming error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.	
		D3	<b>PER2</b>	Ch.2 parity error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.	
		D2	<b>OER2</b>	Ch.2 overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.	
		D1	<b>TDBE2</b>	Ch.2 transmit data buffer empty	1 Empty 0 Buffer full	1	R		
		D0	<b>RDBF2</b>	Ch.2 receive data buffer full	1 Buffer full 0 Empty	0	R		
Serial I/F Ch.2 control register	00401F3 (B)	D7	<b>TXEN2</b>	Ch.2 transmit enable	1 Enabled 0 Disabled	0	R/W		
		D6	<b>RXEN2</b>	Ch.2 receive enable	1 Enabled 0 Disabled	0	R/W		
		D5	<b>EPR2</b>	Ch.2 parity enable	1 With parity 0 No parity	X	R/W	Valid only in asynchronous mode.	
		D4	<b>PMD2</b>	Ch.2 parity mode selection	1 Odd 0 Even	X	R/W		
		D3	<b>STPB2</b>	Ch.2 stop bit selection	1 2 bits 0 1 bit	X	R/W		
		D2	<b>SSCK2</b>	Ch.2 input clock selection	1 #SCLK2 0 Internal clock	X	R/W		
		D1 D0	<b>SMD21</b> <b>SMD20</b>	Ch.2 transfer mode selection	SMD2[1:0] Transfer mode	X	R/W		
1 1	8-bit asynchronous	X							
1 0	7-bit asynchronous								
0 1	Clock sync. Slave								
0 0	Clock sync. Master								
Serial I/F Ch.2 IrDA register	00401F4 (B)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4	<b>DIVMD2</b>	Ch.2 async. clock division ratio	1 1/8 0 1/16	X	R/W		
		D3	<b>IRTL2</b>	Ch.2 IrDA I/F output logic inversion	1 Inverted 0 Direct	X	R/W	Valid only in asynchronous mode.	
		D2	<b>IRRL2</b>	Ch.2 IrDA I/F input logic inversion	1 Inverted 0 Direct	X	R/W		
		D1 D0	<b>IRMD21</b> <b>IRMD20</b>	Ch.2 interface mode selection	IRMD2[1:0] I/F mode	X	R/W		
		1 1	reserved	X					
		1 0	IrDA 1.0						
0 1	reserved								
0 0	General I/F								

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Serial I/F Ch.3 transmit data register	00401F5 (B)	D7	TXD37	Serial I/F Ch.3 transmit data TXD37(36) = MSB TXD30 = LSB	0x0 to 0xFF(0x7F)		X	R/W			
		D6	TXD36				X				
		D5	TXD35				X				
		D4	TXD34				X				
		D3	TXD33				X				
		D2	TXD32				X				
		D1	TXD31				X				
		D0	TXD30				X				
Serial I/F Ch.3 receive data register	00401F6 (B)	D7	RXD37	Serial I/F Ch.3 receive data RXD37(36) = MSB RXD30 = LSB	0x0 to 0xFF(0x7F)		X	R			
		D6	RXD36				X				
		D5	RXD35				X				
		D4	RXD34				X				
		D3	RXD33				X				
		D2	RXD32				X				
		D1	RXD31				X				
		D0	RXD30				X				
Serial I/F Ch.3 status register	00401F7 (B)	D7-6	–	reserved	–		–	–	0 when being read.		
		D5	TEND3	Ch.3 transmit-completion flag	1	Transmitting	0	End	0	R	
		D4	FER3	Ch.3 flaming error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	PER3	Ch.3 parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D2	OER3	Ch.3 overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D1	TDBE3	Ch.3 transmit data buffer empty	1	Empty	0	Buffer full	1	R	
		D0	RDBF3	Ch.3 receive data buffer full	1	Buffer full	0	Empty	0	R	
Serial I/F Ch.3 control register	00401F8 (B)	D7	TXEN3	Ch.3 transmit enable	1	Enabled	0	Disabled	0	R/W	
		D6	RXEN3	Ch.3 receive enable	1	Enabled	0	Disabled	0	R/W	
		D5	EPR3	Ch.3 parity enable	1	With parity	0	No parity	X	R/W	Valid only in asynchronous mode.
		D4	PMD3	Ch.3 parity mode selection	1	Odd	0	Even	X	R/W	
		D3	STPB3	Ch.3 stop bit selection	1	2 bits	0	1 bit	X	R/W	
		D2	SSCK3	Ch.3 input clock selection	1	#SCLK3	0	Internal clock	X	R/W	
		D1	SMD31	Ch.3 transfer mode selection	SMD3[1:0]		Transfer mode		X	R/W	
		D0	SMD30		1	1	8-bit asynchronous		X		
1	0				7-bit asynchronous						
0	1				Clock sync. Slave						
0	0			Clock sync. Master							
Serial I/F Ch.3 IrDA register	00401F9 (B)	D7-5	–	reserved	–		–	–	0 when being read.		
		D4	DIVMD3	Ch.3 async. clock division ratio	1	1/8	0	1/16	X	R/W	
		D3	IRTL3	Ch.3 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.
		D2	IRRL3	Ch.3 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W	
		D1	IRMD31	Ch.3 interface mode selection	IRMD3[1:0]		I/F mode		X	R/W	
		D0	IRMD30		1	1	reserved		X		
1	0	IrDA 1.0									
0	1	reserved									
0	0	General I/F									
8-bit timer, serial I/F Ch.0 interrupt priority register	0040269 (B)	D7	–	reserved	–		–	–	0 when being read.		
		D6	PSIO02	Serial interface Ch.0 interrupt level	0 to 7		X	R/W			
		D5	PSIO01				X				
		D4	PSIO00				X				
		D3	–	reserved	–		–	–	0 when being read.		
		D2	P8TM2	8-bit timer 0-5 interrupt level	0 to 7		X	R/W			
D1	P8TM1	X									
D0	P8TM0	X									
Serial I/F Ch.1, A/D interrupt priority register	004026A (B)	D7	–	reserved	–		–	–	0 when being read.		
		D6	PAD2	A/D converter interrupt level	0 to 7		X	R/W			
		D5	PAD1				X				
		D4	PAD0				X				
		D3	–	reserved	–		–	–	0 when being read.		
		D2	PSIO12	Serial interface Ch.1 interrupt level	0 to 7		X	R/W			
D1	PSIO11	X									
D0	PSIO10	X									
Serial I/F Ch.2/3 interrupt priority register	004026E (B)	D7	–	reserved	–		–	–	0 when being read.		
		D6	PSIO32	Serial interface Ch.3 interrupt level	0 to 7		X	R/W			
		D5	PSIO31				X				
		D4	PSIO30				X				
		D3	–	reserved	–		–	–	0 when being read.		
		D2	PSIO22	Serial interface Ch.2 interrupt level	0 to 7		X	R/W			
D1	PSIO21	X									
D0	PSIO20	X									

### III-9 PERIPHERAL BLOCK: SERIAL INTERFACE

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.0/1 interrupt enable register	0040276 (B)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	ESTX	SIF Ch.1 transmit buffer empty	1 Enabled	0 Disabled	0 R/W	
		D4	ESRX1	SIF Ch.1 receive buffer full			0 R/W	
		D3	ESERR1	SIF Ch.1 receive error			0 R/W	
		D2	ESTX0	SIF Ch.0 transmit buffer empty			0 R/W	
		D1	ESRX0	SIF Ch.0 receive buffer full			0 R/W	
		D0	ESERR0	SIF Ch.0 receive error			0 R/W	
Serial I/F Ch.2/3 interrupt enable register	0040279 (B)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	ESTX3	SIF Ch.3 transmit buffer empty	1 Enabled	0 Disabled	0 R/W	
		D4	ESRX3	SIF Ch.3 receive buffer full			0 R/W	
		D3	ESERR3	SIF Ch.3 receive error			0 R/W	
		D2	ESTX2	SIF Ch.2 transmit buffer empty			0 R/W	
		D1	ESRX2	SIF Ch.2 receive buffer full			0 R/W	
		D0	ESERR2	SIF Ch.2 receive error			0 R/W	
Serial I/F Ch.0/1 interrupt factor flag register	0040286 (B)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	FSTX1	SIF Ch.1 transmit buffer empty	1 Factor is generated	0 No factor is generated	X R/W	
		D4	FSRX1	SIF Ch.1 receive buffer full			X R/W	
		D3	FSERR1	SIF Ch.1 receive error			X R/W	
		D2	FSTX0	SIF Ch.0 transmit buffer empty			X R/W	
		D1	FSRX0	SIF Ch.0 receive buffer full			X R/W	
		D0	FSERR0	SIF Ch.0 receive error			X R/W	
Serial I/F Ch.2/3 interrupt factor flag register	0040289 (B)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	FSTX3	SIF Ch.3 transmit buffer empty	1 Factor is generated	0 No factor is generated	X R/W	
		D4	FSRX3	SIF Ch.3 receive buffer full			X R/W	
		D3	FSERR3	SIF Ch.3 receive error			X R/W	
		D2	FSTX2	SIF Ch.2 transmit buffer empty			X R/W	
		D1	FSRX2	SIF Ch.2 receive buffer full			X R/W	
		D0	FSERR2	SIF Ch.2 receive error			X R/W	
16-bit timer 5, 8-bit timer 0-3, serial I/F Ch.0 IDMA request register	0040292 (B)	D7	RSTX0	SIF Ch.0 transmit buffer empty	1 IDMA request	0 Interrupt request	0 R/W	
		D6	RSRX0	SIF Ch.0 receive buffer full			0 R/W	
		D5	R8TU3	8-bit timer 3 underflow			0 R/W	
		D4	R8TU2	8-bit timer 2 underflow			0 R/W	
		D3	R8TU1	8-bit timer 1 underflow			0 R/W	
		D2	R8TU0	8-bit timer 0 underflow			0 R/W	
		D1	R16TC5	16-bit timer 5 comparison A			0 R/W	
		D0	R16TU5	16-bit timer 5 comparison B			0 R/W	
Serial I/F Ch.1, A/D, port input 4-7 IDMA request register	0040293 (B)	D7	RP7	Port input 7	1 IDMA request	0 Interrupt request	0 R/W	
		D6	RP6	Port input 6			0 R/W	
		D5	RP5	Port input 5			0 R/W	
		D4	RP4	Port input 4			0 R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	RADE	A/D converter	1 IDMA request	0 Interrupt request	0 R/W	
		D1	RSTX1	SIF Ch.1 transmit buffer empty			0 R/W	
		D0	RSRX1	SIF Ch.1 receive buffer full			0 R/W	
16-bit timer 5, 8-bit timer 0-3, serial I/F Ch.0 IDMA enable register	0040296 (B)	D7	DESTX0	SIF Ch.0 transmit buffer empty	1 IDMA enabled	0 IDMA disabled	0 R/W	
		D6	DESRX0	SIF Ch.0 receive buffer full			0 R/W	
		D5	DE8TU3	8-bit timer 3 underflow			0 R/W	
		D4	DE8TU2	8-bit timer 2 underflow			0 R/W	
		D3	DE8TU1	8-bit timer 1 underflow			0 R/W	
		D2	DE8TU0	8-bit timer 0 underflow			0 R/W	
		D1	DE16TC5	16-bit timer 5 comparison A			0 R/W	
		D0	DE16TU5	16-bit timer 5 comparison B			0 R/W	
Serial I/F Ch.1, A/D, port input 4-7 IDMA enable register	0040297 (B)	D7	DEP7	Port input 7	1 IDMA enabled	0 IDMA disabled	0 R/W	
		D6	DEP6	Port input 6			0 R/W	
		D5	DEP5	Port input 5			0 R/W	
		D4	DEP4	Port input 4			0 R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	DEADE	A/D converter	1 IDMA enabled	0 IDMA disabled	0 R/W	
		D1	DESTX1	SIF Ch.1 transmit buffer empty			0 R/W	
		D0	DESRX1	SIF Ch.1 receive buffer full			0 R/W	
8-bit timer 4/5 serial I/F Ch.2/3 IDMA request register	004029B (B)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	RSTX3	SIF Ch.3 transmit buffer empty	1 IDMA request	0 Interrupt request	0 R/W	
		D4	RSRX3	SIF Ch.3 receive buffer full			0 R/W	
		D3	RSTX2	SIF Ch.2 transmit buffer empty			0 R/W	
		D2	RSRX2	SIF Ch.2 receive buffer full			0 R/W	
		D1	R8TU5	8-bit timer 5 underflow			0 R/W	
		D0	R8TU4	8-bit timer 4 underflow			0 R/W	

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
8-bit timer 4/5 serial I/F Ch.2/3 IDMA enable register	004029C (B)	D7-6	–	reserved		–	–	–	0 when being read.		
		D5	DESTX3	SIF Ch.3 transmit buffer empty	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D4	DESRX3	SIF Ch.3 receive buffer full					0	R/W	
		D3	DESTX2	SIF Ch.2 transmit buffer empty					0	R/W	
		D2	DESRX2	SIF Ch.2 receive buffer full					0	R/W	
		D1	DE8TU5	8-bit timer 5 underflow					0	R/W	
		D0	DE8TU4	8-bit timer 4 underflow					0	R/W	
P0 function select register	00402D0 (B)	D7	CFP07	P07 function selection	1	#SRDY1	0	P07	0	R/W	Extended functions (0x402DF)
		D6	CFP06	P06 function selection	1	#SCLK1	0	P06	0	R/W	
		D5	CFP05	P05 function selection	1	SOUT1	0	P05	0	R/W	
		D4	CFP04	P04 function selection	1	SIN1	0	P04	0	R/W	
		D3	CFP03	P03 function selection	1	#SRDY0	0	P03	0	R/W	Extended functions (0x300040)
		D2	CFP02	P02 function selection	1	#SCLK0	0	P02	0	R/W	
		D1	CFP01	P01 function selection	1	SOUT0	0	P01	0	R/W	
		D0	CFP00	P00 function selection	1	SIN0	0	P00	0	R/W	
Port SIO function extension register	00402D7 (B)	D7-4	–	reserved		–	–	–	–	0 when being read.	
		D3	SSRDY3	Serial I/F Ch.3 SRDY selection	1	#SRDY3	0	P32/ #DMAACK0	0	R/W	
		D2	SSCLK3	Serial I/F Ch.3 SCLK selection	1	#SCLK3	0	P15/EXCL4/ #DMAEND0	0	R/W	
		D1	SSOUT3	Serial I/F Ch.3 SOUT selection	1	SOUT3	0	P16/EXCL5/ #DMAEND1	0	R/W	
		D0	SSIN3	Serial I/F Ch.3 SIN selection	1	SIN3	0	P33/ #DMAACK1	0	R/W	
Port SIO function extension register	00402DB (B)	D7-4	–	reserved		–	–	–	–	0 when being read.	
		D3	SSRDY2	Serial I/F Ch.2 SRDY selection	1	#SRDY2	0	P24/TM2	0	R/W	
		D2	SSCLK2	Serial I/F Ch.2 SCLK selection	1	#SCLK2	0	P25/TM3	0	R/W	
		D1	SSOUT2	Serial I/F Ch.2 SOUT selection	1	SOUT2	0	P26/TM4	0	R/W	
		D0	SSIN2	Serial I/F Ch.2 SIN selection	1	SIN2	0	P27/TM5	0	R/W	
Port function extension register	00402DF (B)	D7	CFEX7	P07 port extended function	1	#DMAEND3	0	P07, etc.	0	R/W	
		D6	CFEX6	P06 port extended function	1	#DMAACK3	0	P06, etc.	0	R/W	
		D5	CFEX5	P05 port extended function	1	#DMAEND2	0	P05, etc.	0	R/W	
		D4	CFEX4	P04 port extended function	1	#DMAACK2	0	P04, etc.	0	R/W	
		D3	CFEX3	P31 port extended function	1	#GARD	0	P31, etc.	0	R/W	
		D2	CFEX2	P21 port extended function	1	#GAAS	0	P21, etc.	0	R/W	
		D1	CFEX1	P10, P11, P13 port extended function	1	DST0 DST1 DPCO	0	P10, etc. P11, etc. P13, etc.	1	R/W	
		D0	CFEX0	P12, P14 port extended function	1	DST2 DCLK	0	P12, etc. P14, etc.	1	R/W	

---

**CFP07–CFP00:** P0[7:0] pin function selection (D[7:0]) / P0 function select register (0x402D0)

---

Selects the pins used for the serial interface.

Write "1": Serial-interface input/output pin  
 Write "0": I/O port pin  
 Read: Valid

Select the pins used for the serial interface from among P00 through P07 by writing "1" to CFP00 through CFP07. P00–P03 (SIN0, SOUT0, #SCLK0, #SRDY0) are used for channel 0; P04–P07 (SIN1, SOUT1, #SCLK1, #SRDY1) are used for channel 1. If the bit for a pin is set to "0", the pin functions as an I/O port. The necessary input/output pins differ depending on the transfer mode set (see Table III.9.3). At cold start, CFP is set to "0" (I/O port). At hot start, CFP retains its state from prior to the initial reset.

---

**CFEX7–CFEX4:** P0[7:4] pin function selection (D[7:4]) / Port function extension register (0x402DF)

---

Selects the extended function of pins P07–P04.

Write "1": Function-extended pin  
 Write "0": I/O-port/serial I/O pin  
 Read: Valid

When CFEX[7:4] is set to "1", the P07–P04 ports function as DMA signal output ports. When CFEX[7:4] = "0", the CFP0[7:4] bit becomes effective, so the settings of these bits determine whether the P07–P04 ports function as I/O ports or serial interface Ch.1 signal output ports. At cold start, CFEX[7:4] is set to "0" (I/O-port/serial I/O pin). At hot start, CFEX[7:4] retains its state from prior to the initial reset.

---

**SSIN2:** Serial I/F Ch.2 SIN selection (D0) / Port SIO function extension register (0x402DB)

---

Switches the function of pin P27/TM5/SIN2.

Write "1": SIN2  
 Write "0": P27/TM5  
 Read: Valid

To use the pin as SIN2, set SSIN2 (D0 / 0x402DB) to "1" and CFP27 (D7 / 0x402D8) to "0".  
 To use the pin as P27 or TM5, set this bit to "0".  
 At power-on, this bit is set to "0".

---

**SSOUT2:** Serial I/F Ch.2 SOUT selection (D1) / Port SIO function extension register (0x402DB)

---

Switches the function of pin P26/TM4/SOUT2.

Write "1": SOUT2  
 Write "0": P26/TM4  
 Read: Valid

To use the pin as SOUT2, set SSOUT2 (D1 / 0x402DB) to "1" and CFP26 (D6 / 0x402D8) to "0".  
 To use the pin as P26 or TM4, set this bit to "0".  
 At power-on, this bit is set to "0".

---

**SSCLK2:** Serial I/F Ch.2 SCLK selection (D2) / Port SIO function extension register (0x402DB)

---

Switches the function of pin P25/TM3/#SCLK2.

Write "1": #SCLK2  
 Write "0": P25/TM3  
 Read: Valid

To use the pin as #SCLK2, set SSCLK2 (D2 / 0x402DB) to "1" and CFP25 (D5 / 0x402D8) to "0".  
 To use the pin as P25 or TM3, set this bit to "0".  
 At power-on, this bit is set to "0".

**SSRDY2: Serial I/F Ch.2 SRDY selection (D3) / Port SIO function extension register (0x402DB)**

Switches the function of pin P24/TM2/#SRDY2.

Write "1": #SRDY2

Write "0": P24/TM2

Read: Valid

To use the pin as #SRDY2, set SSRDY2 (D3 / 0x402DB) to "1" and CFP24 (D4 / 0x402D8) to "0".

To use the pin as P24 or TM2, set this bit to "0".

At power-on, this bit is set to "0".

**SSIN3: Serial I/F Ch.3 SIN selection (D0) / Port SIO function extension register (0x402D7)**

Switches the function of pin P33/#DMAACK1/SIN3.

Write "1": SIN3

Write "0": P33/#DMAACK1

Read: Valid

To use the pin as SIN3, set SSIN3 (D0 / 0x402D7) to "1" and CFP33 (D3 / 0x402DC) to "0".

To use the pin as P33 or #DMAACK1, set this bit to "0".

At power-on, this bit is set to "0".

**SSOUT3: Serial I/F Ch.3 SOUT selection (D1) / Port SIO function extension register (0x402D7)**

Switches the function of pin P16/EXCL5/#DMAEND1/SOUT3.

Write "1": SOUT3

Write "0": P16/EXCL5/#DMAEND1

Read: Valid

To use the pin as SOUT3, set SSOUT3 (D1 / 0x402D7) to "1" and CFP16 (D6 / 0x402D4) to "0".

To use the pin as P16, EXCL5, or #DMAEND1, set this bit to "0".

At power-on, this bit is set to "0".

**SSCLK3: Serial I/F Ch.3 SCLK selection (D2) / Port SIO function extension register (0x402D7)**

Switches the function of pin P15/EXCL4/#DMAEND0/#SCLK3.

Write "1": #SCLK3

Write "0": P15/EXCL4/#DMAEND0

Read: Valid

To use the pin as #SCLK3, set SSCLK3 (D2 / 0x402D7) to "1" and CFP15 (D5 / 0x402D4) to "0".

To use the pin as P15, EXCL4, or #DMAEND0, set this bit to "0".

At power-on, this bit is set to "0".

**SSRDY3: Serial I/F Ch.3 SRDY selection (D3) / Port SIO function extension register (0x402D7)**

Switches the function of pin P32/#DMAACK0/#SRDY3.

Write "1": #SRDY3

Write "0": P32/#DMAACK0

Read: Valid

To use the pin as #SRDY3, set SSRDY3 (D3 / 0x402D7) to "1" and CFP32 (D2 / 0x402DC) to "0".

To use the pin as P32 or #DMAACK0, set this bit to "0".

At power-on, this bit is set to "0".

---

**TXD07–TXD00:** Ch.0 transmit data (D[7:0]) / Serial I/F Ch.0 transmit data register (0x401E0)

**TXD17–TXD10:** Ch.1 transmit data (D[7:0]) / Serial I/F Ch.1 transmit data register (0x401E5)

**TXD27–TXD20:** Ch.2 transmit data (D[7:0]) / Serial I/F Ch.2 transmit data register (0x401F0)

**TXD37–TXD30:** Ch.3 transmit data (D[7:0]) / Serial I/F Ch.3 transmit data register (0x401F5)

---

Sets transmit data.

When data is written to this register (transmit buffer) after "1" is written to TXEN<sub>x</sub>, a transmit operation is begun. TDBEx is set to "1" (transmit-buffer empty) when the data is transferred to the shift register. A transmit-buffer empty interrupt factor is simultaneously generated. The next transmit data can be written to the buffer at any time thereafter, even when the serial interface is sending data.

In the 7-bit asynchronous mode, TXD<sub>x</sub>7 (MSB) is ignored.

The serial-converted data is output from the SOUT pin beginning with the LSB, in which the bits set to "1" are output as high-level signals and those set to "0" output as low-level signals.

This register can be read as well as written.

At initial reset, the content of TXD<sub>x</sub> becomes indeterminate.

---

**RXD07–RXD00:** Ch.0 receive data (D[7:0]) / Serial I/F Ch.0 receive data register (0x401E1)

**RXD17–RXD10:** Ch.1 receive data (D[7:0]) / Serial I/F Ch.1 receive data register (0x401E6)

**RXD27–RXD20:** Ch.2 receive data (D[7:0]) / Serial I/F Ch.2 receive data register (0x401F1)

**RXD37–RXD30:** Ch.3 receive data (D[7:0]) / Serial I/F Ch.3 receive data register (0x401F6)

---

Stores received data.

When a receive operation is completed and the data received in the shift register is transferred to this register (receive buffer), RDBF<sub>x</sub> is set to "1" (receive buffer full). At the same time, a receive-buffer full interrupt factor is generated. Thereafter, the data can be read out at any time before a receive operation for the next data is completed. If the next data receive operation is completed before this register is read out, the data in it is overwritten with the newly received data, causing an overrun error to occur.

In the 7-bit asynchronous mode, "0" is stored in RXD<sub>x</sub>7.

The serial data input from the SIN<sub>x</sub> pin is converted into parallel data beginning with the LSB, with the high-level signals changed to "1"s and the low-level signals changed to "0"s. The resulting data is stored in this buffer.

This register is a read-only register, so no data can be written to it.

At initial reset, the content of RXD<sub>x</sub> becomes indeterminate.

---

**TEND0:** Ch.0 transmit-completion flag (D5) / Serial I/F Ch.0 status register (0x401E2)

**TEND1:** Ch.1 transmit-completion flag (D5) / Serial I/F Ch.1 status register (0x401E7)

**TEND2:** Ch.2 transmit-completion flag (D5) / Serial I/F Ch.2 status register (0x401F2)

**TEND3:** Ch.3 transmit-completion flag (D5) / Serial I/F Ch.3 status register (0x401F7)

---

Indicates the transmission status.

Read "1": During transmitting

Read "0": End of transmission

Write: Invalid

TEND<sub>x</sub> goes "1" when data is being transmitted and goes "0" when the transmission has completed.

When data is transmitted successively in clock-synchronized master mode or asynchronous mode, TEND<sub>x</sub> maintains "1" until all data is transmitted (see Figure III.9.4 and Figure III.9.12). In clock-synchronized slave mode, TEND<sub>x</sub> goes "0" every time 1-byte data is transmitted (see Figure III.9.5).

At initial reset, TEND<sub>x</sub> is set to "0" (End of transmission).



---

**FER0:** Ch.0 framing-error flag (D4) / Serial I/F Ch.0 status register (0x401E2)

**FER1:** Ch.1 framing-error flag (D4) / Serial I/F Ch.1 status register (0x401E7)

**FER2:** Ch.2 framing-error flag (D4) / Serial I/F Ch.2 status register (0x401F2)

**FER3:** Ch.3 framing-error flag (D4) / Serial I/F Ch.3 status register (0x401F7)

---

Indicates whether a framing error occurred.

Read "1": An error occurred

Read "0": No error occurred

Write "1": Invalid

Write "0": Reset to "0"

The FERx flag is an error flag indicating whether a framing error occurred. When an error has occurred, it is set to "1". A framing error occurs when data with a stop bit = "0" is received in the asynchronous mode.

The FERx flag is reset by writing "0".

At initial reset, as well as when RXENx and TXENx both are set to "0", the FERx flag is set to "0" (no error).

---

**PER0:** Ch.0 parity-error flag (D3) / Serial I/F Ch.0 status register (0x401E2)

**PER1:** Ch.1 parity-error flag (D3) / Serial I/F Ch.1 status register (0x401E7)

**PER2:** Ch.2 parity-error flag (D3) / Serial I/F Ch.2 status register (0x401F2)

**PER3:** Ch.3 parity-error flag (D3) / Serial I/F Ch.3 status register (0x401F7)

---

Indicates whether a parity error occurred.

Read "1": An error occurred

Read "0": No error occurred

Write "1": Invalid

Write "0": Reset to "0"

The PERx flag is an error flag indicating whether a parity error occurred. When an error has occurred, it is set to "1". Parity checks are valid only in the asynchronous mode with EPRx set to "1" (parity added). This check is performed when the received data is transferred from the shift register to the receive data register.

The PERx flag is reset by writing "0".

At initial reset, as well as when RXENx and TXENx both are set to "0", PERx is set to "0" (no error).

---

**OER0:** Ch.0 overrun-error flag (D2) / Serial I/F Ch.0 status register (0x401E2)

**OER1:** Ch.1 overrun-error flag (D2) / Serial I/F Ch.1 status register (0x401E7)

**OER2:** Ch.2 overrun-error flag (D2) / Serial I/F Ch.2 status register (0x401F2)

**OER3:** Ch.3 overrun-error flag (D2) / Serial I/F Ch.3 status register (0x401F7)

---

Indicates whether an overrun error occurred.

Read "1": An error occurred

Read "0": No error occurred

Write "1": Invalid

Write "0": Reset to "0"

The OERx flag is an error flag indicating whether an overrun error occurred. When an error has occurred, it is set to "1". An overrun error occurs when the next receive operation is completed before the receive data register is read out, resulting in the receive data register being overwritten.

The OERx flag is reset by writing "0".

At initial reset, as well as when RXENx and TXENx both are set to "0", OERx is set to "0" (no error).

---

**TDBE0:** Ch.0 transmit data buffer empty (D1) / Serial I/F Ch.0 status register (0x401E2)

**TDBE1:** Ch.1 transmit data buffer empty (D1) / Serial I/F Ch.1 status register (0x401E7)

**TDBE2:** Ch.2 transmit data buffer empty (D1) / Serial I/F Ch.2 status register (0x401F2)

**TDBE3:** Ch.3 transmit data buffer empty (D1) / Serial I/F Ch.3 status register (0x401F7)

---

Indicates the status of the transmit data register (buffer).

Read "1": Buffer empty

Read "0": Buffer full

Write: Invalid

TDBEx is set to "0" when transmit data is written to the transmit data register, and is set to "1" when this data is transferred to the shift register (transmit operation started).

Transmit data is written to the transmit data register when this bit = "1".

At initial reset, TDBEx is set to "1" (buffer empty).

---

**RDBF0:** Ch.0 receive data buffer full (D0) / Serial I/F Ch.0 status register (0x401E2)

**RDBF1:** Ch.1 receive data buffer full (D0) / Serial I/F Ch.1 status register (0x401E7)

**RDBF2:** Ch.2 receive data buffer full (D0) / Serial I/F Ch.2 status register (0x401F2)

**RDBF3:** Ch.3 receive data buffer full (D0) / Serial I/F Ch.3 status register (0x401F7)

---

Indicates the status of the receive data register (buffer).

Read "1": Buffer full

Read "0": Buffer empty

Write: Invalid

RDBFx is set to "1" when the data received in the shift register is transferred to the receive data register (receive operation completed), indicating that the received data can be read out. This bit is reset to "0" when the data is read out.

At initial reset, RDBFx is set to "0" (buffer empty).

---

**TXEN0:** Ch.0 transmit enable (D7) / Serial I/F Ch.0 control register (0x401E3)

**TXEN1:** Ch.1 transmit enable (D7) / Serial I/F Ch.1 control register (0x401E8)

**TXEN2:** Ch.2 transmit enable (D7) / Serial I/F Ch.2 control register (0x401F3)

**TXEN3:** Ch.3 transmit enable (D7) / Serial I/F Ch.3 control register (0x401F8)

---

Enables each channel for transmit operations.

Write "1": Transmit enabled

Write "0": Transmit disabled

Read: Valid

When TXENx for a channel is set to "1", the channel is enabled for transmit operations. When TXENx is set to "0", the channel is disabled for transmit operations.

Always make sure the TXENx = "0" before setting the transfer mode and other conditions.

At initial reset, TXENx is set to "0" (transmit disabled).

---

**RXEN0:** Ch.0 receive enable (D6) / Serial I/F Ch.0 control register (0x401E3)

**RXEN1:** Ch.1 receive enable (D6) / Serial I/F Ch.1 control register (0x401E8)

**RXEN2:** Ch.2 receive enable (D6) / Serial I/F Ch.2 control register (0x401F3)

**RXEN3:** Ch.3 receive enable (D6) / Serial I/F Ch.3 control register (0x401F8)

---

Enables each channel for receive operations.

Write "1": Receive enabled

Write "0": Receive disabled

Read: Valid

When RXENx for a channel is set to "1", the channel is enabled for receive operations. When RXENx is set to "0", the channel is disabled for receive operations.

Always make sure the RXENx = "0" before setting the transfer mode and other conditions.

At initial reset, RXENx is set to "0" (receive disabled).

---

**EPR0:** Ch.0 parity enable (D5) / Serial I/F Ch.0 control register (0x401E3)

**EPR1:** Ch.1 parity enable (D5) / Serial I/F Ch.1 control register (0x401E8)

**EPR2:** Ch.2 parity enable (D5) / Serial I/F Ch.2 control register (0x401F3)

**EPR3:** Ch.3 parity enable (D5) / Serial I/F Ch.3 control register (0x401F8)

---

Selects a parity function.

Write "1": Parity added

Write "0": No parity added

Read: Valid

EPRx is used to select whether receive data is to be checked for parity, and whether a parity bit is to be added to transmit data. When EPRx is set to "1", the receive data is checked for parity. A parity bit is automatically added to the transmit data. When EPRx is set to "0", parity is not checked and no parity bit is added.

The parity function is only valid in the asynchronous mode. Settings of EPRx have no effect in the clock-synchronized mode.

At initial reset, EPRx becomes indeterminate.

---

**PMD0:** Ch.0 parity mode selection (D4) / Serial I/F Ch.0 control register (0x401E3)

**PMD1:** Ch.1 parity mode selection (D4) / Serial I/F Ch.1 control register (0x401E8)

**PMD2:** Ch.2 parity mode selection (D4) / Serial I/F Ch.2 control register (0x401F3)

**PMD3:** Ch.3 parity mode selection (D4) / Serial I/F Ch.3 control register (0x401F8)

---

Selects an odd or even parity.

Write "1": Odd parity

Write "0": Even parity

Read: Valid

Odd parity is selected by writing "1" to PMDx, and even parity is selected by writing "0". Parity check and the addition of a parity bit are only effective in asynchronous transfers in which EPRx is set to "1". If EPRx = "0", settings of PMDx do not have any effect.

At initial reset, PMDx becomes indeterminate.

---

**STPB0:** Ch.0 stop bit selection (D3) / Serial I/F Ch.0 control register (0x401E3)

**STPB1:** Ch.1 stop bit selection (D3) / Serial I/F Ch.1 control register (0x401E8)

**STPB2:** Ch.2 stop bit selection (D3) / Serial I/F Ch.2 control register (0x401F3)

**STPB3:** Ch.3 stop bit selection (D3) / Serial I/F Ch.3 control register (0x401F8)

---

Selects a stop-bit length during the performance of an asynchronous transfer.

Write "1": 2 bits

Write "0": 1 bit

Read: Valid

STPBx is only valid in an asynchronous transfer. Two stop bits are selected by writing "1" to STPBx, and one stop bit is selected by writing "0". The start bit is fixed at 1 bit.

Settings of STPBx are ignored during the performance of a clock-synchronized transfer.

At initial reset, STPBx becomes indeterminate.

---

**SSCK0:** Ch.0 input clock selection (D2) / Serial I/F Ch.0 control register (0x401E3)

**SSCK1:** Ch.1 input clock selection (D2) / Serial I/F Ch.1 control register (0x401E8)

**SSCK2:** Ch.2 input clock selection (D2) / Serial I/F Ch.2 control register (0x401F3)

**SSCK3:** Ch.3 input clock selection (D2) / Serial I/F Ch.3 control register (0x401F8)

---

Selects the clock source for an asynchronous transfer.

Write "1": #SCLK (external clock)

Write "0": Internal clock

Read: Valid

During operation in the asynchronous mode, this bit is used to select the clock source between an internal clock (output by an 8-bit programmable timer) and an external clock (input from the #SCLKx pin). An external clock is selected by writing "1" to this bit, and an internal clock is selected by writing "0".

At initial reset, SSCKx becomes indeterminate.

### III-9 PERIPHERAL BLOCK: SERIAL INTERFACE

---

**SMD01–SMD00:** Ch.0 transfer mode selection (D[1:0]) / Serial I/F Ch.0 control register (0x401E3)

**SMD11–SMD10:** Ch.1 transfer mode selection (D[1:0]) / Serial I/F Ch.1 control register (0x401E8)

**SMD21–SMD20:** Ch.2 transfer mode selection (D[1:0]) / Serial I/F Ch.2 control register (0x401F3)

**SMD31–SMD30:** Ch.3 transfer mode selection (D[1:0]) / Serial I/F Ch.3 control register (0x401F8)

---

Sets the transfer mode of the serial interface as shown in Table III.9.14 below.

Table III.9.14 Setting of Transfer Mode

SMDx1	SMDx0	Transfer mode
1	1	8-bit asynchronous mode
1	0	7-bit asynchronous mode
0	1	Clock-synchronized slave mode
0	0	Clock-synchronized master mode

The SMDx bit can be read as well as written.

When using the IrDA interface, always be sure to set an asynchronous mode for the transfer mode.

At initial reset, SMDx becomes indeterminate.

---

**DIVMD0:** Sampling clock division ratio (D4) / Serial I/F Ch.0 IrDA register (0x401E4)

**DIVMD1:** Sampling clock division ratio (D4) / Serial I/F Ch.1 IrDA register (0x401E9)

**DIVMD2:** Sampling clock division ratio (D4) / Serial I/F Ch.2 IrDA register (0x401F4)

**DIVMD3:** Sampling clock division ratio (D4) / Serial I/F Ch.3 IrDA register (0x401F9)

---

Selects the division ratio of the sampling clock.

Write "1": 1/8

Write "0": 1/16

Read: Valid

Select the division ratio necessary to generate the sampling clock for asynchronous transfers. When DIVMDx is set to "1", the sampling clock is generated from the input clock of the serial interface (output by an 8-bit programmable timer or input from #SCLKx) by dividing it by 8. When DIVMDx is set to "0", the input clock is divided by 16.

At initial reset, DIVMDx becomes indeterminate.

---

**IRTL0:** Ch.0 IrDA output logic inversion (D3) / Serial I/F Ch.0 IrDA register (0x401E4)

**IRTL1:** Ch.1 IrDA output logic inversion (D3) / Serial I/F Ch.1 IrDA register (0x401E9)

**IRTL2:** Ch.2 IrDA output logic inversion (D3) / Serial I/F Ch.2 IrDA register (0x401F4)

**IRTL3:** Ch.3 IrDA output logic inversion (D3) / Serial I/F Ch.3 IrDA register (0x401F9)

---

Inverts the logic of the IrDA output signal.

Write "1": Inverted

Write "0": Not inverted

Read: Valid

When using the IrDA interface, set the logic of the SOUTx output signal to suit the infrared-ray communication circuit that is connected external to the chip. If IRTLx is set to "1", a high pulse is output when the output data = "0" (held low-level when the output data = "1"). If IRTLx is set to "0", a low pulse is output when the output data = "0" (held high-level when the output data = "1").

At initial reset, IRTLx becomes indeterminate.

---

**IRRL0:** Ch.0 IrDA input logic inversion (D2) / Serial I/F Ch.0 IrDA register (0x401E4)

**IRRL1:** Ch.1 IrDA input logic inversion (D2) / Serial I/F Ch.1 IrDA register (0x401E9)

**IRRL2:** Ch.2 IrDA input logic inversion (D2) / Serial I/F Ch.2 IrDA register (0x401F4)

**IRRL3:** Ch.3 IrDA input logic inversion (D2) / Serial I/F Ch.3 IrDA register (0x401F9)

---

Inverts the logic of the IrDA input signal.

Write "1": Inverted

Write "0": Not inverted

Read: Valid

When using the IrDA interface, set the logic of the signal that is input from an external infrared-ray communication circuit to the chip to suit the serial interface. If IRRLx is set to "1", a high pulse is input as a logic "0". If IRRLx is set to "0", a low pulse is input as a logic "0".

At initial reset, IRRLx becomes indeterminate.

---

**IRMD01–IRMD00:** Ch.0 IrDA interface mode selection (D[1:0]) / Serial I/F Ch.0 IrDA register (0x401E4)

**IRMD11–IRMD10:** Ch.1 IrDA interface mode selection (D[1:0]) / Serial I/F Ch.1 IrDA register (0x401E9)

**IRMD21–IRMD20:** Ch.2 IrDA interface mode selection (D[1:0]) / Serial I/F Ch.2 IrDA register (0x401F4)

**IRMD31–IRMD30:** Ch.3 IrDA interface mode selection (D[1:0]) / Serial I/F Ch.3 IrDA register (0x401F9)

---

Selects the IrDA interface function.

Table III.9.15 IrDA Interface Setting

IRMDx1	IRMDx0	Interface mode
1	1	Do not set. (reserved)
1	0	IrDA 1.0 interface
0	1	Do not set. (reserved)
0	0	Normal interface

When using the IrDA interface function, write "10" to IRMDx while setting to an asynchronous mode for the transfer mode. If the IrDA interface function is not to be used, write "00" to IRMDx.

At initial reset, IRMDx becomes indeterminate.

**Note:** This selection must always be performed before the transfer mode and other conditions are set.

---

**PSIO02–PSIO00:** Ch.0 interrupt level (D[6:4]) / 8-bit timer, serial I/F Ch.0 interrupt priority register (0x40269)

**PSIO12–PSIO10:** Ch.1 interrupt level (D[2:0]) / Serial I/F Ch.1, A/D interrupt priority register (0x4026A)

**PSIO22–PSIO20:** Ch.2 interrupt level (D[2:0]) / Serial I/F Ch.2/3 interrupt priority register (0x4026E)

**PSIO32–PSIO30:** Ch.3 interrupt level (D[6:4]) / Serial I/F Ch.2/3 interrupt priority register (0x4026E)

---

Sets the priority level of the serial-interface interrupt.

The interrupt priority level can be set for each channel in the range of 0 to 7.

At initial reset, PSIOx becomes indeterminate.

---

**ESERR0, ESRX0, ESTX0:** Ch.0 interrupt enable (D0,D1,D2) / Serial I/F Ch.0/1 interrupt enable register (0x40276)

**ESERR1, ESRX1, ESTX1:** Ch.1 interrupt enable (D3,D4,D5) / Serial I/F Ch.0/1 interrupt enable register (0x40276)

**ESERR2, ESRX2, ESTX2:** Ch.2 interrupt enable (D0,D1,D2) / Serial I/F Ch.2/3 interrupt enable register (0x40279)

**ESERR3, ESRX3, ESTX3:** Ch.3 interrupt enable (D3,D4,D5) / Serial I/F Ch.2/3 interrupt enable register (0x40279)

---

Enable or disable interrupt generation to the CPU.

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

The ESERRx, ESRXx, and ESTXx bits are interrupt enable bits corresponding to receive-error, receive-buffer full, and transmit-buffer empty interrupt factors, respectively, in each channel. The interrupts for which this bit is set to "1" are enabled, and the interrupts for which this bit is set to "0" are disabled.

At initial reset, all these bits are set to "0" (interrupts disabled).

---

<b>FSERR0, FSRX0, FSTX0:</b>	Ch.0 interrupt factor flags (D0,D1,D2) / Serial I/F Ch.0/1 interrupt factor flag register (0x40286)
<b>FSERR1, FSRX1, FSTX1:</b>	Ch.1 interrupt factor flags (D3,D4,D5) / Serial I/F Ch.0/1 interrupt factor flag register (0x40286)
<b>FSERR2, FSRX2, FSTX2:</b>	Ch.2 interrupt factor flags (D0,D1,D2) / Serial I/F Ch.2/3 interrupt factor flag register (0x40289)
<b>FSERR3, FSRX3, FSTX3:</b>	Ch.3 interrupt factor flags (D3,D4,D5) / Serial I/F Ch.2/3 interrupt factor flag register (0x40289)

---

Indicate the status of serial-interface interrupt generation.

**When read**

- Read "1": An interrupt factor occurred
- Read "0": No interrupt factor occurred

**When written using the reset-only method (default)**

- Write "1": Flag is reset
- Write "0": Invalid

**When written using the read/write method**

- Write "1": Flag is set
- Write "0": Flag is reset

The FSERRx, FSRXx, and FSTXx flags are interrupt factor flags corresponding to receive-error, receive-buffer full, and transmit-buffer empty interrupts, respectively, in each channel. The flag is set to "1" when each interrupt factor occurs.

A transmit-buffer empty interrupt factor occurs when transmit data is transferred from the transmit data register to the shift register.

A receive-buffer full interrupt factor occurs when receive data is transferred from the shift register to the receive data register.

A receive-error interrupt factor occurs when a parity, framing, or overrun error is detected during reception of data. At this time, if the following conditions are met, an interrupt to the CPU is generated:

1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The PSR's IE bit is set to "1" (interrupts enabled).
4. The set value of the corresponding interrupt priority register is higher than the CPU interrupt level (IL).

When using the receive-buffer full or transmit-buffer empty interrupt factor as an IDMA request, the fact that the above conditions are met does not necessarily mean that an interrupt request to the CPU has been output simultaneously when an interrupt factor occurs. An interrupt is generated under the above conditions upon completion of the data transfer by IDMA, provided that interrupts are enabled by settings on the IDMA side.

The interrupt factor flag is set to "1" whenever an interrupt factor occurs, regardless of the settings of the interrupt-enable and interrupt priority registers.

If the next interrupt is to be accepted following the occurrence of an interrupt, it is necessary that the interrupt factor flag be reset, and that the PSR be set up again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).

The interrupt factor flag can only be reset by writing to it in the software. Note that if the PSR is set up again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.

At initial reset, all of these flags become indeterminate, so be sure to reset them in the software.

---

**RSRX0, RSTX0:** Ch.0 IDMA request

(D6, D7) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA request register (0x40292)

**RSRX1, RSTX1:** Ch.1 IDMA request (D0, D1) / Serial I/F Ch.1, A/D IDMA request register (0x40293)

**RSRX2, RSTX2:** Ch.2 IDMA request (D2, D3) / 8-bit timer 4/5, Serial I/F Ch.2/3 IDMA request register (0x4029B)

**RSRX3, RSTX3:** Ch.3 IDMA request (D4, D5) / 8-bit timer 4/5, Serial I/F Ch.2/3 IDMA request register (0x4029B)

Specifies whether to invoke IDMA when an interrupt factor occurs.

**When using the set-only method (default)**

Write "1": IDMA request

Write "0": Not changed

Read: Valid

**When using the read/write method**

Write "1": IDMA request

Write "0": Interrupt request

Read: Valid

The RSRXx and RSTXx bits are IDMA request bits corresponding to receive-buffer full and transmit-buffer empty interrupt factors, respectively. If the bit is set to "1", IDMA is invoked when an interrupt factor occurs, thus performing a programmed data transfer. If this bit is set to "0", normal interrupt processing is performed, without invoking IDMA.

For details on IDMA, refer to "IDMA (Intelligent DMA)".

At initial reset, these bits are set to "0" (interrupt request).

---

**DESRX0, DESTX0:** Ch.0 IDMA enable

(D6, D7) / 16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA enable register (0x40296)

**DESRX1, DESTX1:** Ch.1 IDMA enable (D0, D1) / Serial I/F Ch.1, A/D IDMA enable register (0x40297)

**DESRX2, DESTX2:** Ch.2 IDMA enable (D2, D3) / 8-bit timer 4/5, Serial I/F Ch.2/3 IDMA enable register (0x4029C)

**DESRX3, DESTX3:** Ch.3 IDMA enable (D4, D5) / 8-bit timer 4/5, Serial I/F Ch.2/3 IDMA enable register (0x4029C)

Enables IDMA transfer by means of an interrupt factor.

**When using the set-only method (default)**

Write "1": IDMA enabled

Write "0": Not changed

Read: Valid

**When using the read/write method**

Write "1": IDMA enabled

Write "0": IDMA disabled

Read: Valid

The DESRXx and DESTXx bits are IDMA enable bits corresponding to receive-buffer full and transmit-buffer empty interrupt factors, respectively. If the bit is set to "1", the IDMA request by the interrupt factor is enabled. If the bit is set to "0", the IDMA request is disabled.

At initial reset, these bits are set to "0" (IDMA disabled).

III

SIF

## Programming Notes

- (1) Before setting various serial-interface parameters, make sure the transmit and receive operations are disabled (TXEN<sub>x</sub> = RXEN<sub>x</sub> = "0").
- (2) When the serial interface is transmitting or receiving data, do not set TXEN<sub>x</sub> or RXEN<sub>x</sub> to "0", and do not execute the slp instruction.
- (3) In clock-synchronized transfers, the mode of communication is half-duplex, in which the clock line is shared between the transmit and receive units. Therefore, RXEN<sub>x</sub> and TXEN<sub>x</sub> cannot be enabled simultaneously.
- (4) After an initial reset, the interrupt factor flag becomes indeterminate. To prevent generation of an unwanted interrupt or IDMA request, reset this flag in the program.
- (5) If a receive error occurs, the receive-error interrupt and receive-buffer full interrupt factors occur simultaneously. However, since the receive-error interrupt has priority over the receive-buffer full interrupt, the receive-error interrupt is processed first. Therefore, it is necessary to reset the receive-buffer full interrupt factor flag through the use of the receive-error interrupt processing routine.
- (6) To prevent the regeneration of interrupts due to the same factor following the occurrence of an interrupt, always be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.
- (7) Follow the procedure described below to initialize the serial interface.

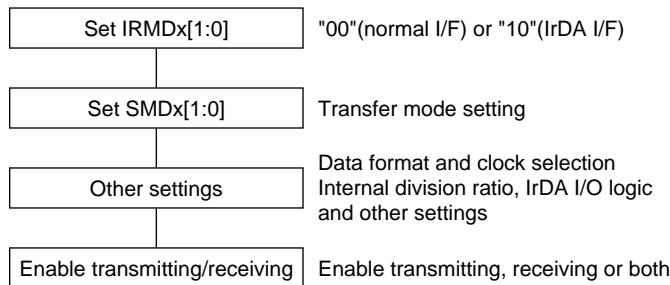


Figure III.9.18 Serial Interface Initialize Procedure

- (8) When transmitting data in the clock-synchronized master mode, transmit data is written to the transmit data register after the initial setting is performed following the flow in item (7). However, the clock generated by the 8-bit timer must be supplied to the serial interface (at least one underflow has had to have occurred in the 8-bit timer) before this writing. Otherwise, 0xFF will be transmitted prior to the written data.
- (9) The maximum transfer rate of the serial interface is limited to 1 Mbps.
- (10) If the receive circuit is stopped during reception, set both transmission and reception to the disabled status.
- (11) When performing data transfer in the clock-synchronized mode, the division ratio of the prescaler and the reload data for the 8-bit programmable timer should be set so that the baud-rate is 1/4 of the system clock frequency or lower.
- (12) The serial interface operates only when the prescaler is operating.
- (13) During IrDA receive operations, the RZI circuit recognizes low pulses by means of the signal edge (rising edge when IRRL<sub>x</sub> = "0"; falling edge when IRRL<sub>x</sub> = "1"). Note that noise may cause a malfunction.



## III-10 SERIAL INTERFACE with FIFO

### Configuration of Serial Interface with FIFO

#### Features of Serial Interface with FIFO

In addition to the four channels of standard serial interfaces, the S1C33301 contains one channel (Ch.0) of serial interface with FIFO, the features of which are described below.

- A clock-synchronized or asynchronous mode can be selected for the transfer method.

Clock-synchronized mode

Data length: 8 bits, fixed (No start, stop, and parity bits)

Receive error: An overrun error can be detected.

Asynchronous mode

Data length: 7 or 8 bits, selectable

Receive error: Overrun, framing, or parity errors can be detected.

Start bit: 1 bit, fixed

Stop bit: 1 or 2 bits, selectable

Parity bit: Even, odd, or none; selectable

Since the transmit and receive units are independent, full-duplex communication is possible.

- Baud rate setting: Any desired baud rate can be set by selecting the 10-bit baud-rate timer, or using external clock input (asynchronous mode only).
- 4-byte receive buffer (FIFO) and 2-byte transmit buffer (FIFO) are built in, allowing for successive receive and transmit operations.
- Data transfers using IDMA or HSDMA are possible.
- Three types of interrupts (transmit data empty, receive data full, and receive error) can be generated.

Figure III.10.1 shows the configuration of the serial interface with FIFO.

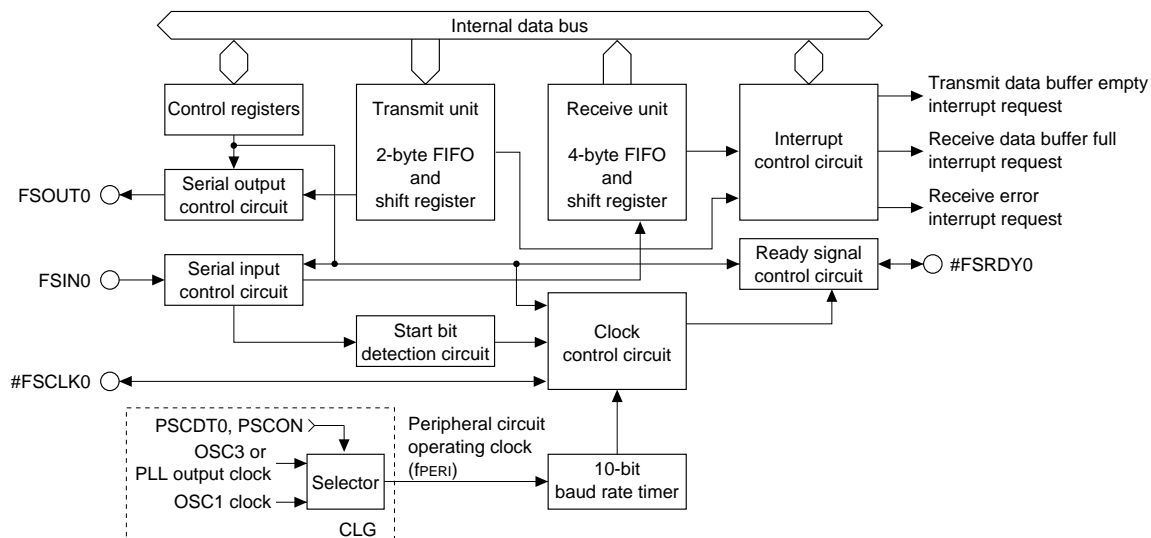


Figure III.10.1 Configuration of Serial Interface

**Note:** The I/O pins for the serial interface with FIFO are shared with the standard serial interface Ch.0 explained in the previous chapter. Use the P00–P03 port function extension register (0x300040) to select which serial interface to be used.

## I/O Pins of Serial Interface with FIFO

Table III.10.1 lists the I/O pins used by the serial interface with FIFO.

Table III.10.1 Configuration of Serial Interface with FIFO

Pin name	I/O	Function	Function select bit
P00(SIN0/ <b>FSIN0</b> )	I/O	I/O port / Serial IF Ch.0 data input / Serial IF with FIFO Ch.0 data input	EFP00[1:0] (D[1:0])/P00–P03 function extension register(0x300040)
P01(SOUT0/ <b>FSOUT0</b> )	I/O	I/O port / Serial IF Ch.0 data output / Serial IF with FIFO Ch.0 data output	EFP01[1:0] (D[3:2])/P00–P03 function extension register(0x300040)
P02(#SCLK0/ <b>#FSCLK0</b> )	I/O	I/O port / Serial IF Ch.0 clock input/output / Serial IF with FIFO Ch.0 clock input/output	EFP02[1:0] (D[5:4])/P00–P03 function extension register(0x300040)
P03(#SRDY0/ <b>#FSRDY0</b> )	I/O	I/O port / Serial IF Ch.0 ready input/output / Serial IF with FIFO Ch.0 ready input/output	EFP03[1:0] (D[7:6])/P00–P03 function extension register(0x300040)

### **FSIN0 (serial-data input pin)**

This pin is used to input serial data to the device, regardless of the transfer mode.

### **FSOUT0 (serial-data output pin)**

This pin is used to output serial data from the device, regardless of the transfer mode.

### **#FSCLK0 (clock input/output pin)**

This pin is used to input or output a clock.

In the clock-synchronized slave mode, it is used as a clock input pin; in the clock-synchronized master mode, it is used as a clock output pin.

In the asynchronous mode, this pin is used as clock input when an external clock is used. This pin is not used when the internal clock is used, so it can be used as an I/O port.

### **#FSRDY0 (ready-signal input/output pin)**

This pin is used to input or output the ready signal that is used in the clock-synchronized mode.

In the clock-synchronized slave mode, it is used as a ready-signal output pin; in the clock-synchronized master mode, it is used as a ready-signal input pin.

This pin is not used in the asynchronous mode, so it can be used as an I/O port.

### **Method for setting the serial-interface input/output pins**

All of the pins used in the serial interface with FIFO are shared with the standard serial interface Ch.0 and P00–P03 I/O ports. At cold start, they are all set for I/O port pins P0x (function extension bits EFP0xx = "0"). When using the serial interface with FIFO, make function extension bit settings for the pins used, according to the transfer mode to be used.

## Setting BCU for using the Serial Interface with FIFO

The control registers for the serial interface with FIFO (except for interrupt control registers) are mapped into Area 6 addresses 0x300200 to 0x300209. Therefore, in order for the registers to be accessed, the BCU register for Area 6 must be set up in accordance with the procedure described below.

1. A6IO (D9) / Access control register (0x48132) = "1"  
This sets Area 6 so that the internal device can be accessed.
2. A6EC (D1) / Access control register (0x48132) = "0"  
This sets Area 6 so that it can be accessed in the little endian format.
3. A6WT[2:0] (D[A:8]) / Areas 6–4 setup register (0x4812A) = "001"  
This sets Area 6 so that it can be accessed with 1 wait state inserted.

## Setting Transfer Mode

The transfer mode of the serial interface with FIFO can be set using SMD0[1:0] as shown in Table III.10.2 below.

Table III.10.2 Transfer Mode

SMD01	SMD00	Transfer mode
1	1	8-bit asynchronous mode
1	0	7-bit asynchronous mode
0	1	Clock-synchronized slave mode
0	0	Clock-synchronized master mode

At initial reset, SMD0 becomes indeterminate, so be sure to initialize it in the software.

When using the IrDA interface, set the transfer mode for the asynchronous 7-bit or asynchronous 8-bit mode.

The input/output pins are configured differently, depending on the transfer mode. The pin configuration in each mode is shown in Table III.10.3.

Table III.10.3 Pin Configuration by Transfer Mode

Transfer mode	FSIN0	FSOUT0	#FSCLK0	#FSRDY0
8-bit asynchronous mode	Data input	Data output	Clock input/P port	P port
7-bit asynchronous mode	Data input	Data output	Clock input/P port	P port
Clock-synchronized slave mode	Data input	Data output	Clock input	Ready output
Clock-synchronized master mode	Data input	Data output	Clock output	Ready input

All four pins are used in the clock-synchronized mode.

In the asynchronous mode, since #FSRDY0 is unused, P03 can be used as an I/O (P) port. In addition, when an external clock is not used, P02 can also be used as an I/O port.

The I/O control and data registers for the I/O ports used in this serial interface can be used as general-purpose read/write registers.

**Note:** To enable the IrDA interface to be set, IRMD0[1:0] (D[1:0]) / FIFO serial I/F Ch.0 IrDA register (0x300204) is provided. Since these bits become indeterminate at initial reset, be sure to initialize them by writing "00" when using as the normal interface or "10" when using as the IrDA interface.

## Baud-Rate Timer (Setting Baud Rate)

The clock-synchronized master mode uses the internal clock for data transfer. Also in the asynchronous mode, the internal clock can be selected as the operating clock. This serial interface has a dedicated baud-rate timer (10-bit programmable timer) built-in to generate this clock. The counter initial value can be set by software, this makes it possible to program a flexible transfer rate/sampling frequency.

It is not necessary to configure and run the baud-rate timer, when this serial interface is used in the clock-synchronized slave mode or in the asynchronous mode using an external clock.

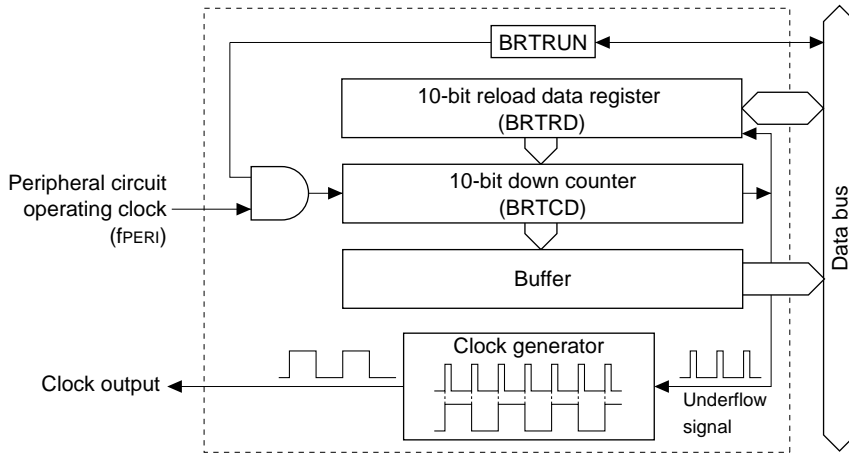


Figure III.10.2 Baud-Rate Timer

The baud-rate timer is configured with a 10-bit presettable down counter BRTCD[9:0] (D[1:0] / 0x300209, D[7:0] / 0x300208) and a 10-bit reload data register BRTRD[9:0] (D[1:0] / 0x300207, D[7:0] / 0x300206) for setting an initial value to the counter.

The counter uses the peripheral circuit operating clock for counting.

The peripheral circuit operating clock is one of the clocks below selected using the PSCDT0 (D0) / Prescaler clock select register (0x40181).

When PSCDT0 = "0", the OSC3 clock (when PLL is not used) or the PLL output clock (when PLL is used) is selected.

When PSCDT1 = "1", the OSC1 clock (32 kHz Typ.) is selected.

At initial reset, the OSC3/PLL clock is selected.

Note that the selected clock is supplied to this serial interface and other peripheral circuits only when the PSCON (D5) / Power control register (0x40180) is set to "1". At initial reset, PSCON is set to "1" and the clock supply is enabled. When PSCON is set to "0", the clock supply stops.

Refer to "CLG (Clock Generator)" for controlling the oscillation circuit and the CPU operating clock, and "Prescaler" for controlling the prescaler clock.

The following procedure generates the clock by the baud-rate timer.

1. Set an initial value to the reload data register BRTRD[9:0] (D[1:0] / 0x300207, D[7:0] / 0x300206)
2. Set BRTRUN (D0) / FIFO serial I/F Ch.0 baud-rate timer control register (0x300205) to "1"

The baud-rate timer loads the initial value set in the reload data register to the counter when "1" is written to BRTRUN, and then starts counting down. When the counter underflows, it outputs an underflow pulse and loads the reload data again to continue counting.

The underflow occurs in the cycle determined by the reload data. The clock generator reverses its output signal level using the underflow signal to generate a clock with 50% duty ratio and 1/2 the frequency of the underflow signal.

The baud-rate timer should be stopped (set BRTRUN to "0") when serial communication is not needed to reduce current consumption.

## Calculating the reload data

The initial value for the reload data register is determined by the expressions shown below. Note that the expression for the clock-synchronized master mode is different from the asynchronous mode, because the clock-synchronized master mode uses the baud-rate timer output clock directly as the synchronous clock, and the asynchronous mode uses it as the sampling clock.

### Clock-synchronized master mode

$$\text{BRTRD} = \frac{f_{\text{PERI}}}{2 \times \text{bps}} - 1 \quad (\text{Eq. 1})$$

BRTRD: Reload data register setup value of the baud-rate timer

$f_{\text{PERI}}$ : C33 peripheral circuit operating clock frequency (Hz)

bps: Transfer rate (bits/second)

### Asynchronous mode

$$\text{BRTRD} = \frac{f_{\text{PERI}} \times \text{DIVMD}}{2 \times \text{bps}} - 1 \quad (\text{Eq. 2})$$

BRTRD: Reload data register setup value of the baud-rate timer

$f_{\text{PERI}}$ : C33 peripheral circuit operating clock frequency (Hz)

bps: Transfer rate (bits/second)

DIVMD: Internal division ratio of the serial interface (1/16 or 1/8 selected by DIVMD0)

Table III.10.4 shows examples of the reload data settings, in cases in which the internal division ratio of the serial interface is set to 1/16 (asynchronous mode).

Table III.10.4 Example of Transfer Rate Settings

Transfer rate (bps)	$f_{\text{PERI}} = 20 \text{ MHz}$		$f_{\text{PERI}} = 25 \text{ MHz}$		$f_{\text{PERI}} = 33 \text{ MHz}$	
	BRTRD	Error (%)	BRTRD	Error (%)	BRTRD	Error (%)
1200	520	-0.16026	650	0.00640	858	0.04366
2400	259	0.16026	325	-0.14698	429	-0.07267
4800	129	0.16026	162	-0.14698	214	-0.07267
9600	64	0.16026	80	0.46939	106	0.39427
14400	42	0.93669	53	0.46939	71	-0.53530
28800	21	-1.35732	26	0.46939	35	-0.53530

Make sure the error is within 1%. Calculate the error using the following equation:

$$\text{Error} = \left\{ \frac{f_{\text{PERI}} \times \text{DIVMD}}{(\text{BRTRD} + 1) \times 2 \times \text{bps}} - 1 \right\} \times 100 [\%]$$

## Clock-Synchronized Interface

### Outline of Clock-Synchronized Interface

In the clock-synchronized transfer mode, 8 bits of data are synchronized to the common clock on both the transmit and receive sides when the data is transferred. Since the transmit unit has 2-byte buffer and the receive unit has 4-byte buffer (FIFO), successive transmit and receive operations are possible. Since the clock line is shared between the transmit and receive units, the communication mode is half-duplex.

#### Master and slave modes

Either the clock-synchronized master mode or the clock-synchronized slave mode can be selected using SMD0[1:0] (D[1:0]) / FIFO serial I/F Ch.0 control register (0x300203).

#### Clock-synchronized master mode (SMD0[1:0] = "00")

In this mode, clock-synchronized 8-bit serial transfers, in which the serial interface functions as the master, can be performed using the internal clock to synchronize the operation of the internal shift registers.

The synchronizing clock is output from the #FSCLK0 pin, enabling an external (slave side) serial input/output device to be controlled. The #FSRDY0 pin is also used to input a signal that indicates whether the external serial input/output device is ready to transmit or receive (when ready in a low level).

#### Clock-synchronized slave mode (SMD0[1:0] = "01")

In this mode, clock-synchronized 8-bit serial transfers, in which the serial interface functions as a slave, can be performed using the synchronizing clock that is supplied by an external (master side) serial input/output device. The synchronizing clock is input from the #FSCLK0 pin for use as the synchronizing clock of the serial interface. In addition, a #FSRDY0 signal indicating whether the serial interface is ready to transmit or receive (when ready in a low level) is output from the #FSRDY0 pin.

Figure III.10.3 shows an example of how the input/output pins are connected in the clock-synchronized mode.

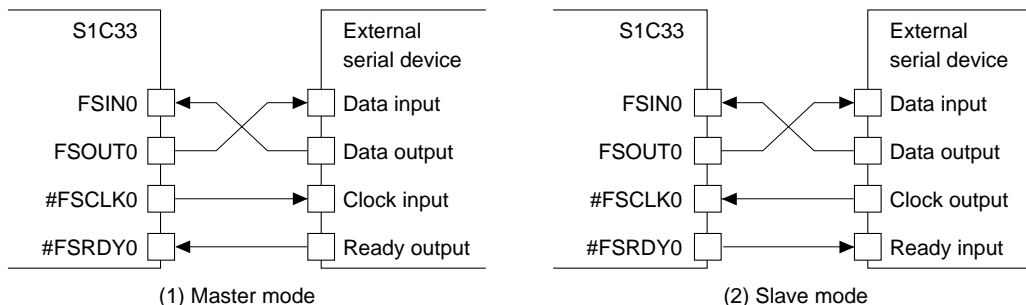


Figure III.10.3 Example of Connection in Clock-Synchronized Mode

#### Clock-synchronized transfer data format

In clock-synchronized transfers, the data format is fixed as shown below.

- Data length: 8 bits
- Start bit: None
- Stop bit: None
- Parity bit: None

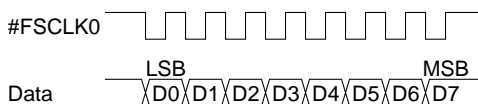


Figure III.10.4 Clock-Synchronized Transfer Data Format

Serial data is transmitted and received starting with the LSB.

## Setting Clock-Synchronized Interface

When performing clock-synchronized transfers via the serial interface with FIFO, the following settings must be made before data transfer is actually begun:

1. Setting input/output pins
2. Setting the interface mode
3. Setting the transfer mode
4. Setting the input clock
5. Setting the receive FIFO level
6. Setting interrupts and IDMA/HSDMA

The following explains the content of each setting. For details on interrupt/DMA settings, refer to "FIFO Serial Interface Interrupts and DMA".

**Note:** Always make sure the serial interface is inactive (TXEN0 and RXEN0 = "0") before these settings are made. A change of settings during operation may cause a malfunction.

### Setting input/output pins

All four pins—FSIN0, FSOUT0, #FSCLK0, and #FSRDY0—are used in the clock-synchronized mode. Set EFP0xx (D[7:0]) / P00–P03 port function extension register (0x300040) to "01010101".

### Setting the interface mode

Write "00" to IRMD0[1:0] (D[1:0]) / FIFO serial I/F Ch.0 IrDA register (0x300204), which is used to set the interface mode (ordinary or IrDA interface), to choose the ordinary interface. Since IRMD0[1:0] becomes indeterminate at initial reset, it must be initialized.

### Setting the transfer mode

Use SMD0 to set the transfer mode of the serial interface as described earlier. When using the serial interface as the master for clock-synchronized transfer, set SMD0[1:0] to "00"; when using the serial interface as a slave, set SMD0[1:0] to "01".

### Setting the input clock

#### • Clock-synchronized master mode

This mode operates using the internal clock generated by the baud-rate timer. Setup and run the baud-rate timer as described in "Baud-Rate Timer (Setting Baud Rate)".

The FIFO serial I/F control register contains an SSCK0 bit to select the clock source used for the asynchronous mode. Although this bit does not affect the clock in the clock-synchronized mode, its content becomes indeterminate at initial reset. Therefore, be sure to initialize this bit by writing "0" (Internal clock), even when using the serial interface in the clock-synchronized master mode.

#### • Clock-synchronized slave mode

This mode operates using the clock that is output by the external master. This clock is input from the #FSCLK0 pin.

Therefore, there is no need to control the baud-rate timer.

Initialize SSCK0 by writing "1" (#FSCLK0).

### Setting the receive FIFO level

This serial interface incorporates a 4-byte receive FIFO allowing up to 4 bytes of data that can be received without an error even when the receive data register is not read. This serial interface can generate a receive-buffer full interrupt when the specified number of data are received in the receive FIFO. Use FIFOINT0[1:0] (D[6:5]) / FIFO serial I/F Ch.0 IrDA register (0x300204) to set this number of data. Writing 0–3 to FIFOINT0 sets the number of data to 1–4. The default setting at initial reset is 0 so that a receive-buffer full interrupt will generate when one data is received.

## Control and Operation of Clock-Synchronized Transfer

### Transmit control

#### (1) Enabling transmit operation

Use the transmit-enable bit TXEN0 (D7) / FIFO serial I/F Ch.0 control register (0x300203) for transmit control. When transmit is enabled by writing "1" to this bit, the clock input to the shift register is enabled (ready for input), thus allowing for data to be transmitted. The synchronizing clock input/output of the #FSCLK0 pin is also enabled (ready for input/output).

Transmit is disabled and the transmit data buffer (FIFO) is cleared by writing "0" to TXEN0.

After the port function extension register is set for the serial interface with FIFO, the I/O direction of the #FSRDY0 and #FSCLK0 pins are changed at follows:

#FSRDY0: When slave mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

#FSCLK0: When master mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

**Note:** In clock-synchronized transfers, the clock line is shared between the transmit and receive units, so the communication mode is half-duplex. Therefore, TXEN0 and receive-enable bit RXEN0 cannot be enabled simultaneously. When transmitting data, fix RXEN0 at "0" and do not change it during a transmit operation.

In addition, make sure TXEN0 is not set to "0" during a transmit operation.

#### (2) Transmit procedure

The serial interface contains a transmit shift register and a transmit data register, which are provided independently of those used for a receive operation.

Transmit data is written to TXD0[7:0] (D[7:0]) / FIFO serial I/F Ch.0 transmit data register (0x300200). The data written to TXD0 enters the transmit data buffer and waits for transmission.

The transmit data buffer is a 2-byte FIFO and up to two data can be written to it successively if empty. Older data will be transmitted first and cleared after transmission. The next transmit data can be written to the transmit data register, even during data transmission. The transmit data buffer status flag TDBE0 (D1) / FIFO serial I/F Ch.0 status register (0x300202) is provided to check whether this buffer is full or not. This flag is set to "1" when the transmit data buffer has a free space for transmit data to be written and reset to "0" when the transmit data buffer becomes full by writing transmit data.

The serial interface starts transmitting when data is written to the transmit data register. The transfer status can be checked using the transmit-completion flag TEND0 (D5) / FIFO serial I/F Ch.0 status register (0x300202).

This flag goes "1" when data is being transmitted and goes "0" when the transmission has completed.

When data is transmitted successively in clock-synchronized master mode, TEND0 maintains "1" until all data is transmitted (Figure III.10.5). In slave mode, TEND0 goes "0" every time 1-byte data is transmitted (Figure III.10.6).

When all the data in the transmit data buffer are transferred, a transmit-data empty interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the next piece of transmit data can be written using an interrupt processing routine. In addition, since this interrupt factor can be used to invoke DMA, the data prepared in memory can be transmitted successively to the transmit-data register through DMA transfers. For details on how to control interrupts and DMA requests, refer to "FIFO Serial Interface Interrupts and DMA".

Following explains transmit operation in both the master and slave modes.

- **Clock-synchronized master mode**

The timing at which the device starts transmitting in the master mode is as follows:

When #FSRDY0 is on a low level while the transmit-data buffer contains data written to it or when data has been written to the transmit-data buffer while #FSRDY0 is on a low level.

Figure III.10.5 shows a transmit timing chart in the clock-synchronized master mode.



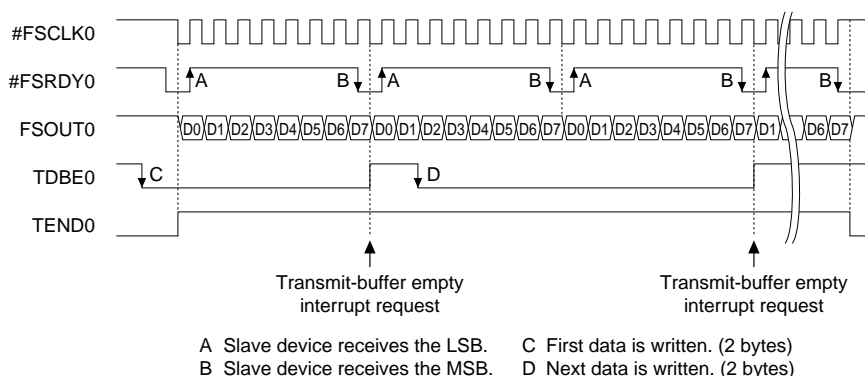


Figure III.10.5 Transmit Timing Chart in Clock-Synchronized Master Mode

1. If the #FSRDY0 signal from the slave is on a high level, the master waits until it is on a low level (ready to receive).
2. If #FSRDY0 is on a low level, the synchronizing clock input to the serial interface begins. The synchronizing clock is also output from the #FSCLK0 pin to the slave device.
3. The content of the data buffer is transferred to the shift register synchronously with the first falling edge of the clock. At the same time, the LSB of the data transferred to the shift register is output from the FSOUT0 pin. If the transmit data buffer becomes empty at this point, a transmit-buffer empty interrupt request occurs.
4. The data in the shift register is shifted 1 bit by the next falling edge of the clock, and the bit following the LSB is output from FSOUT0. This operation is repeated until all 8 bits of data are transmitted. The slave device takes in each bit synchronously with the rising edges of the synchronizing clock.
5. The next data transfer begins if the transmit data buffer contains other data.

#### • Clock-synchronized slave mode

Figure III.10.6 shows a transmit timing chart in the clock-synchronized slave mode.

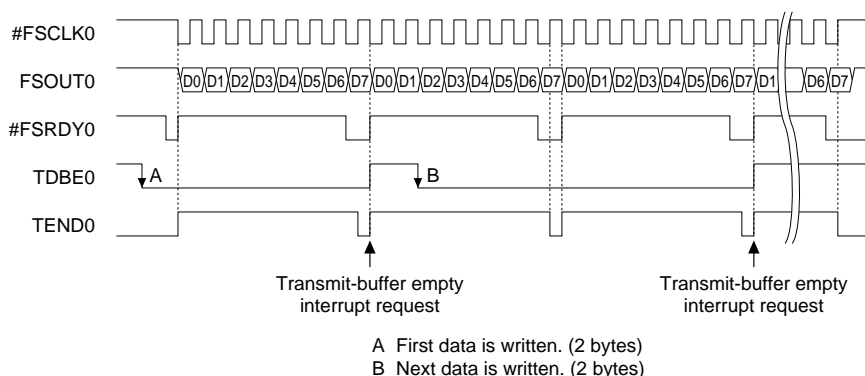


Figure III.10.6 Transmit Timing Chart in Clock-Synchronized Slave Mode

1. After setting the #FSRDY0 signal to a low level (ready to transmit), the slave waits for clock input from the master.
2. When the synchronizing clock is input from the #FSCLK0 pin, the content of the data register is transferred to the shift register synchronously with the first falling edge of the clock. At the same time, the LSB of the data transferred to the shift register is output from the FSOUT0 pin. The #FSRDY0 signal is returned to a high level at this point.
3. The data in the shift register is shifted 1 bit by the next falling edge of the clock, and the bit following the LSB is output from FSOUT0. This operation is repeated until all 8 bits of data are transmitted.
4. The #FSRDY0 signal is set to a low level when the last bit (8th bit) is output from the FSOUT0 pin. The master device takes in each bit synchronously with the rising edges of the synchronizing clock.
5. The next data transfer begins if the transmit data buffer contains other data.

### (3) Terminating transmit operation

Upon completion of data transmission, write "0" to the transmit-enable bit TXEN0 to disable transmit operation. This operation clears (initializes) the transmit data buffer (FIFO), therefore, make sure that the transmit data buffer does not contain any data waiting for transmission before writing "0" to TXEN0.

## Receive control

### (1) Enabling receive operation

Use the receive-enable bit RXEN0 (D6) / FIFO serial I/F Ch.0 control register (0x300203) for receive control. When receive operations are enabled by writing "1" to this bit, clock input to the shift register is enabled (ready for input), thereby starting a data-receive operation. The synchronizing clock input/output on the #FSCLK0 pin also is enabled (ready for input/output). Receive operations are disabled and the receive data buffer (FIFO) is cleared by writing "0" to RXEN0.

After the function select register is set for the serial interface, the I/O direction of the #FSRDY0 and #FSCLK0 pins are changed at follows:

#FSRDY0: When slave mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

#FSCLK0: When master mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

**Note:** In clock-synchronized transfers, the clock line is shared between the transmit and receive units, so the communication mode is half-duplex. Therefore, RXEN0 and transmit-enable bit TXEN0 cannot be enabled simultaneously. When receiving data, fix TXEN0 at "0" and do not change it during a receive operation. In addition, make sure RXEN0 is not set to "0" during a receive operation.

### (2) Receive procedure

This serial interface has a receive shift register, receive data buffer and a receive data register that are provided independently of those used for transmit operations.

The received data enters the received data buffer. The receive data buffer is a 4-byte FIFO and can receive data until it becomes full unless the received data is not read out.

The received data in the buffer can be read by accessing the received data register RXD0[7:0] (0x300201). The older data is output first and cleared by reading.

The number of data in the receive data buffer can be checked by reading RXD0NUM[1:0] (D[7:6]) / FIFO serial I/F Ch.0 status register (0x300202). When RXD0NUM[1:0] is 0, the buffer contains 0 or 1 data. When RXD0NUM[1:0] is 1–3, the buffer contains 2–4 data.

Furthermore, the RDBF0 flag (D0) / FIFO serial I/F Ch.0 status register (0x300202) is provided for indicating whether the received data buffer is empty or not. This flag is set to "1" when the receive data buffer contains one or more received data, and is reset to "0" when the receive data buffer becomes empty by reading all the received data.

When the receive data buffer has received the specified number or more data, a receive-data full interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the received data can be read by an interrupt processing routine. In addition, since this interrupt factor can be used to invoke DMA, the received data can be received successively in locations prepared in memory through DMA transfers.

For details on how to control interrupts/DMA, refer to "FIFO Serial Interface Interrupts and DMA".

The following describes a receive operation in the master and slave modes.

### • Clock-synchronized master mode

Figure III.10.7 shows a receive timing chart in the clock-synchronized master mode.

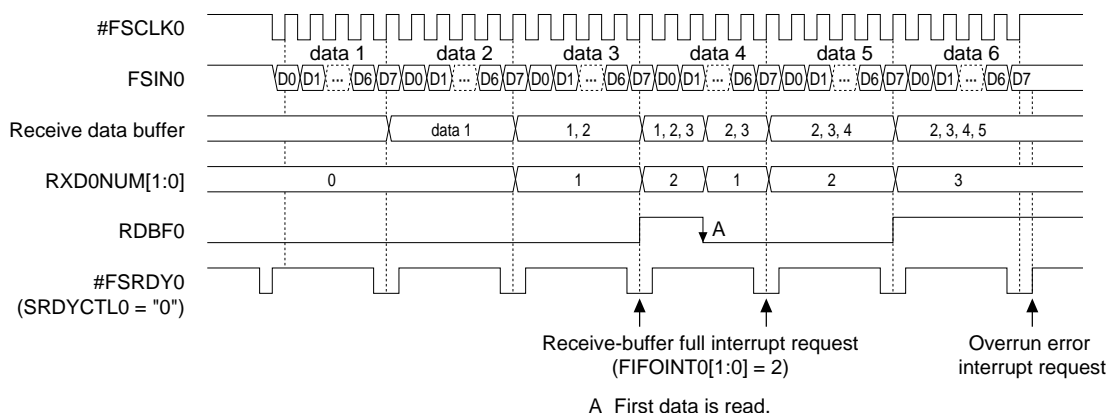


Figure III.10.7 Receive Timing Chart in Clock-Synchronized Master Mode

1. If the #FSRDY0 signal from the slave is on a high level, the master waits until it turns to a low level (ready to receive).
2. If #FSRDY0 is on a low level, synchronizing clock input to the serial interface begins. The synchronizing clock is also output from the #FSClk0 pin to the slave device.
3. The slave device outputs each bit of data synchronously with the falling edges of the clock. The LSB is output first.
4. This serial interface takes the FSIN0 input into the shift register at the rising edges of the clock. The data in the shift register is sequentially shifted as bits are taken in. This operation is repeated until the MSB of data is received.
5. When the MSB is taken in, the data in the shift register is transferred to the receive data buffer, enabling the data to be read out.

### • Clock-synchronized slave mode

Figure III.10.8 shows a receive timing chart in the clock-synchronized slave mode.

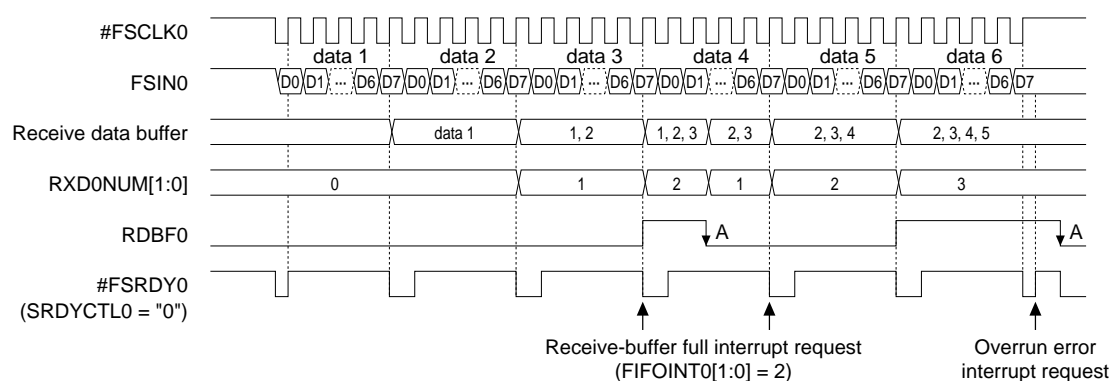


Figure III.10.8 Receive Timing Chart in Clock-Synchronized Slave Mode

1. After setting the #FSRDY0 signal to a low level (ready to receive), the slave waits for clock input from the master.
2. The master device outputs each bit of data synchronously with the falling edges of the clock. The LSB is output first.
3. This serial interface takes the FSIN0 input into the shift register at the rising edges of the clock that is input from #FSClk0. The data in the shift register is sequentially shifted as bits are taken in. This operation is repeated until the MSB of data is received.
4. When the MSB is taken in, the data in the shift register is transferred to the receive data buffer, enabling the data to be read out.

**(3) Overrun error**

Even when the receive data buffer is full (4 data have been received), the next (5th) data can be received into the shift register. If there is no space in the buffer (data has not been read) when the 5th data has been received, the 5th data in the shift register cannot be transferred to the buffer. If one more (6th) data is transferred to this serial interface, the shift register (5th data) is overwritten with the 6th data and an overrun error is generated.

When an overrun error is generated, the overrun error flag OER0 (D2) / FIFO serial I/F Ch.0 status register (0x300202) is set to "1". Once the overrun error flag is set to "1", it remains set until it is reset by writing "0" to it in the software.

The overrun error is one of the receive-error interrupt factors in the serial interface. An interrupt can be generated for this error by setting the interrupt controller as necessary, so that the error can be processed by an interrupt processing routine.

Generation of overrun error can be disabled by controlling the #FSRDY0 as shown below.

**(4) Controlling the #FSRDY0 signal**

When the slave device is in receive mode, the #FSRDY0 signal is output from the slave device to the master device to notify whether the slave device is ready to receive data or not.

When this serial interface is in the clock-synchronized slave mode, the #FSRDY0 signal is turned to a low level by writing "1" to RXEN0 to enable receive operations, thereby indicating to the master device that the slave is ready to receive. When the LSB of data is received, #FSRDY0 is turned to a high level; when the MSB is received, #FSRDY0 is returned to a low level, in preparation for the next receive operation.

If an overrun error occurs, #FSRDY0 is turned to a high level (unable to receive) at that point, so receive operations for the following data are suspended. In this case, #FSRDY0 is returned to low by reading out the receive data buffer, and if any receive data follows, the slave restarts receiving data.

In the normal mode, the #FSRDY0 signal indicating ready to receive is output even if the receive data buffer is full. If the receive data buffer cannot be read in this case, an overrun error occurs in the next data transfer. To prevent this error, the serial interface with FIFO provides #FSRDY0 high mask mode. In this mode, if the receive data buffer is full, the #FSRDY0 signal is forcibly fixed at high in order to suspend data transfer from the master device until the data in the buffer is read.

To use this function, set SRDYCTL0 (D7) / FIFO serial I/F Ch.0 IrDA register (0x300204) to "1".

This function is effective in the clock-synchronized master mode as well. In this case, the #FSRDY0 signal (low) from the slave device is ignored when the receive data buffer is full and the serial interface stops outputting the #FSCLK0 signal until the buffer data is read.

When the receive data buffer is not full, normal receive operation is performed even if this function is enabled.

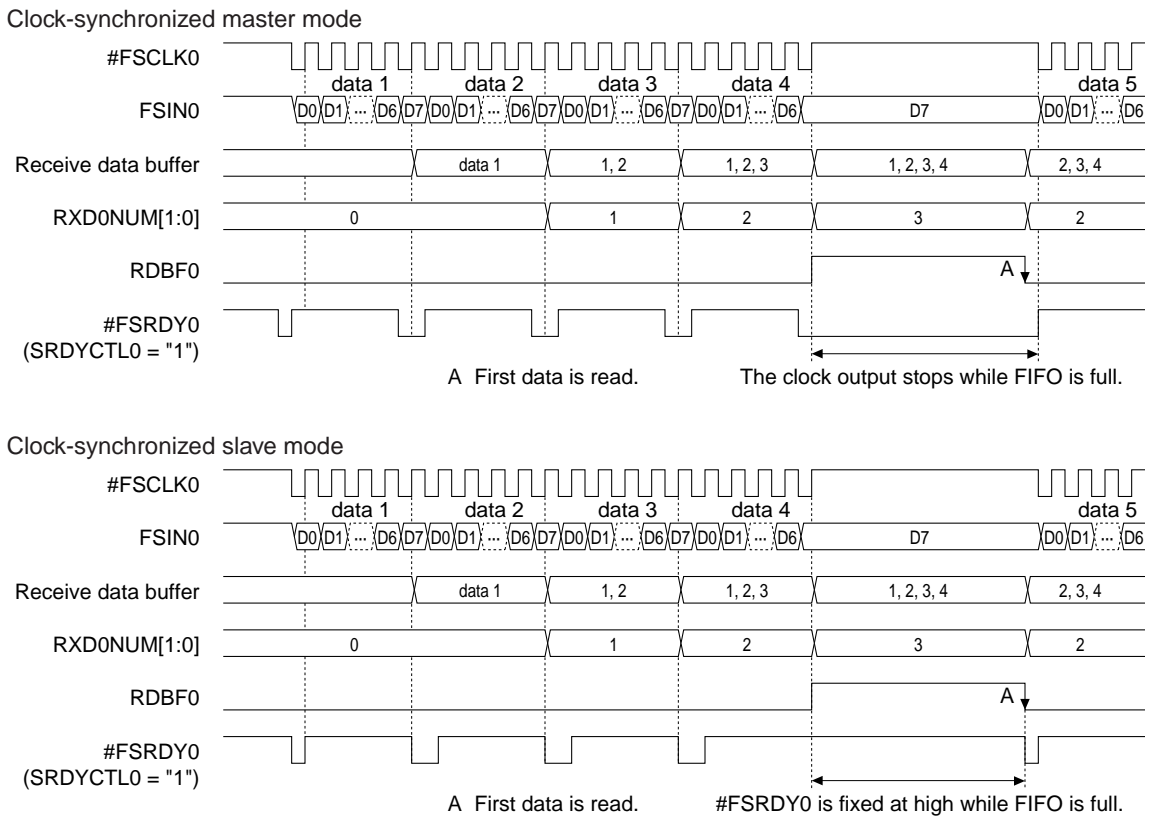


Figure III.10.9 #FSRDY0 High Mask Mode

**(5) Terminating receive operation**

Upon completion of a data receive operation, write "0" to the receive-enable bit RXEN0 to disable receive operations. This operation clears (initializes) the receive data buffer (FIFO), therefore, make sure that there is no data that has not been read in the receive data buffer before setting RXEN0 to "0".

## Asynchronous Interface

### Outline of Asynchronous Interface

Asynchronous transfers are performed by adding a start bit and a stop bit to the start and end points of each serial-converted data. With this method, there is no need to use a clock that is fully synchronized on the transmit and receive sides; instead, transfer operations are timed by the start and stop bits added to the start and end points of each data.

In the 8-bit asynchronous mode (SMD0[1:0] = "11"), 8 bits of data can be transferred; in the 7-bit asynchronous mode (SMD0[1:0] = "10"), 7 bits of data can be transferred.

In either mode, it is possible to select the stop-bit length, add a parity bit, and choose between even and odd parity. The start bit is fixed at "1".

The operating clock can be selected between an internal clock generated by the baud-rate timer or an external clock that is input from the #FSCLK0 pin.

Since the transmit unit has 2-byte buffer and the receive unit has 4-byte buffer (FIFO), successive transmit and receive operations are possible. Furthermore, since the transmit and receive units are independent, full-duplex communication in which transmit and receive operations are performed simultaneously is also possible.

Figure III.10.10 shows an example of how input/output pins are connected for transfers in the asynchronous mode.

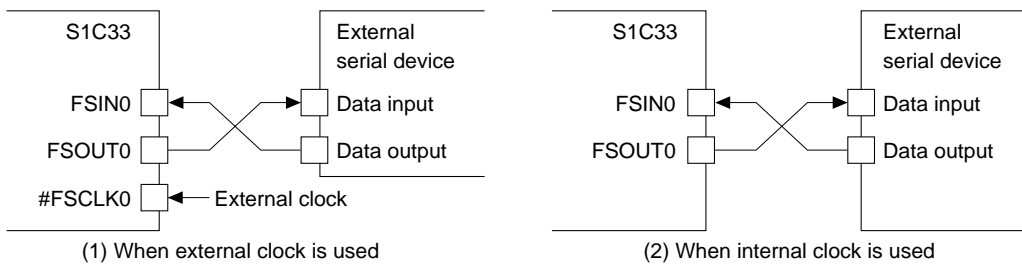


Figure III.10.10 Example of Connection in Asynchronous Mode

When the asynchronous mode is selected, it is possible to use the IrDA interface function.

### Asynchronous-transfer data format

The data format for asynchronous transfer is shown below.

Data length: 7 or 8 bits (determined by the selected transfer mode)

Start bit: 1 bit, fixed

Stop bit: 1 or 2 bits

Parity bit: Even or odd parity, or none

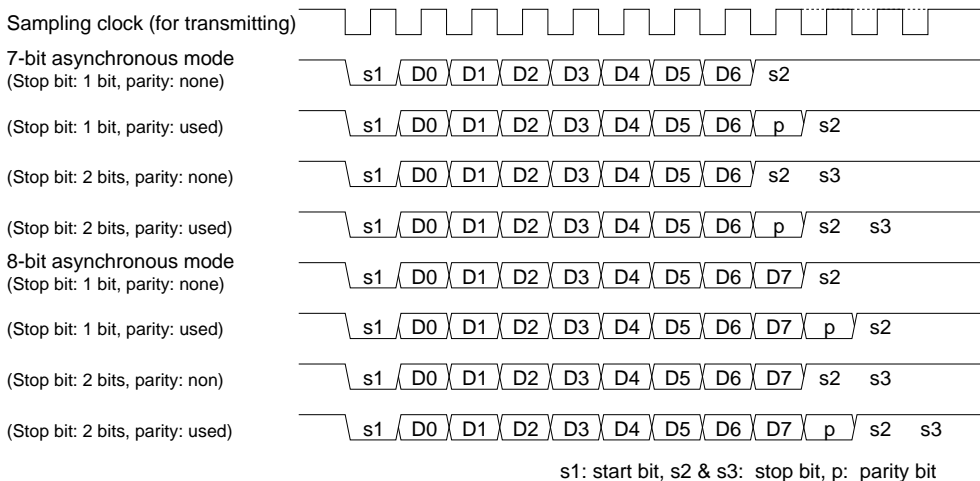


Figure III.10.11 Data Format for Asynchronous Transfer

Serial data is transmitted and received, starting with the LSB.

## Setting Asynchronous Interface

When performing asynchronous transfer via the serial interface with FIFO, the following must be done before data transfer can be started:

1. Setting input/output pins
2. Setting the interface mode
3. Setting the transfer mode
4. Setting the input clock
5. Setting the data format
6. Setting the receive FIFO level
7. Setting interrupt/IDMA/HSDMA

The following describes how to set each of the above. For details on interrupt/DMA settings, refer to "FIFO Serial Interface Interrupts and DMA".

**Note:** Always make sure the serial interface is inactive (TXEN0 and RXEN0 = "0") before making these settings. A change in settings during operation may result in a malfunction.

### Setting input/output pins

In the asynchronous mode, two pins—FSIN0 and FSOUT0—are used, so set both EFP01[1:0] (D[1:0]) and EFP02[1:0] (D[3:2]) / P00–P03 port function extension register (0x300040) to "01".

When external clock input is used, one more pin, #FSCLK0, is also used. Set EFP03[1:0] (D[5:4]) / P00–P03 port function extension register (0x300040) to "01".

### Setting the interface mode

IRMD0[1:0] (D[1:0]) / FIFO serial I/F Ch.0 IrDA register (0x300204) is used to set the IrDA interface. Since IRMD0[1:0] becomes indeterminate at initial reset, initialize it by writing "00" when using the serial interface as a normal interface, or "10" when using the serial interface as an IrDA interface. This setting must be made before a transfer mode is set.

### Setting the transfer mode

Use SMD0 to set the transfer mode of the serial interface as described earlier. When using the serial interface in the 8-bit asynchronous mode, set SMD0[1:0] to "11", when using the serial interface in the 7-bit asynchronous mode, set SMD0[1:0] to "10".

### Setting the input clock

In the asynchronous mode, the operating clock can be selected between the internal clock and an external clock. The external clock is selected (input from the #FSCLK0 pin) by writing "1" to SSCK0 (D2) / FIFO serial I/F Ch.0 control register (0x300203), and an internal clock is selected by writing "0".

**Note:** SSCK0 becomes indeterminate at initial reset, so be sure to reset it in the software.

#### • Internal clock

When the internal clock is selected, the serial interface is clocked by a clock generated using the baud-rate timer. Setup and run the baud-rate timer as described in "Baud-Rate Timer (Setting Baud Rate)".

#### • External clock

When an external clock is selected, the serial interface is clocked by a clock input from the #FSCLK0 pin. Therefore, there is no need to control the baud-rate timer.

Any desired clock frequency can be set. The clock input from the #FSCLK0 pin is internally divided by 16 or 8 in the serial interface, in order to create a sampling clock (refer to "Sampling clock"). This division ratio must also be considered when setting the transfer rate.

• **Sampling clock**

In the asynchronous mode, TCLK (the clock output by the baud-rate timer or input from the #FSCLK0 pin) is internally divided in the serial interface, in order to create a sampling clock.

A 1/16 division ratio is selected by writing "0" to DIVMD0 (D4) / FIFO serial I/F Ch.0 IrDA register (0x300204), and a 1/8 ratio is selected by writing "1".

**Note:** The DIVMD0 bit becomes indeterminate at initial reset, so be sure to reset it in the software. Settings of this bit are valid only in the asynchronous mode (and when using the IrDA interface).

**For receiving**

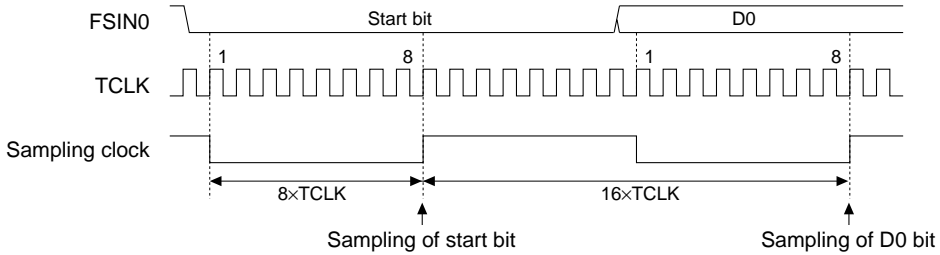


Figure III.10.12 Sampling Timing for Asynchronous Receive Operation (when 1/16 division is selected)

Each bit data is sampled in the timing shown in Figure III.10.12. When the FSIN0 input signal is detected as a low level at the rising edge of TCLK, sampling for the start bit is performed  $8 \times \text{TCLK}$  ( $4 \times \text{TCLK}$  when 1/8 division is selected) after that point. If a low level is not detected in the sampling for the start bit, the interface aborts the subsequent samplings and returns to the start bit detection phase (in this case no error occurs). When the FSIN0 input signal is low at the start bit sampling, subsequent bit data is sampled in  $16 \times \text{TCLK}$  cycles ( $8 \times \text{TCLK}$  cycles when 1/8 division is selected).

**For transmitting**

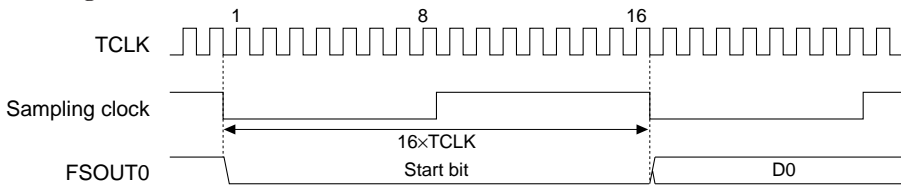


Figure III.10.13 Bit data Output During Asynchronous Transmit Operation (when 1/16 division is selected)

During transmission, each bit data is output from the FSOUT0 pin in  $16 \times \text{TCLK}$  cycles ( $8 \times \text{TCLK}$  cycles when 1/8 division is selected).



## Setting the data format

In the asynchronous mode, the data length is 7 or 8 bits as determined by the transfer mode set. The start bit is fixed at 1.

The stop and parity bits can be set as shown in the Table III.10.5 using the following control bits:

Stop-bit select: STPB0 (D3) / FIFO serial I/F Ch.0 control register (0x300203)

Parity enable: EPR0 (D5) / FIFO serial I/F Ch.0 control register (0x300203)

Parity-mode select: PMD0 (D4) / FIFO serial I/F Ch.0 control register (0x300203)

Table III.10.5 Stop Bit and Parity Bit Settings

STPB0	EPR0	PMD0	Stop bit	Parity bit
1	1	1	2 bits	Odd
		0	2 bits	Even
	0	*	2 bits	None
0	1	1	1 bit	Odd
		0	1 bit	Even
	0	*	1 bit	Non

\* Setting PMD0 is invalid when EPR0 = "0".

**Note:** These bits become indeterminate at initial reset, so be sure to initialize them in the software.

## Setting the receive FIFO level

This serial interface incorporates a 4-byte receive FIFO allowing up to 4 bytes of data that can be received without an error even when the receive data register is not read. This serial interface can generate a receive-buffer full interrupt when the specified number of data are received in the receive FIFO. Use FIFOINT0[1:0] (D[6:5]) / FIFO serial I/F Ch.0 IrDA register (0x300204) to set this number of data. Writing 0–3 to FIFOINT0 sets the number of data to 1–4. The default setting at initial reset is 0 so that a receive-buffer full interrupt will generate when one data is received.

## ***Control and Operation of Asynchronous Transfer***

### **Transmit control**

#### **(1) Enabling transmit operation**

Use the transmit-enable bit TXEN0 (D7) / FIFO serial I/F Ch.0 control register (0x300203) for transmit control. When transmit is enabled by writing "1" to this bit, the clock input to the shift register is enabled (ready for input), thus allowing data to be transmitted.

Transmit is disabled and the transmit data buffer (FIFO) is cleared by writing "0" to TXEN0.

**Note:** Do not set TXEN0 to "0" during a transmit operation.

#### **(2) Transmit procedure**

The serial interface contains a transmit shift register and a transmit data register, which are provided independently of those used for a receive operation.

Transmit data is written to TXD0[7:0] (D[7:0]) / FIFO serial I/F Ch.0 transmit data register (0x300200). In the 7-bit asynchronous mode, bit 7 (MSB) in each register is ignored.

The data written to TXD0 enters the transmit data buffer and waits for transmission.

The transmit data buffer is a 2-byte FIFO and up to two data can be written to it successively if empty. Older data will be transmitted first and cleared after transmission. The next transmit data can be written to the transmit data register, even during data transmission. The transmit data buffer status flag TDBE0 (D1) / FIFO serial I/F Ch.0 status register (0x300202) is provided to check whether this buffer is full or not. This flag is set to "1" when the transmit data buffer has a free space for transmit data to be written and reset to "0" when the transmit data buffer becomes full by writing transmit data.

The serial interface starts transmitting when data is written to the transmit data register. The transfer status can be checked using the transmit-completion flag TEND0 (D5) / FIFO serial I/F Ch.0 status register (0x300202).

This flag goes "1" when data is being transmitted and goes "0" when the transmission has completed.

When all the data in the transmit data buffer are transferred, a transmit-data empty interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the next piece of transmit data can be written using an interrupt processing routine. In addition, since this interrupt factor can be used to invoke DMA, the data prepared in memory can be transmitted successively to the transmit-data register through DMA transfers. For details on how to control interrupts and DMA requests, refer to "FIFO Serial Interface Interrupts and DMA".

Figure III.10.14 shows a transmit timing chart in the asynchronous mode.

Example: Data length 8 bits  
 Stop bit 1 bit  
 Parity bit Included

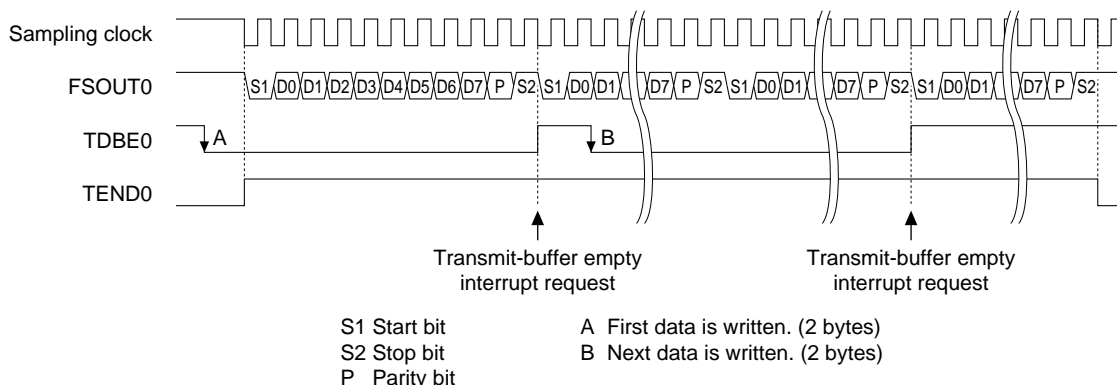


Figure III.10.14 Transmit Timing Chart in Asynchronous Mode

1. The contents of the data register are transferred to the shift register synchronously with the first falling edge of the sampling clock. At the same time, the FSOUT0 pin is setting to a low level to send the start bit.
2. Each bit of data in the shift register is transmitted beginning with the LSB at each falling edge of the subsequent sampling clock. This operation is repeated until all 8 (or 7) bits of data are transmitted.
3. After sending the MSB, the parity bit (if EPR0 = "1") and the stop bit are transmitted insuccession.
4. The next data transfer begins if the transmit data buffer contains other data.

### (3) Terminating transmit operations

When data transmission is completed, write "0" to the transmit-enable bit TXEN0 to disable transmit operations. This operation clears (initializes) the transmit data buffer (FIFO), therefore, make sure that the transmit data buffer does not contain any data waiting for transmission before writing "0" to TXEN0.

## Receive control

### (1) Enabling receive operations

Use the receive-enable bit RXEN0 (D6) / FIFO serial I/F Ch.0 control register (0x300203) for receive control. When receiving enabled by writing "1" to this bit, clock input to the shift register is enabled (ready for input), meaning that it is ready to receive data. Receive operations are disabled and the receive data buffer (FIFO) is cleared by writing "0" to RXEN0.

**Note:** Do not set RXEN0 to "0" during a receive operation.

### (2) Receive procedure

This serial interface has a receive shift register, receive data buffer and a receive data register that are provided independently of those used for transmit operations.

The received data enters the received data buffer. The receive data buffer is a 4-byte FIFO and can receive data until it becomes full unless the received data is not read out.

The received data in the buffer can be read by accessing the received data register RXD0[7:0] (0x300201). The older data is output first and cleared by reading.

The number of data in the receive data buffer can be checked by reading RXD0NUM[1:0] (D[7:6]) / FIFO serial I/F Ch.0 status register (0x300202). When RXD0NUM[1:0] is 0, the buffer contains 0 or 1 data. When RXD0NUM[1:0] is 1–3, the buffer contains 2–4 data.

Furthermore, the RDBF0 flag (D0) / FIFO serial I/F Ch.0 status register (0x300202) is provided for indicating whether the received data buffer is empty or not. This flag is set to "1" when the receive data buffer contains one or more received data, and is reset to "0" when the receive data buffer becomes empty by reading all the received data.

When the receive data buffer has received the specified number or more data, a receive-data full interrupt factor occurs. Since an interrupt can be generated as set by the interrupt controller, the received data can be read by an interrupt processing routine. In addition, since this interrupt factor can be used to invoke DMA, the received data can be received successively in locations prepared in memory through DMA transfers.

For details on how to control interrupts/DMA, refer to "FIFO Serial Interface Interrupts and DMA".

Figure III.10.15 shows a receive timing chart in the asynchronous mode.

Example: Data length 8 bits

Stop bit 1 bit

Parity bit Included

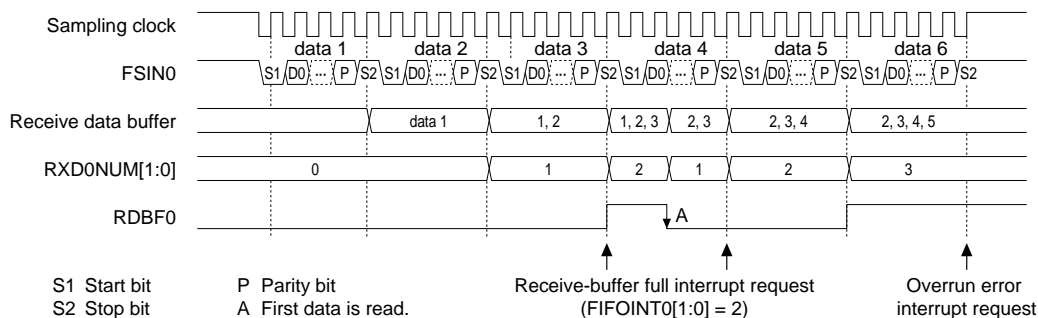


Figure III.10.15 Receive Timing Chart in Asynchronous Mode

1. The serial interface starts sampling when the start bit is input (FSINO = low).
2. When the start bit is sampled at the first rising edge of the sampling clock, each bit of receive data is taken into the shift register, beginning with the LSB at each rising edge of the subsequent clock. This operation is repeated until the MSB of data is received.
3. When the MSB is taken in, the parity bit that follows is also taken in (if EPR0 = "1").
4. When the stop bit is sampled, the data in the shift register is transferred to the receive data register, enabling the data to be read out.

The parity is checked when data is transferred to the receive data register (if EPR0 = "1").

**Note:** The receive operation is terminated when the first stop bit is sampled even if the stop bit is configured with two bits.

**(3) Receive errors**

Three types of receive errors can be detected when receiving data in the asynchronous mode.

Since an interrupt can be generated by setting the interrupt controller, the error can be processed using an interrupt processing routine. For details on receive error interrupts, refer to "FIFO Serial Interface Interrupts and DMA".

- **Parity error**

If EPR0 is set to "1" (parity added), the parity is checked when data is received.

This parity check is performed when the data received in the shift register is transferred to the receive data register in order to check conformity with PMD0 settings (odd or even parity). If any nonconformity is found in this check, a parity error is assumed and the parity error flag PER0 (D3) / FIFO serial I/F Ch.0 status register (0x300202) is set to "1".

Even when this error occurs, the received data in error is transferred to the receive data buffer and the receive operation is continued. However, the content of the received data for which a parity error is flagged cannot be guaranteed.

The PER0 flag is reset to "0" by writing "0".

- **Framing error**

If data with a stop bit = "0" is received, the serial interface assumes that the data is out of synchronization and generates a framing error.

If two stop bits are used, only the first stop bit is checked.

When this error occurs, the framing-error flag FER0 (D4) / FIFO serial I/F Ch.0 status register (0x300202) is set to "1".

Even when this error occurs, the received data in error is transferred to the receive data register and the receive operation is continued. However, the content of the received data for which a framing error is flagged cannot be guaranteed, even if no framing error is found in the following data received.

The FER0 flag is reset to "0" by writing "0".

- **Overrun error**

Even when the receive data buffer is full (4 data have been received), the next (5th) data can be received into the shift register. If there is no space in the buffer (data has not been read) when the 5th data has been received, the 5th data in the shift register cannot be transferred to the buffer. If one more (6th) data is transferred to this serial interface, the shift register (5th data) is overwritten with the 6th data and an overrun error is generated.

When an overrun error is generated, the overrun error flag OER0 (D2) / FIFO serial I/F Ch.0 status register (0x300202) is set to "1".

Even when this error occurs, the receive operation is continued.

The OER0 flag is reset to "0" by writing "0".

**(4) Terminating receive operation**

When a data receive operation is completed, write "0" to the receive-enable bit RXEN0 to disable receive operations. This operation clears (initializes) the receive data buffer (FIFO), therefore, make sure that there is no data that has not been read in the receive data buffer before setting RXEN0 to "0".

## IrDA Interface

### Outline of IrDA Interface

Each channel of the serial interface contains a RZI modulator circuit, allowing an infrared-ray communication circuit to be configured based on IrDA 1.0 simply by adding a simple external circuit.

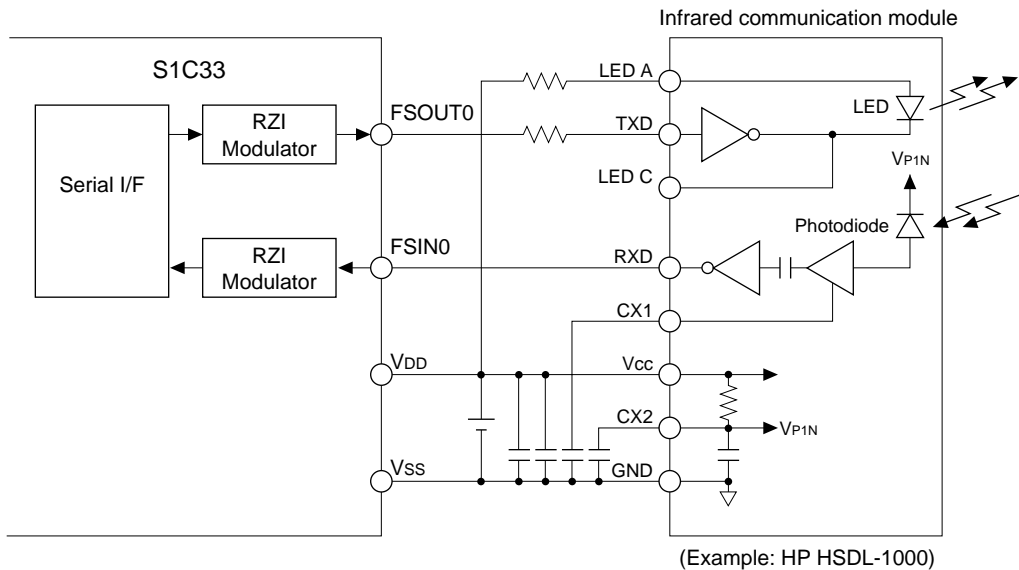


Figure III.10.16 Configuration Example of IrDA Interface

This IrDA interface function can be used only when the selected transfer mode is an asynchronous mode. Since the contents of the asynchronous mode are applied directly for the serial-interface functions other than the IrDA interface unit, refer to "Asynchronous Interface", for details on how to set and control the data formats and data transfers.

### Setting IrDA Interface

When performing infrared-ray communication, the following settings must be made before communication can be started:

1. Setting input/output pins
2. Selecting the interface mode (IrDA interface function)
3. Setting the transfer mode
4. Setting the input clock
5. Setting the data format
6. Setting the receive FIFO level
7. Setting the interrupt/IDMA/HSDMA
8. Setting the input/output logic

The contents for items 1 through 6 have been explained in connection with the asynchronous interface. For details, refer to "Asynchronous Interface". For details on item 7, refer to "FIFO Serial Interface Interrupts and DMA".

**Note:** Before making these settings, always make sure the serial interface with FIFO is inactive (TXEN0 and RXEN0 are both set to "0"), as a change in settings during operation could cause a malfunction.

In addition, be sure to set the transfer mode in (3) and the following items before selecting the IrDA interface function in (2).

### Selecting the IrDA interface function

To use the IrDA interface function, select it using IRMD0[1:0] (D[1:0]) / FIFO serial I/F Ch.0 IrDA register (0x300204) and then set the 8-bit (or 7-bit) asynchronous mode as the transfer mode.

Table III.10.6 Setting of IrDA Interface

IRMD01	IRMD00	Interface mode
1	1	Do not set. (reserved)
1	0	IrDA 1.0 interface
0	1	Do not set. (reserved)
0	0	Normal interface

**Note:** The IRMD0 bit becomes indeterminate when initially reset, so be sure to initialize it in the software.

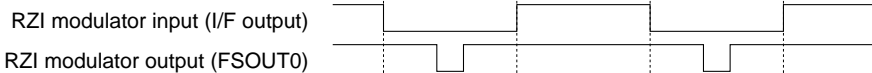
### Setting the input/output logic

When using the IrDA interface, the logic of the input/output signals of the RZI modulator circuit can be changed in accordance with the infrared-ray communication module or the circuit connected externally to the chip. The logic of the internal serial interface is "active-low". If the input/output signals are active-high, the logic of these signals must be inverted before they can be used. The input FSIN0 and output FSOUT0 logic can be set individually through the use of the IRRLO (D2) and IRTL0 (D3) / FIFO serial I/F Ch.0 IrDA register (0x300204), respectively.

The logic of the input/output signal is inverted by writing "1" to each corresponding bit. Logic is not inverted if the bit is set to "0".

When transmitting

(1) IRTL0 = "0"



(2) IRTL0 = "1"



When receiving

(1) IRRLO = "0"



(2) IRRLO = "1"



Figure III.10.17 IRRLO and IRTL0 Settings

**Note:** The IRRLO and IRTL0 bits become indeterminate at initial reset, so be sure to initialize them in the software.

### Control and Operation of IrDA Interface

The transmit/receive procedures have been explained in the section on the asynchronous interface, so refer to "Control and Operation of Asynchronous Transfer".

The following describes the data modulation and demodulation performed using the RZI modulator circuit:

#### When transmitting

During data transmission, the pulse width of the serial interface output signal is set to  $3/16$  before the signal is output from the FSOUT0 pin.

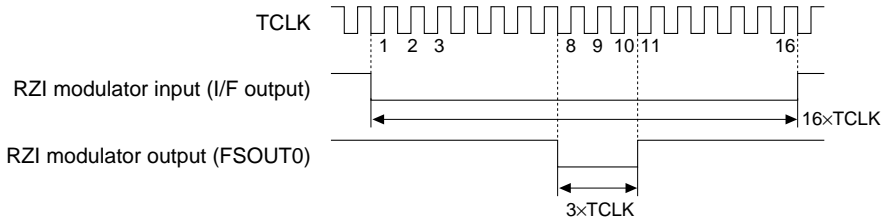


Figure III.10.18 Data Modulation by RZI Circuit

#### When receiving

During data reception, the pulse width of the input signal from FSIN0 is set to  $16/3$  before the signal is transferred to the serial interface.

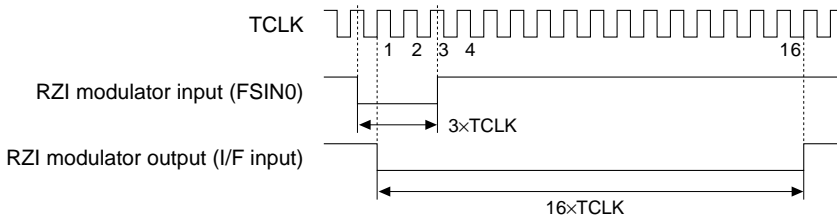


Figure III.10.19 Demodulation by RZI Circuit

**Note:** When using the IrDA interface, set the internal division ratio of the serial interface  $1/16$  (DIVMD0 = "1"), rather than  $1/8$  (DIVMD0 = "0").



## FIFO Serial Interface Interrupts and DMA

The serial interface with FIFO can generate the following three types of interrupts:

- Transmit-buffer empty interrupt
- Receive-buffer full interrupt
- Receive-error interrupt

### Transmit-buffer empty interrupt factor

This interrupt factor occurs when all the transmit data set in the transmit data buffer are transferred to the shift register, in which case the interrupt factor flag FFSTX0 is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated.

Occurrence of this interrupt factor indicates that the next transmit data can be written to the transmit data register.

This interrupt factor can also be used to invoke IDMA, enabling transmit data to be written to the register by means of a DMA transfer.

### Receive-buffer full interrupt

This interrupt factor occurs when the number of data specified with FIFOINT0[1:0] (D[6:5]/0x300204) has been received in the receive data buffer, in which case the interrupt factor flag FFSRX0 is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated.

Occurrence of this interrupt factor indicates that the received data can be read out.

This interrupt factor can also be used to invoke IDMA, enabling the received data to be written into specified memory locations by means of a DMA transfer.

### Receive-error interrupt

This interrupt factor occurs when a parity, framing, or overrun error is detected during data reception, in which case the interrupt factor flag FFSERR0 is set to "1". At this time, if the interrupt conditions set using the interrupt control register are met, an interrupt to the CPU is generated.

Since all three types of errors generate the same interrupt factor, check the error flags PER0 (parity error), OER0 (overrun error), and FER0 (framing error) to identify the type of error that has occurred. In the clock-synchronized mode, parity and framing errors do not occur.

**Note:** If a receive error (parity or framing error) occurs, the receive-error interrupt and receive-buffer full interrupt factors occur simultaneously. However, since the receive-error interrupt has priority over the receive-buffer full interrupt, the receive-error interrupt is processed first. It is therefore necessary for the receive-buffer full interrupt factor flag be cleared through the use of the receive-error interrupt processing routine.

### Control registers of the interrupt controller

Table III.10.7 shows the interrupt controller's control registers.

Table III.10.7 Control Register of Interrupt Controller

Interrupt factor	Interrupt factor flag	Interrupt enable register	Interrupt priority register
Receive-error interrupt	FFSERR0 (D3/0x402B2)	EFERR0 (D3/0x402B1)	PFSIO0[2:0] (D[6:4]/0x402B0)
Receive-buffer full	FFSRX0 (D4/0x402B2)	EFSRX0 (D4/0x402B1)	
Transmit-buffer empty	FFSTX0 (D5/0x402B2)	EFSTX0 (D5/0x402B1)	

When the interrupt factor described above occurs, the corresponding interrupt factor flag is set to "1". If the interrupt enable register bit for that interrupt factor has been set to "1", an interrupt request is generated.

Interrupts caused by an interrupt factor can be disabled by leaving the interrupt enable register bit for that factor set to "0". The interrupt factor flag is set to "1" whenever interrupt conditions are met, regardless of the setting of the interrupt enable register (even if it is set to "0").

The interrupt priority register sets the interrupt priority level of each interrupt source in a range between 0 and 7. An interrupt request to the CPU is accepted only when no other interrupt request of a higher priority has been generated. In addition, only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the input interrupt level set by the interrupt priority register, will the input interrupt request actually be accepted by the CPU.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to "ITC (Interrupt Controller)".

### Intelligent DMA

The receive-buffer full interrupt and transmit-buffer empty interrupt factors can be used to invoke intelligent DMA (IDMA). This enables successive transmit/receive operations between memory and the transmit/receive-buffer to be performed by means of a DAM transfer.

The following shows the IDMA channel numbers set for each interrupt factor:

	IDMA Ch.
Receive-buffer full interrupt:	0x38
Transmit-buffer empty interrupt:	0x39

The IDMA request and enable bits shown in Table III.10.8 must be set to "1" for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

Table III.10.8 Control Bits for IDMA Transfer

Interrupt factor	IDMA request bit	IDMA enable bit
Receive-buffer full	RFSRX0 (D2/0x402B3)	DEFSRX0 (D2/0x402B4)
Transmit-buffer empty	RFSTX0 (D3/0x402B3)	DEFSTX0 (D3/0x402B4)

If an interrupt factor occurs when the IDMA request and enable bits are set to "1", IDMA is invoked. No interrupt request is generated at that point. An interrupt request is generated upon completion of the DMA transfer. The bits can also be set so as not to generate an interrupt, with only a DAM transfer performed. For details on DMA transfer and how to control interrupts upon completion of DMA transfer, refer to "IDMA (Intelligent DMA)".

### High-speed DMA

The receive-buffer full interrupt and transmit-buffer empty interrupt factors can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit:

Table III.10.9 HSDMA Trigger Set-up Bits

HSDMA channel	Trigger set-up bits
1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298)
3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299)

For HSDMA to be invoked by the receive-buffer full interrupt factor, the trigger set-up bits should be set to "1101". For HSDMA to be invoked by the transmit-buffer empty interrupt factor, the trigger set-up bits should be set to "1110". Transfer conditions, etc. must also be set on the HSDMA side.

The HSDMA channel is invoked through generation of the interrupt factor.

For details on HSDMA transfer, refer to "HSDMA (High-Speed DMA)".

### Trap vectors

The trap-vector address of each default interrupt factor is set as follows:

Receive-error interrupt:	0x0C001C0
Receive-buffer full interrupt:	0x0C001C4
Transmit-buffer empty interrupt:	0x0C001C8

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

## I/O Memory of Serial Interface with FIFO

Table III.10.10 shows the control bits of the serial interface with FIFO.

Table III.10.10 Control Bits of Serial Interface with FIFO

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks					
FIFO serial I/F Ch.0 transmit data register	0300200 (B)	D7	TXD07	FIFO serial I/F Ch.0 transmit data TXD07(06) = MSB TXD00 = LSB	0x0 to 0xFF(0x7F)		X	R/W	7-bit asynchronous mode does not use TXD07.				
		D6	TXD06				X						
		D5	TXD05				X						
		D4	TXD04				X						
		D3	TXD03				X						
		D2	TXD02				X						
		D1	TXD01				X						
		D0	TXD00				X						
FIFO serial I/F Ch.0 receive data register	0300201 (B)	D7	RXD07	FIFO serial I/F Ch.0 receive data RXD07(06) = MSB RXD00 = LSB	0x0 to 0xFF(0x7F)		X	R	7-bit asynchronous mode does not use RXD07 (fixed at 0).				
		D6	RXD06				X						
		D5	RXD05				X						
		D4	RXD04				X						
		D3	RXD03				X						
		D2	RXD02				X						
		D1	RXD01				X						
		D0	RXD00				X						
FIFO serial I/F Ch.0 status register	0300202 (B)	D7	RXD0NUM1	Number of Ch.0 receive data in FIFO	RXD0NUM[1:0]		Number of data		R/W				
		D6	RXD0NUM0		1	1	4	0					
					1	0	3						
					0	1	2						
					0	0	1 or 0						
		D5	TEND0		Ch.0 transmit-completion flag	1	Transmitting	0		End	0	R	
		D4	FER0		Ch.0 flaming error flag	1	Error	0		Normal	0	R/W	Reset by writing 0.
		D3	PER0		Ch.0 parity error flag	1	Error	0		Normal	0	R/W	
		D2	OER0		Ch.0 overrun error flag	1	Error	0		Normal	0	R/W	
D1	TDBE0	Ch.0 transmit data buffer status	1	Not full	0	Buffer full	1	R					
D0	RDBF0	Ch.0 receive data buffer status	1	Not empty	0	Empty	0	R					
FIFO serial I/F Ch.0 control register	0300203 (B)	D7	TXEN0	Ch.0 transmit enable	1	Enabled	0	Disabled	0	R/W			
		D6	RXEN0	Ch.0 receive enable	1	Enabled	0	Disabled	0	R/W			
		D5	EPR0	Ch.0 parity enable	1	With parity	0	No parity	X	R/W	Valid only in asynchronous mode.		
		D4	PMD0	Ch.0 parity mode selection	1	Odd	0	Even	X	R/W			
		D3	STPB0	Ch.0 stop bit selection	1	2 bits	0	1 bit	X	R/W			
		D2	SSCK0	Ch.0 input clock selection	1	#FSCLK0	0	Internal clock	X	R/W			
		D1	SMD01	Ch.0 transfer mode selection	SMD0[1:0]		Transfer mode		X	R/W			
		D0	SMD00		1	1	8-bit asynchronous						
		1	0		7-bit asynchronous								
		0	1		Clock sync. Slave								
				0	0	Clock sync. Master							
FIFO serial I/F Ch.0 IrDA register	0300204 (B)	D7	SRDYCTL0	Ch.0 #FSRDY control	1	High mask	0	Normal	0	R/W			
		D6	FIFOINT01	Ch.0 receive FIFO level setting	FIFOINT0[1:0]		Level		0	R/W			
		D5	FIFOINT00		1	1	4						
					1	0	3						
					0	1	2						
					0	0	1						
		D4	DIVMD0	Ch.0 async. clock division ratio	1	1/8	0	1/16	X	R/W			
		D3	IRTL0	Ch.0 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.		
		D2	IRRL0	Ch.0 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W			
		D1	IRMD01	Ch.0 interface mode selection	IRMD0[1:0]		I/F mode		X	R/W			
D0	IRMD00	1	1		reserved								
		1	0	IrDA 1.0									
		0	1	reserved									
		0	0	General I/F									
FIFO serial I/F Ch.0 baud-rate timer control register	0300205 (B)	D7-1	–	reserved	–		–	–	0 when being read.				
		D0	BRTRUN	Baud-rate timer Run/Stop control	1	Run	0	Stop	0	R/W			
FIFO serial I/F Ch.0 baud-rate timer reload data register (LSB)	0300206 (B)	D7	BRTRD07	FIFO serial I/F Ch.0 baud-rate timer reload data [7:0]	0x0 to 0xFF (BRTRD0[9:0] = 0x0 to 0x3FF)		0	R/W					
		D6	BRTRD06				0						
		D5	BRTRD05				0						
		D4	BRTRD04				0						
		D3	BRTRD03				0						
		D2	BRTRD02				0						
		D1	BRTRD01				0						
		D0	BRTRD00				0						

### III-10 PERIPHERAL BLOCK: SERIAL INTERFACE with FIFO

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
FIFO serial I/F Ch.0 baud-rate timer reload data register (MSB)	0300207 (B)	D7-2	–	reserved	–	–	–	0 when being read.		
		D1 D0	<b>BRTRD09</b> <b>BRTRD08</b>	FIFO serial I/F Ch.0 baud-rate timer reload data [9:8]	0x0 to 0x3 (BRTRD0[9:0] = 0x0 to 0x3FF)	0 0	R/W			
FIFO serial I/F Ch.0 baud-rate timer count data register (LSB)	0300208 (B)	D7	<b>BRTCD07</b>	FIFO serial I/F Ch.0	0x0 to 0xFF (BRTCD0[9:0] = 0x0 to 0x3FF)	0	R			
		D6	<b>BRTCD06</b>	baud-rate timer count data [7:0]		0				
		D5	<b>BRTCD05</b>			0				
		D4	<b>BRTCD04</b>			0				
		D3	<b>BRTCD03</b>			0				
		D2	<b>BRTCD02</b>			0				
		D1 D0	<b>BRTCD01</b> <b>BRTCD00</b>			0 0				
FIFO serial I/F Ch.0 baud-rate timer count data register (MSB)	0300209 (B)	D7-2	–	reserved	–	–	–	0 when being read.		
		D1 D0	<b>BRTCD09</b> <b>BRTCD08</b>	FIFO serial I/F Ch.0 baud-rate timer count data [9:8]	0x0 to 0x3 (BRTCD0[9:0] = 0x0 to 0x3FF)	0 0	R			
P00-P03 port function extension register	0300040 (B)	D7 D6	<b>EFP031</b>	P03 port extended function	EFP03[1:0]		Function	0	R/W	
			<b>EFP030</b>		1	*	reserved	0		
		D5 D4	<b>EFP021</b>	P02 port extended function	EFP02[1:0]		Function	0	R/W	
			<b>EFP020</b>		1	*	reserved	0		
		D3 D2	<b>EFP011</b>	P01 port extended function	EFP01[1:0]		Function	0	R/W	
			<b>EFP010</b>		1	*	reserved	0		
		D1 D0	<b>EFP001</b>	P00 port extended function	EFP00[1:0]		Function	0	R/W	
			<b>EFP000</b>		1	*	reserved	0		
FIFO serial I/F Ch.0 interrupt priority register	00402B0 (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	<b>PFSIO02</b>	FIFO serial interface Ch.0	0 to 7	X	R/W			
		D5	<b>PFSIO01</b>	interrupt level		X				
		D4	<b>PFSIO00</b>			X				
D3-0	–	reserved	–	–	–	–	0 when being read.			
FIFO serial I/F Ch.0 interrupt enable register	00402B1 (B)	D7-6	–	reserved	–	–	–	0 when being read.		
		D5	<b>EFSTX0</b>	FSI/F Ch.0 transmit buffer empty	1	Enabled	0	Disabled	0	R/W
		D4	<b>EFSRX0</b>	FSI/F Ch.0 receive buffer full					0	R/W
		D3	<b>EFSERR0</b>	FSI/F Ch.0 receive error					0	R/W
		D2-0	–	reserved	–	–	–	–	0 when being read.	
FIFO serial I/F Ch.0 interrupt factor flag register	00402B2 (B)	D7-6	–	reserved	–	–	–	0 when being read.		
		D5	<b>FFSTX0</b>	FSI/F Ch.0 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W
		D4	<b>FFSRX0</b>	FSI/F Ch.0 receive buffer full					X	R/W
		D3	<b>FFSERR0</b>	FSI/F Ch.0 receive error					X	R/W
D2-0	–	reserved	–	–	–	–	0 when being read.			
FIFO serial I/F Ch.0 IDMA request register	00402B3 (B)	D7-4	–	reserved	–	–	–	0 when being read.		
		D3	<b>RFSTX0</b>	FSI/F Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	0	R/W
		D2	<b>RFSRX0</b>	FSI/F Ch.0 receive buffer full					0	R/W
D1-0	–	reserved	–	–	–	–	0 when being read.			
FIFO serial I/F Ch.0 IDMA enable register	00402B4 (B)	D7-4	–	reserved	–	–	–	0 when being read.		
		D3	<b>DEFSTX0</b>	FSI/F Ch.0 transmit buffer empty	1	IDMA enabled	0	IDMA disabled	0	R/W
		D2	<b>DEFSRX0</b>	FSI/F Ch.0 receive buffer full					0	R/W
D1-0	–	reserved	–	–	–	–	0 when being read.			

**EFP001–EFP000:** P00 port extended function (D[1:0]) / P00–P03 port function extension register (0x300040)  
**EFP011–EFP010:** P01 port extended function (D[3:2]) / P00–P03 port function extension register (0x300040)  
**EFP021–EFP020:** P02 port extended function (D[5:4]) / P00–P03 port function extension register (0x300040)  
**EFP031–EFP030:** P03 port extended function (D[7:6]) / P00–P03 port function extension register (0x300040)

Switches the P00–P03 port functions.

Table III.10.11 P00–P03 port extended functions

EFP0x1	EFP0x0	P00	P01	P02	P03
1	*	reserved	reserved	reserved	reserved
0	1	FSIN0	FSOUT0	#FSCLK0	#FSRDY0
0	0	P00/SIN0	P01/SOUT0	P02/#SCLK0	P03/#SRDY0

Set EFP0x[1:0] to "01" when using P00–P03 for the serial interface with FIFO. Set EFP0x[1:0] to "00" when using P00–P03 for the standard serial interface Ch.0 or general-purpose I/O ports.

At cold start, EFP is set to "00" (I/O port/standard serial interface). At hot start, EFP retains its state from prior to the initial reset.

**TXD07–TXD00:** Ch.0 transmit data (D[7:0]) / FIFO serial I/F Ch.0 transmit data register (0x300200)

Sets transmit data.

When data is written to this register (transmit data buffer) after "1" is written to TXEN0, a transmit operation is begun. The data written to TXD0 enters the transmit data buffer and waits for transmission. The transmit data buffer is a 2-byte FIFO and up to two data can be written to it successively if empty. Older data will be transmitted first and cleared after transmission. When all the data in the transmit data buffer are transferred, a transmit-data empty interrupt factor occurs.

In the 7-bit asynchronous mode, TXD07 (MSB) is ignored.

The serial-converted data is output from the FSOUT0 pin beginning with the LSB, in which the bits set to "1" are output as high-level signals and those set to "0" output as low-level signals.

This register can be read as well as written.

At initial reset, the content of TXD0 becomes indeterminate.

**RXD07–RXD00:** Ch.0 receive data (D[7:0]) / FIFO serial I/F Ch.0 receive data register (0x300201)

The data in the receive data buffer can be read from this register beginning with the oldest data first.

The received data enters the received data buffer. The receive data buffer is a 4-byte FIFO and can receive data until it becomes full unless received data is not read out. When the buffer is full and also the shift register contains received data, an overrun error will occur if the received data is not read until the next data receiving begins. The receive buffer status flag RDBF0 is provided to indicate that it is necessary to read the receive data buffer. This flag is set to "1" when the receive data buffer contains one or more received data, and is reset to "0" when the receive data buffer becomes empty by reading all the received data.

When the receive data buffer has received the number of data specified with FIFOINT[1:0], a receive buffer full interrupt factor occurs.

In the 7-bit asynchronous mode, "0" is stored in RXD07.

The serial data input from the FSIN0 pin is converted into parallel data beginning with the LSB, with the high-level signals changed to "1"s and the low-level signals changed to "0"s. The resulting data is stored in this buffer.

This register is a read-only register, so no data can be written to it.

At initial reset, the content of RXD0 becomes indeterminate.

**RXD0NUM1–RXD0NUM0:** Ch.0 number of receive data in FIFO

(D[7:6]) / FIFO serial I/F Ch.0 status register (0x300202)

Indicates the number of data in the receive data buffer (FIFO) that have not been read. When RXD0NUM is 0, it indicates that the receive data buffer contains 0 or 1 received data. When RXD0NUM is 1 to 3, it indicates that the receive data buffer contains 2 to 4 received data.

At initial reset, RXD0NUM is set to 0.

---

**TEND0:** Ch.0 transmit-completion flag (D5) / FIFO serial I/F Ch.0 status register (0x300202)

---

Indicates the transmission status.

- Read "1": During transmitting
- Read "0": End of transmission
- Write: Invalid

TEND0 goes "1" when data is being transmitted and goes "0" when the transmission has completed. When data is transmitted successively in clock-synchronized master mode or asynchronous mode, TEND0 maintains "1" until all data is transmitted (see Figure III.10.5 and Figure III.10.14). In clock-synchronized slave mode, TEND0 goes "0" every time 1-byte data is transmitted (see Figure III.10.6). At initial reset, TEND0 is set to "0" (end of transmission).

---

**FER0:** Ch.0 framing-error flag (D4) / FIFO serial I/F Ch.0 status register (0x300202)

---

Indicates whether a framing error occurred.

- Read "1": An error occurred
- Read "0": No error occurred
- Write "1": Invalid
- Write "0": Reset to "0"

The FER0 flag is an error flag indicating whether a framing error occurred. When an error has occurred, it is set to "1". A framing error occurs when data with a stop bit = "0" is received in the asynchronous mode. The FER0 flag is reset by writing "0". At initial reset, as well as when RXEN0 and TXEN0 both are set to "0", the FER0 flag is set to "0" (no error).

---

**PER0:** Ch.0 parity-error flag (D3) / FIFO serial I/F Ch.0 status register (0x300202)

---

Indicates whether a parity error occurred.

- Read "1": An error occurred
- Read "0": No error occurred
- Write "1": Invalid
- Write "0": Reset to "0"

The PER0 flag is an error flag indicating whether a parity error occurred. When an error has occurred, it is set to "1". Parity checks are valid only in the asynchronous mode with EPR0 set to "1" (parity added). This check is performed when the received data is transferred from the shift register to the receive data buffer. The PER0 flag is reset by writing "0". At initial reset, as well as when RXEN0 and TXEN0 both are set to "0", PER0 is set to "0" (no error).

---

**OER0:** Ch.0 overrun-error flag (D2) / FIFO serial I/F Ch.0 status register (0x300202)

---

Indicates whether an overrun error occurred.

- Read "1": An error occurred
- Read "0": No error occurred
- Write "1": Invalid
- Write "0": Reset to "0"

The OER0 flag is an error flag indicating whether an overrun error occurred. When an error has occurred, it is set to "1". An overrun error will occur if a new data is transferred to this serial interface when the receive data buffer is full and also the shift register contains received data. The OER0 flag is reset by writing "0". When this error occurs, the shift register is overwritten with the new received data and the receive data in the buffer is maintained as is. At initial reset, as well as when RXEN0 and TXEN0 both are set to "0", OER0 is set to "0" (no error).

**TDBE0: Ch.0 transmit data buffer status (D1) / FIFO serial I/F Ch.0 status register (0x300202)**

Indicates the status of the transmit data buffer.

Read "1": Not full  
 Read "0": Buffer full  
 Write: Invalid

TDBE0 is set to "1" when the transmit data buffer has a free space for transmit data to be written and reset to "0" when the transmit data buffer becomes full by writing transmit data.

Up to two transmit data can be written to the transmit data buffer.

At initial reset, TDBE0 is set to "1" (Not full).

**RDBF0: Ch.0 receive data buffer status (D0) / FIFO serial I/F Ch.0 status register (0x300202)**

Indicates the status of the receive data buffer.

Read "1": Not empty  
 Read "0": Buffer empty  
 Write: Invalid

RDBF0 is set to "1" when the receive data buffer contains one or more received data, and is reset to "0" when the receive data buffer becomes empty by reading all the received data.

At initial reset, RDBF0 is set to "0" (Buffer empty).

**TXEN0: Ch.0 transmit enable (D7) / FIFO serial I/F Ch.0 control register (0x300203)**

Enables this serial interface for transmit operations.

Write "1": Transmit enabled  
 Write "0": Transmit disabled  
 Read: Valid

When TXEN0 is set to "1", transmit operations are enabled. When TXEN0 is set to "0", transmit operations are disabled. Always make sure the TXEN0 = "0" before setting the transfer mode and other conditions.

Writing "0" to TXEN0 clears the transmit data buffer (FIFO) as well as disabling transmit operations.

At initial reset, TXEN0 is set to "0" (transmit disabled).

**RXEN0: Ch.0 receive enable (D6) / FIFO serial I/F Ch.0 control register (0x300203)**

Enables this serial interface for receive operations.

Write "1": Receive enabled  
 Write "0": Receive disabled  
 Read: Valid

When RXEN0 is set to "1", receive operations are enabled. When RXEN0 is set to "0", receive operations are disabled. Always make sure the RXEN0 = "0" before setting the transfer mode and other conditions.

Writing "0" to RXEN0 clears the receive data buffer (FIFO) as well as disabling receive operations.

At initial reset, RXEN0 is set to "0" (receive disabled).

**EPR0: Ch.0 parity enable (D5) / FIFO serial I/F Ch.0 control register (0x300203)**

Selects a parity function.

Write "1": Parity added  
 Write "0": No parity added  
 Read: Valid

EPR0 is used to select whether receive data is to be checked for parity, and whether a parity bit is to be added to transmit data. When EPR0 is set to "1", the receive data is checked for parity. A parity bit is automatically added to the transmit data. When EPR0 is set to "0", parity is not checked and no parity bit is added.

The parity function is only valid in the asynchronous mode. Settings of EPR0 have no effect in the clock-synchronized mode.

At initial reset, EPR0 becomes indeterminate.

### III-10 PERIPHERAL BLOCK: SERIAL INTERFACE with FIFO

---

**PMD0:** Ch.0 parity mode selection (D4) / FIFO serial I/F Ch.0 control register (0x300203)

---

Selects an odd or even parity.

Write "1": Odd parity  
Write "0": Even parity  
Read: Valid

Odd parity is selected by writing "1" to PMD0, and even parity is selected by writing "0". Parity check and the addition of a parity bit are only effective in asynchronous transfers in which EPR0 is set to "1". If EPR0 = "0", settings of PMD0 do not have any effect.

At initial reset, PMD0 becomes indeterminate.

---

**STPB0:** Ch.0 stop bit selection (D3) / FIFO serial I/F Ch.0 control register (0x300203)

---

Selects a stop-bit length during the performance of an asynchronous transfer.

Write "1": 2 bits  
Write "0": 1 bit  
Read: Valid

STPB0 is only valid in an asynchronous transfer. Two stop bits are selected by writing "1" to STPB0, and one stop bit is selected by writing "0". The start bit is fixed at 1 bit.

Setting of STPB0 is ignored during the performance of a clock-synchronized transfer.

At initial reset, STPB0 becomes indeterminate.

---

**SSCK0:** Ch.0 input clock selection (D2) / FIFO serial I/F Ch.0 control register (0x300203)

---

Selects the clock source for an asynchronous transfer.

Write "1": #FSCLK0 (external clock)  
Write "0": Internal clock (baud-rate timer)  
Read: Valid

During operation in the asynchronous mode, this bit is used to select the clock source between an internal clock (output by the baud-rate timer) and an external clock (input from the #FSCLK0 pin). An external clock is selected by writing "1" to this bit, and an internal clock is selected by writing "0".

At initial reset, SSCK0 becomes indeterminate.

---

**SMD01–SMD00:** Ch.0 transfer mode selection (D[1:0]) / FIFO serial I/F Ch.0 control register (0x300203)

---

Sets the transfer mode of the serial interface as shown in Table III.10.12 below.

Table III.10.12 Setting of Transfer Mode

SMD01	SMD00	Transfer mode
1	1	8-bit asynchronous mode
1	0	7-bit asynchronous mode
0	1	Clock-synchronized slave mode
0	0	Clock-synchronized master mode

The SMD0 bit can be read as well as written.

When using the IrDA interface, always be sure to set an asynchronous mode for the transfer mode.

At initial reset, SMD0 becomes indeterminate.



**SRDYCTL0: Ch.0 #ESRDY control (D7) / FIFO serial I/F Ch.0 IrDA register (0x300204)**

Selects a control method for the #FSRDY0 signal.

Write "1": High mask mode

Write "0": Normal output

Read: Valid

When SRDYCTL0 is set to "0", the #FSRDY0 signal is controlled normally and indicates ready to receive even if the receive data buffer is full. When SRDYCTL0 is set to "1", high-mask mode is selected. The following shows the #FSRDY0 controls in the clock-synchronized slave mode and master mode:

**Clock-synchronizes slave mode**

When the receive data buffer is full, the #FSRDY0 signal is forcibly fixed at high in order to suspend data transfer from the master device until the data in the buffer is read.

**Clock-synchronized master mode**

When the receive data buffer is full, the #FSRDY0 signal (low) from the slave device is ignored and the serial interface stops outputting the #FSCLK0 signal until the buffer data is read.

The high mask mode can avoid overrun errors.

When the receive data buffer is not full, normal receive operations are performed even if this function is enabled.

In the asynchronous mode, this bit is ignored as it does not use the #FSRDY0 signal.

At initial reset, SRDYCTL0 is set to "0" (normal output).

**FIFOINT01–FIFOINT0: Ch.0 receive FIFO level setting (D[6:5]) / FIFO serial I/F Ch.0 IrDA register (0x300204)**

Sets the number of data in the receive data buffer to generate a receive-buffer full interrupt.

Writing 0–3 to FIFOINT sets the number of data to 1–4.

When the number of data in the receive data buffer reaches the number specified here, the receive-buffer full interrupt factor flag FFSRX0 are set to "1".

At initial reset, FIFOINT0 is set to "0" (one data).

**DIVMD0: Sampling clock division ratio (D4) / FIFO serial I/F Ch.0 IrDA register (0x300204)**

Selects the division ratio of the sampling clock.

Write "1": 1/8

Write "0": 1/16

Read: Valid

Select the division ratio necessary to generate the sampling clock for asynchronous transfers. When DIVMD0 is set to "1", the sampling clock is generated from the input clock (output by the baud-rate timer or input from #FSCLK0) by dividing it by 8. When DIVMD0 is set to "0", the input clock is divided by 16.

At initial reset, DIVMD0 becomes indeterminate.

**IRTLO: Ch.0 IrDA output logic inversion (D3) / FIFO serial I/F Ch.0 IrDA register (0x300204)**

Inverts the logic of the IrDA output signal.

Write "1": Inverted

Write "0": Not inverted

Read: Valid

When using the IrDA interface, set the logic of the FSOUT0 output signal to suit the infrared-ray communication circuit that is connected external to the chip. If IRTLO is set to "1", a high pulse is output when the output data = "0" (held low-level when the output data = "1"). If IRTLO is set to "0", a low pulse is output when the output data = "0" (held high-level when the output data = "1").

At initial reset, IRTLO becomes indeterminate.

**IRRL0:** Ch.0 IrDA input logic inversion (D2) / FIFO serial I/F Ch.0 IrDA register (0x300204)

Inverts the logic of the IrDA input signal.

- Write "1": Inverted
- Write "0": Not inverted
- Read: Valid

When using the IrDA interface, set the logic of the signal that is input from an external infrared-ray communication circuit to the chip to suit the serial interface. If IRRL0 is set to "1", a high pulse is input as a logic "0". If IRRL0 is set to "0", a low pulse is input as a logic "0".

At initial reset, IRRL0 becomes indeterminate.

**IRMD01–IRMD00:** Ch.0 IrDA interface mode selection (D[1:0]) / FIFO serial I/F Ch.0 IrDA register (0x300204)

Selects the IrDA interface function.

Table III.10.13 IrDA Interface Setting

IRMD01	IRMD00	Interface mode
1	1	Do not set. (reserved)
1	0	IrDA 1.0 interface
0	1	Do not set. (reserved)
0	0	Normal interface

When using the IrDA interface function, write "10" to IRMD0 while setting to an asynchronous mode for the transfer mode. If the IrDA interface function is not to be used, write "00" to IRMD0.

At initial reset, IRMD0 becomes indeterminate.

**Note:** This selection must always be performed before the transfer mode and other conditions are set.

**BRTRUN:** Baud-rate timer RUN/STOP control (D0) / FIFO serial I/F Ch.0 baud-rate timer control register (0x300205)

Controls the counter's RUN/STOP states.

- Write "1": RUN
- Write "0": STOP
- Read: Valid

The baud-rate timer loads the reload data (BRTRD0[9:0]) to its counter and starts counting down when "1" is written to BRTRUN. The baud-rate timer stops counting when "0" is written to BRTRUN.

At initial reset, BRTRUN is set to "0" (STOP).

**BRTRD09–BRTRD00:** Baud-rate timer reload data / FIFO serial I/F Ch.0 baud-rate timer reload data register (D[1:0]/0x300207, D[7:0]/0x300206)

Set the initial counter value of the baud-rate timer.

The reload data set in this register is loaded into the counter, and the counter starts counting down beginning with this data, which is used as the initial count.

There are two cases in which the reload data is loaded into the counter: when the baud-rate timer starts by writing "1" to BRTRUN, or when data is automatically reloaded upon counter underflow.

At initial reset, BRTRD is set to "000H".

**BRTCD09–BRTCD00:** Baud-rate timer count data / FIFO serial I/F Ch.0 baud-rate timer count data register (D[1:0]/0x300209, D[7:0]/0x300208)

The baud-rate timer data can be read out from this register.

These bits function as buffers that retain the counter data when read out, enabling the data to be read out at any time.

At initial reset, BRTCD is set to "000H".

**PFSIO02–PFSIO00:** Ch.0 interrupt level (D[6:4]) / FIFO serial I/F Ch.0 interrupt priority register (0x402B0)

Sets the priority level of the FIFO serial-interface interrupt.

The interrupt priority level can be set for each channel in the range of 0 to 7.

At initial reset, PFSIO0 becomes indeterminate.

**EFERR0, EFSRX0, EFSTX0:** Ch.0 interrupt enable

(D3, D4, D5) / FIFO serial I/F Ch.0 interrupt enable register (0x402B1)

Enable or disable interrupt generation to the CPU.

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

The EFERR0, EFSRX0, and EFSTX0 bits are interrupt enable bits corresponding to receive-error, receive-buffer full, and transmit-buffer empty interrupt factors, respectively. The interrupts for which this bit is set to "1" are enabled, and the interrupts for which this bit is set to "0" are disabled.

At initial reset, all these bits are set to "0" (interrupts disabled).

**FFERR0, FFSRX0, FFSTX0:** Ch.0 interrupt factor flags

(D3, D4, D5) / FIFO serial I/F Ch.0 interrupt factor flag register (0x402B2)

Indicate the status of FIFO serial-interface interrupt generation.

**When read**

Read "1": An interrupt factor occurred

Read "0": No interrupt factor occurred

**When written using the reset-only method (default)**

Write "1": Flag is reset

Write "0": Invalid

**When written using the read/write method**

Write "1": Flag is set

Write "0": Flag is reset

The FFERR0, FFSRX0, and FFSTX0 flags are interrupt factor flags corresponding to receive-error, receive-buffer full, and transmit-buffer empty interrupts, respectively. The flag is set to "1" when each interrupt factor occurs. A transmit-buffer empty interrupt factor occurs when the last data in the transmit data buffer is transferred to the shift register.

A receive-buffer full interrupt factor occurs when the number of data specified with FIFOINT0[1:0] has been received in the receive data buffer.

A receive-error interrupt factor occurs when a parity, framing, or overrun error is detected during reception of data. At this time, if the following conditions are met, an interrupt to the CPU is generated:

1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The PSR's IE bit is set to "1" (interrupts enabled).
4. The set value of the corresponding interrupt priority register is higher than the CPU interrupt level (IL).

When using the receive-buffer full or transmit-buffer empty interrupt factor as an IDMA request, the fact that the above conditions are met does not necessarily mean that an interrupt request to the CPU has been output simultaneously when an interrupt factor occurs. An interrupt is generated under the above conditions upon completion of the data transfer by IDMA, provided that interrupts are enabled by settings on the IDMA side.

The interrupt factor flag is set to "1" whenever an interrupt factor occurs, regardless of the settings of the interrupt-enable and interrupt priority registers.

If the next interrupt is to be accepted following the occurrence of an interrupt, it is necessary that the interrupt factor flag be reset, and that the PSR be set up again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).

The interrupt factor flag can only be reset by writing to it in the software. Note that if the PSR is set up again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.

At initial reset, all of these flags become indeterminate, so be sure to reset them in the software.

**RFSRX0, RFSTX0:** Ch.0 IDMA request (D2, D3) / FIFO serial I/F Ch.0 IDMA request register (0x402B3)

Specifies whether to invoke IDMA when an interrupt factor occurs.

**When using the set-only method (default)**

Write "1": IDMA request

Write "0": Not changed

Read: Valid

**When using the read/write method**

Write "1": IDMA request

Write "0": Interrupt request

Read: Valid

The RFSRX0 and RFSTX0 bits are IDMA request bits corresponding to receive-buffer full and transmit-buffer empty interrupt factors, respectively. If the bit is set to "1", IDMA is invoked when an interrupt factor occurs, thus performing a programmed data transfer. If this bit is set to "0", normal interrupt processing is performed, without invoking IDMA.

For details on IDMA, refer to "IDMA (Intelligent DMA)".

At initial reset, these bits are set to "0" (interrupt request).

**DEFSRX0, DEFFSTX0:** Ch.0 IDMA enable (D2, D3) / FIFO serial I/F Ch.0 IDMA enable register (0x402B4)

Enables IDMA transfer by means of an interrupt factor.

**When using the set-only method (default)**

Write "1": IDMA enabled

Write "0": Not changed

Read: Valid

**When using the read/write method**

Write "1": IDMA enabled

Write "0": IDMA disabled

Read: Valid

The DEFSRX0 and DEFFSTX0 bits are IDMA enable bits corresponding to receive-buffer full and transmit-buffer empty interrupt factors, respectively. If the bit is set to "1", the IDMA request by the interrupt factor is enabled. If the bit is set to "0", the IDMA request is disabled.

At initial reset, these bits are set to "0" (IDMA disabled).

## Programming Notes

- (1) Before setting various serial-interface parameters, make sure the transmit and receive operations are disabled (TXEN0 = RXEN0 = "0").
- (2) When the serial interface is transmitting or receiving data, do not set TXEN0 or RXEN0 to "0", and do not execute the slp instruction.
- (3) In clock-synchronized transfers, the mode of communication is half-duplex, in which the clock line is shared between the transmit and receive units. Therefore, RXEN0 and TXEN0 cannot be enabled simultaneously.
- (4) After an initial reset, the interrupt factor flag becomes indeterminate. To prevent generation of an unwanted interrupt or IDMA request, reset this flag in the program.
- (5) If a receive error occurs, the receive-error interrupt and receive-buffer full interrupt factors may occur simultaneously. However, since the receive-error interrupt has priority over the receive-buffer full interrupt, the receive-error interrupt is processed first. Therefore, it is necessary to reset the receive-buffer full interrupt factor flag through the use of the receive-error interrupt processing routine.
- (6) To prevent the regeneration of interrupts due to the same factor following the occurrence of an interrupt, always be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.
- (7) Follow the procedure described below to initialize the serial interface.

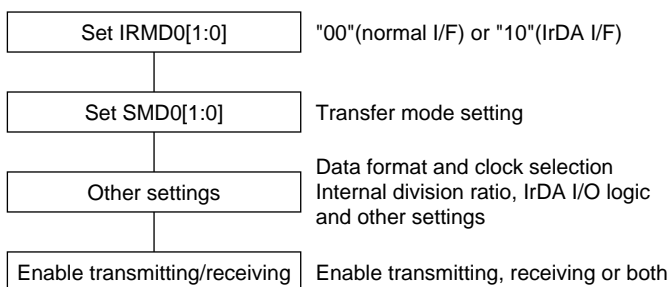


Figure III.10.20 Serial Interface Initialize Procedure

- (8) When transmitting data in the clock-synchronized master mode, transmit data is written to the transmit data register after the initial setting is performed following the flow in item (7). However, the clock generated by the baud-rate timer must be supplied to the serial interface (at least one underflow has had to have occurred in the baud-rate timer) before this writing. Otherwise, 0xFF will be transmitted prior to the written data.
- (9) The maximum transfer rate of the serial interface is limited to 1 Mbps.
- (10) If the receive circuit is stopped during reception, set both transmission and reception to the disabled status.
- (11) When performing data transfer in the clock-synchronized mode, the division ratio of the reload data for the baud-rate timer should be set so that the baud-rate is 1/4 of the system clock frequency or lower.
- (12) When the transmit-enable bit TXEN0 is set to "0" to disable transmit operations, the transmit data buffer (FIFO) is cleared (initialized). Similarly, when the receive-enable bit RXEN0 is set to "0" to disable receive operations, the receive data buffer (FIFO) is cleared (initialized). Therefore, make sure that the buffer does not contain any data waiting for transmission or reading before writing "0" to these bits.
- (13) The baud-rate timer input clock (peripheral circuit operating clock) is supplied to this serial interface when the PSCON (D5) / Power control register (0x40180) is set to "1". Be aware that the clock is not supplied to the baud-rate timer when PSCON = "0".
- (14) The baud-rate timer input clock (peripheral circuit operating clock) is selected between the OSC3/PLL clock and the OSC1 clock with the PSCDT0 (D0) / Prescaler clock select register (0x40181).

THIS PAGE IS BLANK.

## III-11 INPUT/OUTPUT PORTS

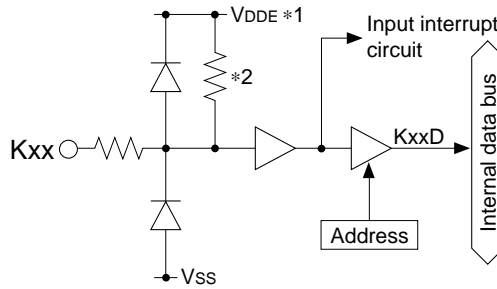
The Peripheral Block has a total of 42 input/output ports. Although each pin is used for input/output from/to the internal peripheral circuits, some pins can be used as general-purpose input/output ports unless they are used for the peripheral circuits.

### Input Ports (K Ports)

#### Structure of Input Port

The Peripheral Block contains 13 bits of input ports (K50 to K54, K60 to K67).

Figure III.11.1 shows the structure of a typical input port.



\*1 AV<sub>DDE</sub> for K60–K67

\*2 A pull-up resistor is provided depending on model.

Figure III.11.1 Structure of Input Port

Each input-port pin is connected directly to the internal data bus via a three-state buffer. The state of the input signal when read at an input port is directly taken into the internal circuit as data.

The power supply for the K60 to K67 input buffers is AV<sub>DDE</sub>. Therefore, when these ports are used as DC input ports, the high level must be AV<sub>DDE</sub>, and the low level V<sub>SS</sub>.

If there is a potential difference between AV<sub>DDE</sub> and V<sub>DDE</sub>, in particular, if the level from outside is V<sub>DDE</sub>, a current may flow in the input buffer (when AV<sub>DDE</sub> > V<sub>DDE</sub>) or between V<sub>DDE</sub> and AV<sub>DDE</sub> (when AV<sub>DDE</sub> < V<sub>DDE</sub>). Therefore, if these ports are not used, when the input level is fixed externally, it should be fixed at V<sub>SS</sub> or AV<sub>DDE</sub>.

In the model that has built-in pull-up resistors for the K60–K67 ports, the ports are pulled up to AV<sub>DDE</sub>.

**Note:** Depending on the model, a pull-up resistor may be provided for each input pin and it can be enabled/disabled by software control. Refer to the "Pin Status Control Registers" chapter for how to control the pull-up resistor.

## Input-Port Pins

The input pins concurrently serve as the input pins for peripheral circuits, as shown in Table III.11.1. Whether they are used as input ports or for peripheral circuits can be set bit-for-bit using a function select register. All pins not used for peripheral circuits can be used as general-purpose input ports that have an interrupt function.

Table III.11.1 Input Pins

Pin name	I/O	Function	Function select bit
<b>K50</b> (#DMAREQ0)	I	Input port / High-speed DMA request 0	CFK50(D0)/K5 function select register(0x402C0)
<b>K51</b> (#DMAREQ1)	I	Input port / High-speed DMA request 1	CFK51(D1)/K5 function select register(0x402C0)
<b>K52</b> (#ADTRG)	I	Input port / AD converter trigger	CFK52(D2)/K5 function select register(0x402C0)
<b>K53</b> (#DMAREQ2)	I	Input port / High-speed DMA request 2	CFK53(D3)/K5 function select register(0x402C0)
<b>K54</b> (#DMAREQ3)	I	Input port / High-speed DMA request 3	CFK54(D4)/K5 function select register(0x402C0)
<b>K60</b> (AD0)	I	Input port / AD converter input 0	CFK60(D0)/K6 function select register(0x402C3)
<b>K61</b> (AD1)	I	Input port / AD converter input 1	CFK61(D1)/K6 function select register(0x402C3)
<b>K62</b> (AD2)	I	Input port / AD converter input 2	CFK62(D2)/K6 function select register(0x402C3)
<b>K63</b> (AD3)	I	Input port / AD converter input 3	CFK63(D3)/K6 function select register(0x402C3)
<b>K64</b> (AD4)	I	Input port / AD converter input 4	CFK64(D4)/K6 function select register(0x402C3)
<b>K65</b> (AD5)	I	Input port / AD converter input 5	CFK65(D5)/K6 function select register(0x402C3)
<b>K66</b> (AD6)	I	Input port / AD converter input 6	CFK66(D6)/K6 function select register(0x402C3)
<b>K67</b> (AD7)	I	Input port / AD converter input 7	CFK67(D7)/K6 function select register(0x402C3)

At cold start, all pins are set for input ports Kxx (function select register CFKxx = "0"). When these pins are used for the internal peripheral circuits, write "1" to CFKxx. For details on pin functions in this case, refer to the description of each peripheral circuit in this manual.

At hot start, the pins retain their state from prior to the reset.

When the ports set for A/D converter input are read, the value obtained is always "0".

## Notes on Use

The input buffers of the K60 to K67 ports use AV<sub>DDE</sub> (power voltage for A/D converter) as their power source.

Furthermore, the pull-up resistors of these ports are connected to AV<sub>DDE</sub>. Therefore, the following precautions must be taken.

- 1) When using K60–K67 as general-purpose input ports, the voltage input to the port must be high level = AV<sub>DDE</sub> and low level = V<sub>SS</sub>.
- 2) When using V<sub>DDE</sub> as high level similar to other ports, V<sub>DDE</sub> must be the same voltage level as AV<sub>DDE</sub>. If the input V<sub>DDE</sub> level is lower than the AV<sub>DDE</sub> level, current flows in the input buffer, or if the input V<sub>DDE</sub> level is higher than the AV<sub>DDE</sub> level, current flows from the V<sub>DDE</sub> power supply to the AV<sub>DDE</sub> power supply.
- 3) To fix the input level externally when the port is not used, the input pin should be connected to V<sub>SS</sub> or AV<sub>DDE</sub>.



## I/O Memory of Input Ports

Table III.11.2 shows the control bits of the input ports.

Table III.11.2 Control Bits of Input Ports

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
K5 function select register	00402C0 (B)	D7-5	—	reserved	—		—	—	0 when being read.	
		D4	<b>CFK54</b>	K54 function selection	1	#DMAREQ3	0	K54	0	R/W
		D3	<b>CFK53</b>	K53 function selection	1	#DMAREQ2	0	K53	0	R/W
		D2	<b>CFK52</b>	K52 function selection	1	#ADTRG	0	K52	0	R/W
		D1	<b>CFK51</b>	K51 function selection	1	#DMAREQ1	0	K51	0	R/W
		D0	<b>CFK50</b>	K50 function selection	1	#DMAREQ0	0	K50	0	R/W
K5 input port data register	00402C1 (B)	D7-5	—	reserved	—		—	—	0 when being read.	
		D4	<b>K54D</b>	K54 input port data	1	High	0	Low	—	R
		D3	<b>K53D</b>	K53 input port data					—	R
		D2	<b>K52D</b>	K52 input port data					—	R
		D1	<b>K51D</b>	K51 input port data					—	R
		D0	<b>K50D</b>	K50 input port data					—	R
K6 function select register	00402C3 (B)	D7	<b>CFK67</b>	K67 function selection	1	AD7	0	K67	0	R/W
		D6	<b>CFK66</b>	K66 function selection	1	AD6	0	K66	0	R/W
		D5	<b>CFK65</b>	K65 function selection	1	AD5	0	K65	0	R/W
		D4	<b>CFK64</b>	K64 function selection	1	AD4	0	K64	0	R/W
		D3	<b>CFK63</b>	K63 function selection	1	AD3	0	K63	0	R/W
		D2	<b>CFK62</b>	K62 function selection	1	AD2	0	K62	0	R/W
		D1	<b>CFK61</b>	K61 function selection	1	AD1	0	K61	0	R/W
		D0	<b>CFK60</b>	K60 function selection	1	AD0	0	K60	0	R/W
K6 input port data register	00402C4 (B)	D7	<b>K67D</b>	K67 input port data	1	High	0	Low	—	R
		D6	<b>K66D</b>	K66 input port data					—	R
		D5	<b>K65D</b>	K65 input port data					—	R
		D4	<b>K64D</b>	K64 input port data					—	R
		D3	<b>K63D</b>	K63 input port data					—	R
		D2	<b>K62D</b>	K62 input port data					—	R
		D1	<b>K61D</b>	K61 input port data					—	R
		D0	<b>K60D</b>	K60 input port data					—	R

**CFK54–CFK50:** K5[4:0] function selection (D[4:0]) / K5 function select register (0x402C0)

**CFK67–CFK60:** K6[7:0] function selection (D[7:0]) / K6 function select register (0x402C3)

Selects the function of each input-port pin.

Write "1": Used for peripheral circuit

Write "0": Input port pin

Read: Invalid

When a bit of the CFK register is set to "1", the corresponding pin is set for use with the peripheral circuit (see Table III.11.1). The pins for which register bits are set to "0" can be used as general-purpose input ports.

At cold start, CFK is set to "0" (input port). At hot start, CFK retains its state from prior to the initial reset.

**K54D–K50D:** K5[4:0] input port data (D[4:0]) / K5 input port data register (0x402C1)

**K67D–K60D:** K6[7:0] input port data (D[7:0]) / K6 input port data register (0x402C4)

The input data on each input port pin can be read from this register.

Read "1": High level

Read "0": Low level

Write: Invalid

The pin voltage of each input port can be read out "1" directly when the voltage is high (V<sub>DD</sub>) or "0" when the voltage is low (V<sub>SS</sub>) respectively.

Since this register is a read-only register, writing to the register is ignored.

When the ports set for A/D converter input are read, the value obtained is always "0".

## I/O Ports (P Ports)

### Structure of I/O Port

The Peripheral Block contains 29 bits of I/O ports (P00 to P07, P10 to P16, P20 to P27, P30 to P35) that can be directed for input or output through the use of a program.

Figure III.11.2 shows the structure of a typical I/O port.

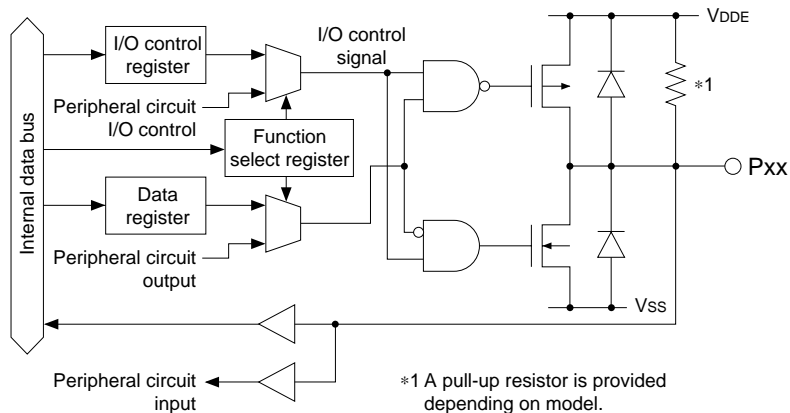


Figure III.11.2 Structure of I/O Port

**Note:** Depending on the model, a pull-up resistor may be provided for each pin and it can be enabled/disabled by software control. Refer to the "Pin Status Control Registers" chapter for how to control the pull-up resistor.

### I/O Port Pins

The I/O ports concurrently serve as the input/output pins for peripheral circuits, as shown in Table III.11.3. Whether they are used as I/O ports or for peripheral circuits can be set bit-for-bit using the function-select registers and function extension registers. All pins not used for peripheral circuits can be used as general-purpose I/O ports.

Table III.11.3 I/O Pins

Pin name	I/O	Function	Function select bit
P00(SIN0)	I/O	I/O port / Serial IF Ch.0 data input	CFP00(D0)/P0 function select register(0x402D0)
P01(SOUT0)	I/O	I/O port / Serial IF Ch.0 data output	CFP01(D1)/P0 function select register(0x402D0)
P02(#SCLK0)	I/O	I/O port / Serial IF Ch.0 clock input/output	CFP02(D2)/P0 function select register(0x402D0)
P03(#SRDY0)	I/O	I/O port / Serial IF Ch.0 ready input/output	CFP03(D3)/P0 function select register(0x402D0)
P04(SIN1/#DMAACK2)	I/O	I/O port / Serial IF Ch.1 data input / #DMAACK2 output (Ex)	CFP04(D4)/P0 function select register(0x402D0) CFEX4(D4)/Port function extension register(0x402DF)
P05(SOUT1/#DMAEND2)	I/O	I/O port / Serial IF Ch.1 data output / #DMAEND2 output (Ex)	CFP05(D5)/P0 function select register(0x402D0) CFEX5(D5)/Port function extension register(0x402DF)
P06(#SCLK1/#DMAACK3)	I/O	I/O port / Serial IF Ch.1 clock input/output / #DMAACK3 output (Ex)	CFP06(D6)/P0 function select register(0x402D0) CFEX6(D6)/Port function extension register(0x402DF)
P07(#SRDY1/#DMAEND3)	I/O	I/O port / Serial IF Ch.1 ready input/output / #DMAEND3 output (Ex)	CFP07(D7)/P0 function select register(0x402D0) CFEX7(D7)/Port function extension register(0x402DF)
DST0(P10/EXCL0/T8UF0)	I/O	DST0 output (Ex) / I/O port / 16-bit timer 0 event counter input (I) / 8-bit timer 0 output (O)	CFP10(D0)/P1 function select register(0x402D4) CFEX1(D1)/Port function extension register(0x402DF)
DST1(P11/EXCL1/T8UF1)	I/O	DST1 output (Ex) / I/O port / 16-bit timer 1 event counter input (I) / 8-bit timer 1 output (O)	CFP11(D1)/P1 function select register(0x402D4) CFEX1(D1)/Port function extension register(0x402DF)
DST2(P12/EXCL2/T8UF2)	I/O	DST2 output (Ex) / I/O port / 16-bit timer 2 event counter input (I) / 8-bit timer 2 output (O)	CFP12(D2)/P1 function select register(0x402D4) CFEX0(D0)/Port function extension register(0x402DF)
DPCO(P13/EXCL3/T8UF3)	I/O	DPCO output (Ex) / I/O port / 16-bit timer 3 event counter input (I) / 8-bit timer 3 output (O)	CFP13(D3)/P1 function select register(0x402D4) CFEX1(D1)/Port function extension register(0x402DF)
DCLK(P14/FOSC1)	I/O	DCLK output (Ex) / I/O port / Low-speed (OSC1) clock output	CFP14(D4)/P1 function select register(0x402D4) CFEX0(D0)/Port function extension register(0x402DF)
P15(EXCL4/#DMAEND0/#SCLK3)	I/O	I/O port / 16-bit timer 4 event counter input (I) / #DMAEND0 output (O) / Serial IF Ch.3 clock input/output	CFP15(D5)/P1 function select register(0x402D4)
P16(EXCL5/#DMAEND1/SOUT3)	I/O	I/O port / 16-bit timer 5 event counter input (I) / #DMAEND1 output (O) / Serial IF Ch.3 data output	CFP16(D6)/P1 function select register(0x402D4)

Pin name	I/O	Function	Function select bit
P20(#DRD)	I/O	I/O port / #DRD output	CFP20(D0)/P2 function select register(0x402D8)
P21(#DWE/#GAAS)	I/O	I/O port / #DWE output / GA address strobe output (Ex)	CFP21(D1)/P2 function select register(0x402D8) CFEX2(D2)/Port function extension register(0x402DF)
P22(TM0)	I/O	I/O port / 16-bit timer 0 output	CFP22(D2)/P2 function select register(0x402D8)
P23(TM1)	I/O	I/O port / 16-bit timer 1 output	CFP23(D3)/P2 function select register(0x402D8)
P24(TM2/#SRDY2)	I/O	I/O port / 16-bit timer 2 output / Serial IF Ch.2 ready input/output	CFP24(D4)/P2 function select register(0x402D8)
P25(TM3/#SCLK2)	I/O	I/O port / 16-bit timer 3 output / Serial IF Ch.2 clock input/output	CFP25(D5)/P2 function select register(0x402D8)
P26(TM4/SOUT2)	I/O	I/O port / 16-bit timer 4 output / Serial IF Ch.2 data output	CFP26(D6)/P2 function select register(0x402D8)
P27(TM5/SIN2)	I/O	I/O port / 16-bit timer 5 output / Serial IF Ch.2 data input	CFP27(D7)/P2 function select register(0x402D8)
P30(#WAIT/#CE4&5)	I/O	I/O port / #WAIT input (I) / #CE4&5 output (O)	CFP30(D0)/P3 function select register(0x402DC)
P31(#BUSGET/#GARD)	I/O	I/O port / #BUSGET output / GA read signal output (Ex)	CFP31(D1)/P3 function select register(0x402DC) CFEX3(D3)/Port function extension register(0x402DF)
P32(#DMAACK0/#SRDY3)	I/O	I/O port / #DMAACK0 output / Serial IF Ch.3 ready input/output	CFP32(D2)/P3 function select register(0x402DC)
P33(#DMAACK1/SIN3)	I/O	I/O port / #DMAACK1 output / Serial IF Ch.3 data input	CFP33(D3)/P3 function select register(0x402DC)
P34(#BUSREQ/#CE6)	I/O	I/O port / #BUSREQ input (I) / #CE6 output (O)	CFP34(D4)/P3 function select register(0x402DC)
P35(#BUSACK)	I/O	I/O port / #BUSACK output	CFP35(D5)/P3 function select register(0x402DC)

(I): Input mode, (O): Output mode, (Ex): Extended function

At cold start, all pins are set for I/O ports Pxx (function select register CFPxx = "0"). When these pins are used for the internal peripheral circuits, write "1" to CFPxx. For details on pin functions in this case, refer to the description of each peripheral circuit in this manual. At hot start, the pins retain their state from prior to the reset.

In addition to being an I/O port, the P10–P13, P15–P16, P30 and P34 pins are shared with two types (three types for P10–P13) of peripheral circuits. The type of peripheral circuit for which these pins are used is determined by the direction (input or output) in which the pin is set using an I/O control register, as will be described later.

The P04–P07, P10–P14, P21 and P31 ports have extended functions indicated with (Ex) in the table. They can be selected by writing "1" to CFEXx / Port function extension register (0x402DF).

The setting of CFEXx has priority over the CFPxx. At cold start, CFEX1 and CFEX0 are set to "1", so the P10–P14 pins are set for debug signal outputs.

## I/O Control Register and I/O Modes

The I/O ports are directed for input or output modes by writing data to an I/O control register corresponding to each port bit.

P07–P00 I/O control: IOC0[7:0] (D[7:0]) / P0 I/O control register (0x402D2)

P16–P10 I/O control: IOC1[6:0] (D[6:0]) / P1 I/O control register (0x402D6)

P27–P20 I/O control: IOC2[7:0] (D[7:0]) / P2 I/O control register (0x402DA)

P35–P30 I/O control: IOC3[5:0] (D[5:0]) / P3 I/O control register (0x402DE)

To set an I/O port for input, write "0" to the I/O control bit. I/O ports set for input mode are placed in the high-impedance state, and thus function as input ports. In the input mode, the state of the input pin is read directly, so the data is "1" when the pin state is high (V<sub>DD</sub> level) or "0" when the pin state is low (V<sub>SS</sub> level).

Even in the input mode, data can be written to the data register without affecting the pin state.

To set an I/O port for output, write "1" to the I/O control bit. I/O port set for output function as output ports. When the port output data is "1", the port outputs a high level (V<sub>DD</sub> level); when the data is "0", the port outputs a low level (V<sub>SS</sub> level).

At cold start, the I/O control register is set to "0" (input mode).

At hot start, the pins retain their state from prior to the reset.

**Note:** If pins P10–P14, P15–P16, P30 and P34 are set for use with peripheral circuits, their pin functions vary depending on the input/output direction control by the IOC1x register.

### I/O Memory of I/O Ports

Table III.11.4 shows the control bits of the I/O ports.

Table III.11.4 Control Bits of I/O Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
P0 function select register	00402D0 (B)	D7	CFP07	P07 function selection	1 #SRDY1	0 P07	0	R/W	Extended functions (0x402DF)		
		D6	CFP06	P06 function selection	1 #SCLK1	0 P06	0	R/W			
		D5	CFP05	P05 function selection	1 SOUT1	0 P05	0	R/W			
		D4	CFP04	P04 function selection	1 SIN1	0 P04	0	R/W			
		D3	CFP03	P03 function selection	1 #SRDY0	0 P03	0	R/W			
		D2	CFP02	P02 function selection	1 #SCLK0	0 P02	0	R/W	Extended functions (0x300040)		
		D1	CFP01	P01 function selection	1 SOUT0	0 P01	0	R/W			
		D0	CFP00	P00 function selection	1 SIN0	0 P00	0	R/W			
P0 I/O port data register	00402D1 (B)	D7	P07D	P07 I/O port data	1	High	0	Low	0	R/W	
		D6	P06D	P06 I/O port data					0	R/W	
		D5	P05D	P05 I/O port data					0	R/W	
		D4	P04D	P04 I/O port data					0	R/W	
		D3	P03D	P03 I/O port data					0	R/W	
		D2	P02D	P02 I/O port data					0	R/W	
		D1	P01D	P01 I/O port data					0	R/W	
		D0	P00D	P00 I/O port data					0	R/W	
P0 I/O control register	00402D2 (B)	D7	IOC07	P07 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D6	IOC06	P06 I/O control					0	R/W	
		D5	IOC05	P05 I/O control					0	R/W	
		D4	IOC04	P04 I/O control					0	R/W	
		D3	IOC03	P03 I/O control					0	R/W	
		D2	IOC02	P02 I/O control					0	R/W	
		D1	IOC01	P01 I/O control					0	R/W	
		D0	IOC00	P00 I/O control					0	R/W	
P1 function select register	00402D4 (B)	D7	–	reserved	–		–	–	0 when being read.		
		D6	CFP16	P16 function selection	1 EXCL5 #DMAEND1	0 P16	0	R/W	Extended functions (0x402D7)		
		D5	CFP15	P15 function selection	1 EXCL4 #DMAEND0	0 P15	0	R/W			
		D4	CFP14	P14 function selection	1 FOSC1	0 P14	0	R/W	Extended functions (0x402DF)		
		D3	CFP13	P13 function selection	1 EXCL3 T8UF3	0 P13	0	R/W			
		D2	CFP12	P12 function selection	1 EXCL2 T8UF2	0 P12	0	R/W			
		D1	CFP11	P11 function selection	1 EXCL1 T8UF1	0 P11	0	R/W			
		D0	CFP10	P10 function selection	1 EXCL0 T8UF0	0 P10	0	R/W			
P1 I/O port data register	00402D5 (B)	D7	–	reserved	–		–	–	0 when being read.		
		D6	P16D	P16 I/O port data	1	High	0	Low	0	R/W	
		D5	P15D	P15 I/O port data					0	R/W	
		D4	P14D	P14 I/O port data					0	R/W	
		D3	P13D	P13 I/O port data					0	R/W	
		D2	P12D	P12 I/O port data					0	R/W	
		D1	P11D	P11 I/O port data					0	R/W	
		D0	P10D	P10 I/O port data					0	R/W	
P1 I/O control register	00402D6 (B)	D7	–	reserved					–		
		D6	IOC16	P16 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D5	IOC15	P15 I/O control					0	R/W	
		D4	IOC14	P14 I/O control					0	R/W	
		D3	IOC13	P13 I/O control					0	R/W	
		D2	IOC12	P12 I/O control					0	R/W	
		D1	IOC11	P11 I/O control					0	R/W	
		D0	IOC10	P10 I/O control					0	R/W	
Port SIO function extension register	00402D7 (B)	D7–4	–	reserved					–		
		D3	SSRDY3	Serial I/F Ch.3 SRDY selection	1 #SRDY3	0 P32/ #DMAACK0	0	R/W			
		D2	SSCLK3	Serial I/F Ch.3 SCLK selection	1 #SCLK3	0 P15/EXCL4/ #DMAEND0	0	R/W			
		D1	SSOUT3	Serial I/F Ch.3 SOUT selection	1 SOUT3	0 P16/EXCL5/ #DMAEND1	0	R/W			
		D0	SSIN3	Serial I/F Ch.3 SIN selection	1 SIN3	0 P33/ #DMAACK1	0	R/W			

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P2 function select register	00402D8 (B)	D7	CFP27	P27 function selection	1	TM5	0	P27	0	R/W	Ext. func.(0x402DB)
		D6	CFP26	P26 function selection	1	TM4	0	P26	0	R/W	
		D5	CFP25	P25 function selection	1	TM3	0	P25	0	R/W	
		D4	CFP24	P24 function selection	1	TM2	0	P24	0	R/W	
		D3	CFP23	P23 function selection	1	TM1	0	P23	0	R/W	
		D2	CFP22	P22 function selection	1	TM0	0	P22	0	R/W	
		D1	CFP21	P21 function selection	1	#DWE	0	P21	0	R/W	
		D0	CFP20	P20 function selection	1	#DRD	0	P20	0	R/W	
P2 I/O port data register	00402D9 (B)	D7	P27D	P27 I/O port data	1	High	0	Low	0	R/W	
		D6	P26D	P26 I/O port data					0	R/W	
		D5	P25D	P25 I/O port data					0	R/W	
		D4	P24D	P24 I/O port data					0	R/W	
		D3	P23D	P23 I/O port data					0	R/W	
		D2	P22D	P22 I/O port data					0	R/W	
		D1	P21D	P21 I/O port data					0	R/W	
		D0	P20D	P20 I/O port data					0	R/W	
P2 I/O control register	00402DA (B)	D7	IOC27	P27 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D6	IOC26	P26 I/O control					0	R/W	
		D5	IOC25	P25 I/O control					0	R/W	
		D4	IOC24	P24 I/O control					0	R/W	
		D3	IOC23	P23 I/O control					0	R/W	
		D2	IOC22	P22 I/O control					0	R/W	
		D1	IOC21	P21 I/O control					0	R/W	
		D0	IOC20	P20 I/O control					0	R/W	
Port SIO function extension register	00402DB (B)	D7-4	–	reserved				–	–	0 when being read.	
		D3	SSRDY2	Serial I/F Ch.2 SRDY selection	1	#SRDY2	0	P24/TM2	0	R/W	
		D2	SSCLK2	Serial I/F Ch.2 SCLK selection	1	#SCLK2	0	P25/TM3	0	R/W	
		D1	SSOUT2	Serial I/F Ch.2 SOUT selection	1	SOUT2	0	P26/TM4	0	R/W	
		D0	SSIN2	Serial I/F Ch.2 SIN selection	1	SIN2	0	P27/TM5	0	R/W	
P3 function select register	00402DC (B)	D7-6	–	reserved				–	–	0 when being read.	
		D5	CFP35	P35 function selection	1	#BUSACK	0	P35	0	R/W	Ext. func.(0x300047)
		D4	CFP34	P34 function selection	1	#BUSREQ #CE6	0	P34	0	R/W	
		D3	CFP33	P33 function selection	1	#DMAACK1	0	P33	0	R/W	Ext. func.(0x402D7)
		D2	CFP32	P32 function selection	1	#DMAACK0	0	P32	0	R/W	
		D1	CFP31	P31 function selection	1	#BUSGET	0	P31	0	R/W	Ext. func.(0x402DF)
		D0	CFP30	P30 function selection	1	#WAIT #CE4/#CE5	0	P30	0	R/W	
P3 I/O port data register	00402DD (B)	D7-6	–	reserved				–	–	0 when being read.	
		D5	P35D	P35 I/O port data	1	High	0	Low	0	R/W	
		D4	P34D	P34 I/O port data					0	R/W	
		D3	P33D	P33 I/O port data					0	R/W	
		D2	P32D	P32 I/O port data					0	R/W	
		D1	P31D	P31 I/O port data					0	R/W	
		D0	P30D	P30 I/O port data					0	R/W	
P3 I/O control register	00402DE (B)	D7-6	–	reserved							
		D5	IOC35	P35 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D4	IOC34	P34 I/O control					0	R/W	
		D3	IOC33	P33 I/O control					0	R/W	
		D2	IOC32	P32 I/O control					0	R/W	
		D1	IOC31	P31 I/O control					0	R/W	
		D0	IOC30	P30 I/O control					0	R/W	
Port function extension register	00402DF (B)	D7	CFEX7	P07 port extended function					1	#DMAEND3	
		D6	CFEX6	P06 port extended function	1	#DMAACK3	0	P06, etc.	0	R/W	
		D5	CFEX5	P05 port extended function	1	#DMAEND2	0	P05, etc.	0	R/W	
		D4	CFEX4	P04 port extended function	1	#DMAACK2	0	P04, etc.	0	R/W	
		D3	CFEX3	P31 port extended function	1	#GARD	0	P31, etc.	0	R/W	
		D2	CFEX2	P21 port extended function	1	#GAAS	0	P21, etc.	0	R/W	
		D1	CFEX1	P10, P11, P13 port extended function	1	DST0 DST1 DPCO	0	P10, etc. P11, etc. P13, etc.	1	R/W	
		D0	CFEX0	P12, P14 port extended function	1	DST2 DCLK	0	P12, etc. P14, etc.	1	R/W	

---

**P07D–P00D:** P0[7:0] I/O port data (D[7:0]) / P0 I/O port data register (0x402D1)

**P16D–P10D:** P1[6:0] I/O port data (D[6:0]) / P1 I/O port data register (0x402D5)

**P27D–P20D:** P2[7:0] I/O port data (D[7:0]) / P2 I/O port data register (0x402D9)

**P35D–P30D:** P3[5:0] I/O port data (D[5:0]) / P3 I/O port data register (0x402DD)

---

This register reads data from I/O-port pins or sets output data.

#### When writing data

Write "1": High level

Write "0": Low level

When an I/O port is set for output, the data written to it is directly output to the I/O port pin. If the data written to the port is "1", the port pin is set high (VDD and VDDE level); if the data is "0", the port pin is set low (VSS level). Even in the input mode, data can be written to the port data register.

#### When reading data

Read "1": High level

Read "0": Low level

The voltage level on the port pin is read out regardless of whether an I/O port is set for input or output mode. If the pin voltage is high (VDD and VDDE level), "1" is read out as input data; if the pin voltage is low (VSS level), "0" is read out as input data.

At cold start, all data bits are set to "0". At hot start, they retain their state from prior to the initial reset.

---

**IOC07–IOC00:** P0[7:0] port I/O control (D[7:0]) / P0 port I/O control register (0x402D2)

**IOC16–IOC10:** P1[6:0] port I/O control (D[6:0]) / P1 port I/O control register (0x402D6)

**IOC27–IOC20:** P2[7:0] port I/O control (D[7:0]) / P2 port I/O control register (0x402DA)

**IOC35–IOC30:** P3[5:0] port I/O control (D[5:0]) / P3 port I/O control register (0x402DE)

---

Directs an I/O port for input or output and indicates the I/O control signal value of the port.

#### When writing data

Write "1": Output mode

Write "0": Input mode

This I/O control register corresponds bit-for-bit to each I/O port. When an IOC bit is set to "1", the corresponding I/O port is directed for output; if it is set to "0", the I/O port is directed for input.

At cold start, all IOC bits are set to "0" (input). At hot start, IOC retains its state from prior to the initial reset.

If pins P10–P13, P15–P16, P30 and P34 are set for use with peripheral circuits, their pin functions vary depending on the input/output direction control by the IOC1x register.

#### When reading data

Read "1": I/O control signal (output)

Read "0": I/O control signal (input)

The I/O control signal value for the port pin is read from this register. When I/O port function is selected using the CFEX and CFP registers, the value written to the IOC register is read out as is. When peripheral function is selected, the read value depends on the peripheral circuit status and may not indicate the value written to the IOC register. However, the read values of the IOC bits for P10–P13, P15–P16, P30, and P34 are the same as the written value even if the peripheral function is selected.

---

**CFP07–CFP00:** P0[7:0] function selection (D[7:0]) / P0 function select register (0x402D0)

**CFP16–CFP10:** P1[6:0] function selection (D[6:0]) / P1 function select register (0x402D4)

**CFP27–CFP20:** P2[7:0] function selection (D[7:0]) / P2 function select register (0x402D8)

**CFP35–CFP30:** P3[5:0] function selection (D[5:0]) / P3 function select register (0x402DC)

---

Selects the function of each I/O port pin.

Write "1": Used for peripheral circuit

Write "0": I/O port pin

Read: Valid

When a bit of the CFP register is set to "1", the corresponding pin is set for use with peripheral circuits (see Table III.11.3). The pins for which register bits are set to "0" can be used as general-purpose I/O ports.

At cold start, CFP is set to "0" (I/O port). At hot start, CFP retains its state from prior to the initial reset.

---

**CFEX0:** P12, P14 function extension (D0) / Port function extension register (0x402DF)  
**CFEX1:** P10, P11, P13 function extension (D1) / Port function extension register (0x402DF)  
**CFEX2:** P21 function extension (D2) / Port function extension register (0x402DF)  
**CFEX3:** P31 function extension (D3) / Port function extension register (0x402DF)  
**CFEX4:** P04 function extension (D4) / Port function extension register (0x402DF)  
**CFEX5:** P05 function extension (D5) / Port function extension register (0x402DF)  
**CFEX6:** P06 function extension (D6) / Port function extension register (0x402DF)  
**CFEX7:** P07 function extension (D7) / Port function extension register (0x402DF)

---

Sets whether the function of an I/O-port pin is to be extended.

Write "1": Function-extended pin  
 Write "0": I/O-port/peripheral-circuit pin  
 Read: Valid

When CFEXx is set to "1", the corresponding pin is set to the extended function input/output pin. When CFEXx = "0", the corresponding CFP bit becomes effective.

At cold start, CFEX0 and CFEX1 are set to "1" (function-extended pin) and other bits are set to "0" (I/O-port/peripheral-circuit pin). At hot start, CFEX retains its state from prior to the initial reset.

---

**SSIN3:** Serial I/F Ch.3 SIN selection (D0) / Port SIO function extension register (0x402D7)

---

Switches the function of pin P33/#DMAACK1/SIN3.

Write "1": SIN3  
 Write "0": P33/#DMAACK1  
 Read: Valid

To use the pin as SIN3, set SSIN3 (D0 / 0x402D7) to "1" and CFP33 (D3 / 0x402DC) to "0".

To use the pin as P33 or #DMAACK1, set this bit to "0".

At cold start, SSIN3 is set to "0" (P33/#DMAACK1). At hot start, SSIN3 retains its state from prior to the initial reset.

---

**SSOUT3:** Serial I/F Ch.3 SOUT selection (D1) / Port SIO function extension register (0x402D7)

---

Switches the function of pin P16/EXCL5/#DMAEND1/SOUT3.

Write "1": SOUT3  
 Write "0": P16/EXCL5/#DMAEND1  
 Read: Valid

To use the pin as SOUT3, set SSOUT3 (D1 / 0x402D7) to "1" and CFP16 (D6 / 0x402D4) to "0".

To use the pin as P16, EXCL5, or #DMAEND1, set this bit to "0".

At cold start, SSOUT3 is set to "0" (P16/EXCL5/#DMAEND1). At hot start, SSOUT3 retains its state from prior to the initial reset.

---

**SSCLK3:** Serial I/F Ch.3 SCLK selection (D2) / Port SIO function extension register (0x402D7)

---

Switches the function of pin P15/EXCL4/#DMAEND0/#SCLK3.

Write "1": #SCLK3  
 Write "0": P15/EXCL4/#DMAEND0  
 Read: Valid

To use the pin as #SCLK3, set SSCLK3 (D2 / 0x402D7) to "1" and CFP15 (D5 / 0x402D4) to "0".

To use the pin as P15, EXCL4, or #DMAEND0, set this bit to "0".

At cold start, SSCLK3 is set to "0" (P15/EXCL4/#DMAEND0). At hot start, SSCLK3 retains its state from prior to the initial reset.

---

**SSRDY3:** Serial I/F Ch.3 SRDY selection (D3) / Port SIO function extension register (0x402D7)

---

Switches the function of pin P32/#DMAACK0/#SRDY3.

Write "1": #SRDY3

Write "0": P32/#DMAACK0

Read: Valid

To use the pin as #SRDY3, set SSRDY3 (D3 / 0x402D7) to "1" and CFP32 (D2 / 0x402DC) to "0".

To use the pin as P32 or #DMAACK0, set this bit to "0".

At cold start, SSRDY3 is set to "0" (P32/#DMAACK0). At hot start, SSRDY3 retains its state from prior to the initial reset.

---

**SSIN2:** Serial I/F Ch.2 SIN selection (D0) / Port SIO function extension register (0x402DB)

---

Switches the function of pin P27/TM5/SIN2.

Write "1": SIN2

Write "0": P27/TM5

Read: Valid

To use the pin as SIN2, set SSIN2 (D0 / 0x402DB) to "1" and CFP27 (D7 / 0x402D8) to "0".

To use the pin as P27 or TM5, set this bit to "0".

At cold start, SSIN2 is set to "0" (P27/TM5). At hot start, SSIN2 retains its state from prior to the initial reset.

---

**SSOUT2:** Serial I/F Ch.2 SOUT selection (D1) / Port SIO function extension register (0x402DB)

---

Switches the function of pin P26/TM4/SOUT2.

Write "1": SOUT2

Write "0": P26/TM4

Read: Valid

To use the pin as SOUT2, set SSOUT2 (D1 / 0x402DB) to "1" and CFP26 (D6 / 0x402D8) to "0".

To use the pin as P26 or TM4, set this bit to "0".

At cold start, SSOUT2 is set to "0" (P26/TM4). At hot start, SSOUT2 retains its state from prior to the initial reset.

---

**SSCLK2:** Serial I/F Ch.2 SCLK selection (D2) / Port SIO function extension register (0x402DB)

---

Switches the function of pin P25/TM3/#SCLK2.

Write "1": #SCLK2

Write "0": P25/TM3

Read: Valid

To use the pin as #SCLK2, set SSCLK2 (D2 / 0x402DB) to "1" and CFP25 (D5 / 0x402D8) to "0".

To use the pin as P25 or TM3, set this bit to "0".

At cold start, SSCLK2 is set to "0" (P25/TM3). At hot start, SSCLK2 retains its state from prior to the initial reset.

---

**SSRDY2:** Serial I/F Ch.2 SRDY selection (D3) / Port SIO function extension register (0x402DB)

---

Switches the function of pin P24/TM2/#SRDY2.

Write "1": #SRDY2

Write "0": P24/TM2

Read: Valid

To use the pin as #SRDY2, set SSRDY2 (D3 / 0x402DB) to "1" and CFP24 (D4 / 0x402D8) to "0".

To use the pin as P24 or TM2, set this bit to "0".

At cold start, SSRDY2 is set to "0" (P24/TM2). At hot start, SSRDY2 retains its state from prior to the initial reset.



## Input Interrupt

The input ports and the I/O ports support eight system of port input interrupts and two systems of key input interrupts.

### Port Input Interrupt

The port input interrupt circuit has eight interrupt systems (FPT7–FPT0) and a port can be selected for generating each interrupt factor.

The interrupt condition can also be selected from between input signal edge and input signal level.

Figure III.11.3 shows the configuration of the port input interrupt circuit.

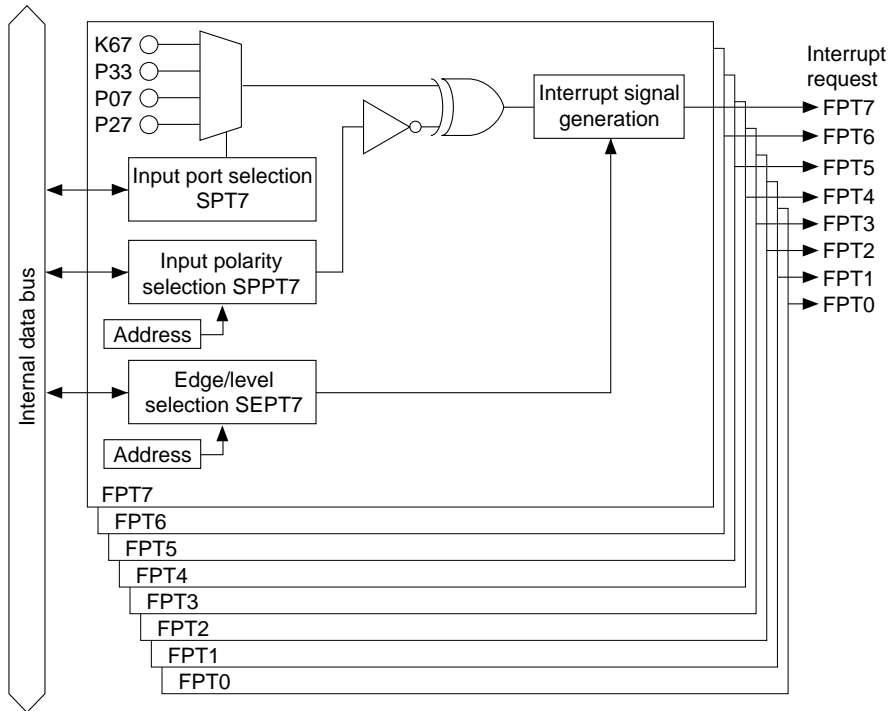


Figure III.11.3 Configuration of Port Input Interrupt Circuit

### Selecting input pins

The interrupt factors allows selection of an input pin from the four predefined pins independently.

Table III.11.5 shows the control bits and the selectable pins for each factor.

Table III.11.5 Selecting Pins for Port Input Interrupts

Interrupt factor	Control bit	SPT settings			
		11	10	01	00
FPT7	SPT7[1:0] (D[7:6])/Port input interrupt select register 2 (0x402C7)	P27	P07	P33	K67
FPT6	SPT6[1:0] (D[5:4])/Port input interrupt select register 2 (0x402C7)	P26	P06	P32	K66
FPT5	SPT5[1:0] (D[3:2])/Port input interrupt select register 2 (0x402C7)	P25	P05	P31	K65
FPT4	SPT4[1:0] (D[1:0])/Port input interrupt select register 2 (0x402C7)	P24	P04	K54	K64
FPT3	SPT3[1:0] (D[7:6])/Port input interrupt select register 1 (0x402C6)	P23	P03	K53	K63
FPT2	SPT2[1:0] (D[5:4])/Port input interrupt select register 1 (0x402C6)	P22	P02	K52	K62
FPT1	SPT1[1:0] (D[3:2])/Port input interrupt select register 1 (0x402C6)	P21	P01	K51	K61
FPT0	SPT0[1:0] (D[1:0])/Port input interrupt select register 1 (0x402C6)	P20	P00	K50	K60

### Conditions for port input-interrupt generation

Each port input interrupt can be generated by the edge or level of the input signal. The SEPTx bit of the edge/level select register (0x402C9) is used for this selection. When SEPTx is set to "1", the FPTx interrupt will be generated at the signal edge. When SEPTx is set to "0", the FPTx interrupt will be generated by the input signal level.

### III-11 PERIPHERAL BLOCK: INPUT/OUTPUT PORTS

Furthermore, the signal polarity can be selected using the SPPTx bit of the input parity select register (0x402C8). With these registers, the port input interrupt condition is decided as shown in Table III.11.6.

Table III.11.6 Port Input Interrupt Condition

SEPTx	SPPTx	FPTx interrupt condition
1	1	Rising edge
1	0	Falling edge
0	1	High level
0	0	Low level

When the input signal goes to the selected status, the interrupt factor flag FP is set to "1" and, if other interrupt conditions set by the interrupt controller are met, an interrupt is generated.

### Key Input Interrupt

The key input interrupt circuit has two interrupt systems (FPK1 and FPK0) and a port group can be selected for generating each interrupt factor.

The interrupt condition can also be set by software.

Figure III.11.4 shows the configuration of the port input interrupt circuit.

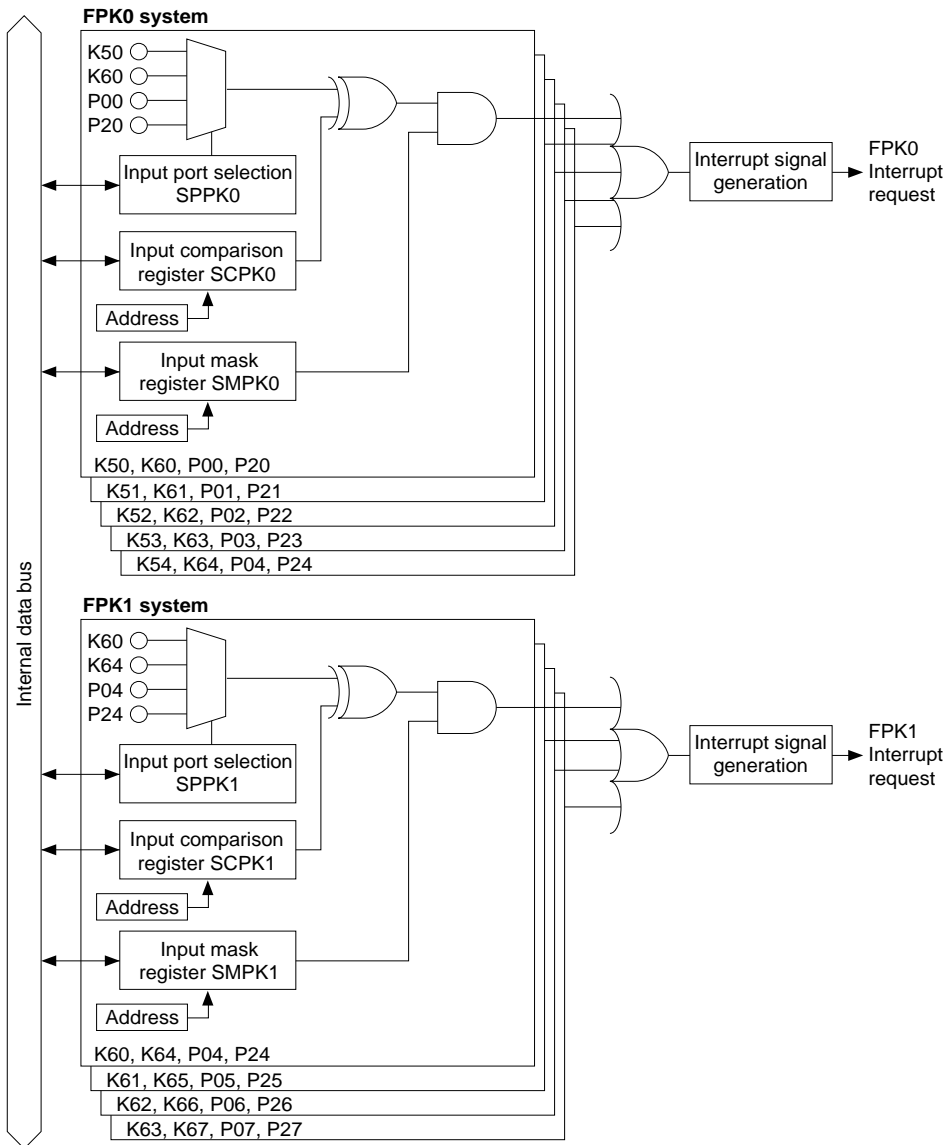


Figure III.11.4 Configuration of Key Input Interrupt Circuit

### Selecting input pins

For the FPK1 interrupt system, a four-bit input pin group can be selected from the four predefined groups. For the FPK0 system, a five-bit input pin group can be selected.

Table III.11.7 shows the control bits and the selectable groups for each factor.

Table III.11.7 Selecting Pins for Key Input Interrupts

Interrupt factor	Control bit	SPPK settings			
		11	10	01	00
FPK1	SPPK1[1:0] (D[3:2])/Key input interrupt select register (0x402CA)	P2[7:4]	P0[7:4]	K6[7:4]	K6[3:0]
FPK0	SPPK0[1:0] (D[1:0])/Key input interrupt select register (0x402CA)	P2[4:0]	P0[4:0]	K6[4:0]	K5[4:0]

### Conditions for key input-interrupt generation

The key input interrupt circuit has two input mask registers (SMPK0[4:0] for FPK0 and SMPK1[3:0] for FPK1) and two input comparison registers (SCPK0[4:0] for FPK0 and SCPK0[3:0] for FPK1) to set input-interrupt conditions.

The input mask register SMPK is used to mask the input pin that is not used for an interrupt. This register masks each input pin, whereas the interrupt enable register of the interrupt controller masks the interrupt factor for each interrupt group.

The input comparison register SCPK is used to select whether an interrupt for each input port is to be generated at the rising or falling edge of the input.

A change in state occurs so that the input pin enabled for interrupt by the interrupt mask register SMPK and the content of the input comparison register SCPK become unmatched after being matched, the interrupt factor flag FK is set to "1" and, if other interrupt conditions are met, an interrupt is generated.

Figure III.11.5 shows cases in which a FPK0 interrupt is generated. Here, it is assumed that the K5[4:0] pins are selected for the input-pin group and the control register of the interrupt controller is set so as to enable generation of a FPK0 interrupt.

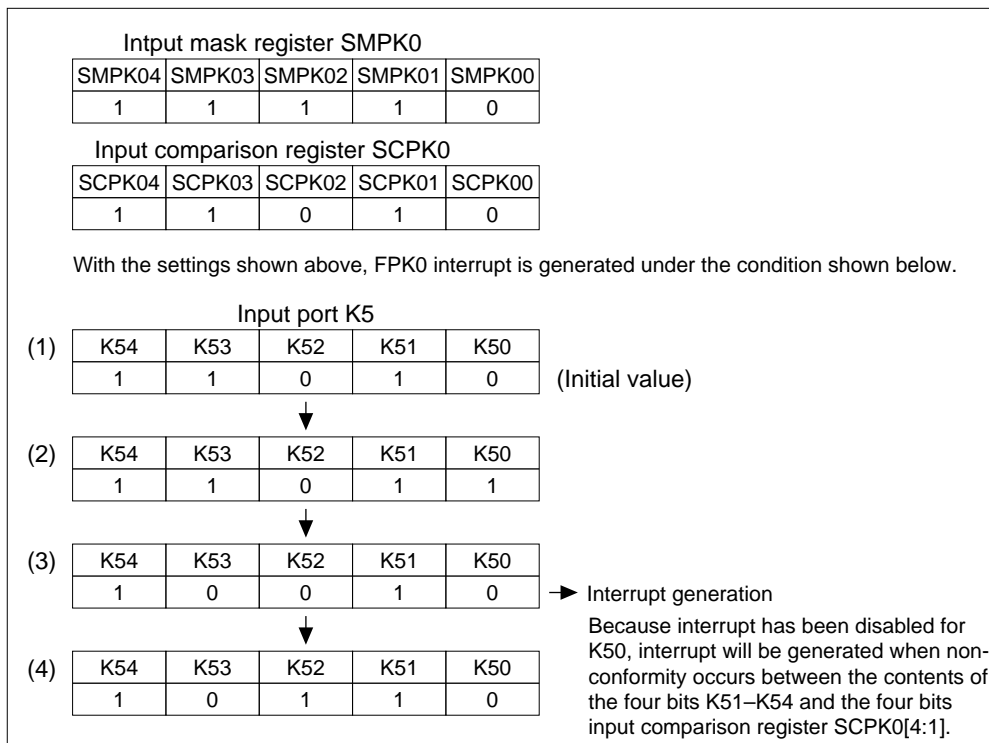


Figure III.11.5 FPK0 Interrupt Generation Example (when K5[4:0] is selected by SPPK[1:0])

Since K50 is masked from interrupt by SMPK00, no interrupt occurs at that point (2) above.

Next, because K53 becomes "0" at (3), an interrupt is generated due to the lack of a match between the data of the input pin K5[4:1] that is enabled for interrupt and that of the input comparison register SCPK0[4:1].

Since only a change in states in which the input data and the content of the input comparison register SCPK become unmatched after being matched constitutes an interrupt generation condition as described above, no interrupt is generated when a change in states from one unmatched state to another, as in (4), occurs. Consequently, if another interrupt is to be generated again following the occurrence of an interrupt, the state of the input pin must be temporarily restored to the same content as that of the input comparison register SCPK, or the input comparison register SCPK must be set again. Note that the input pins masked from interrupt by the SMPK register do not affect interrupt generation conditions.

An interrupt is generated for FPK1 in the same way as described above.

### Control Registers of the Interrupt Controller

Table III.11.8 shows the control registers of the interrupt controller that are provided for each input-interrupt system.

Table III.11.8 Control Registers of Interrupt Controller

System	Interrupt factor flag	Interrupt enable register	Interrupt priority register
FPT7	FP7(D5/0x40287)	EP7(D5/0x40277)	PP7L[2:0](D[6:4]/0x4026D)
FPT6	FP6(D4/0x40287)	EP6(D4/0x40277)	PP6L[2:0](D[2:0]/0x4026D)
FPT5	FP5(D3/0x40287)	EP5(D3/0x40277)	PP5L[2:0](D[6:4]/0x4026C)
FPT4	FP4(D2/0x40287)	EP4(D2/0x40277)	PP4L[2:0](D[2:0]/0x4026C)
FPT3	FP3(D3/0x40280)	EP3(D3/0x40270)	PP3L[2:0](D[6:4]/0x40261)
FPT2	FP2(D2/0x40280)	EP2(D2/0x40270)	PP2L[2:0](D[2:0]/0x40261)
FPT1	FP1(D1/0x40280)	EP1(D1/0x40270)	PP1L[2:0](D[6:4]/0x40260)
FPT0	FP0(D0/0x40280)	EP0(D0/0x40270)	PP0L[2:0](D[2:0]/0x40260)
FPK1	FK1(D5/0x40280)	EK1(D5/0x40270)	PK1L[2:0](D[6:4]/0x40262)
FPK0	FK0(D4/0x40280)	EK0(D4/0x40270)	PK0L[2:0](D[2:0]/0x40262)

When the interrupt generation condition described above is met, the corresponding interrupt factor flag is set to "1". If the interrupt enable register bit for that interrupt factor has been set to "1", an interrupt request is generated.

Interrupts due to an interrupt factor can be disabled by leaving the interrupt enable register bit for that factor set to "0". The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of the setting of the interrupt enable register.

The interrupt priority register sets the interrupt priority level (0 to 7) for each interrupt system. An interrupt request to the CPU is accepted only when no other interrupt request of a higher priority has been generated.

In addition, only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the input interrupt level set using the interrupt priority register will the input interrupt request actually be accepted by the CPU.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to "ITC (Interrupt Controller)".

## Intelligent DMA

The port input interrupt system can invoke an intelligent DMA (IDMA) through the use of its interrupt factor. This enables the port inputs to be used as a trigger to perform DMA transfer.

The following shows the IDMA channel numbers assigned to each interrupt factor:

IDMA Ch.	IDMA Ch.
FPT0 input interrupt: 1	FPT4 input interrupt: 28
FPT1 input interrupt: 2	FPT5 input interrupt: 29
FPT2 input interrupt: 3	FPT6 input interrupt: 30
FPT3 input interrupt: 4	FPT7 input interrupt: 31

For IDMA to be invoked, the IDMA request and IDMA enable bits shown in Table III.11.9 must be set to "1" in advance. Transfer conditions, etc. must also be set on the IDMA side in advance.

Table III.11.9 Control Bits for IDMA Transfer

System	IDMA request bit	IDMA enable bit
FPT7	RP7(D7/0x40293)	DEP7(D7/0x40297)
FPT6	RP6(D6/0x40293)	DEP6(D6/0x40297)
FPT5	RP5(D5/0x40293)	DEP5(D5/0x40297)
FPT4	RP4(D4/0x40293)	DEP4(D4/0x40297)
FPT3	RP3(D3/0x40290)	DEP3(D3/0x40294)
FPT2	RP2(D2/0x40290)	DEP2(D2/0x40294)
FPT1	RP1(D1/0x40290)	DEP1(D1/0x40294)
FPT0	RP0(D0/0x40290)	DEP0(D0/0x40294)

If the IDMA request and enable bits are set to "1", IDMA is invoked through generation of an interrupt factor. No interrupt request is generated at that point. An interrupt request is generated after the DMA transfer is completed. The registers can also be set so as not to generate an interrupt, with only DMA transfers performed. For details on IDMA transfers and interrupt control upon completion of IDMA transfer, refer to "IDMA (Intelligent DMA)".

## Trap vectors

The trap-vector address of each input default interrupt factor is set as follows:

FPT0 input interrupt:	0x0C00040
FPT1 input interrupt:	0x0C00044
FPT2 input interrupt:	0x0C00048
FPT3 input interrupt:	0x0C0004C
FPK0 input interrupt:	0x0C00050
FPK1 input interrupt:	0x0C00054
FPT4 input interrupt:	0x0C00110
FPT5 input interrupt:	0x0C00114
FPT6 input interrupt:	0x0C00118
FPT7 input interrupt:	0x0C0011C

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

### I/O Memory for Input Interrupts

Table III.11.10 shows the control bits for the port input and key input interrupts.

Table III.11.10 Control Bits for Input Interrupts

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Port input 0/1 interrupt priority register	0040260 (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PP1L2	Port input 1 interrupt level	0 to 7	X	R/W			
		D5	PP1L1							
		D4	PP1L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PP0L2	Port input 0 interrupt level	0 to 7	X	R/W			
		D1	PP0L1							
		D0	PP0L0							
Port input 2/3 interrupt priority register	0040261 (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PP3L2	Port input 3 interrupt level	0 to 7	X	R/W			
		D5	PP3L1							
		D4	PP3L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PP2L2	Port input 2 interrupt level	0 to 7	X	R/W			
		D1	PP2L1							
		D0	PP2L0							
Key input interrupt priority register	0040262 (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PK1L2	Key input 1 interrupt level	0 to 7	X	R/W			
		D5	PK1L1							
		D4	PK1L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PK0L2	Key input 0 interrupt level	0 to 7	X	R/W			
		D1	PK0L1							
		D0	PK0L0							
Port input 4/5 interrupt priority register	004026C (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PP5L2	Port input 5 interrupt level	0 to 7	X	R/W			
		D5	PP5L1							
		D4	PP5L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PP4L2	Port input 4 interrupt level	0 to 7	X	R/W			
		D1	PP4L1							
		D0	PP4L0							
Port input 6/7 interrupt priority register	004026D (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PP7L2	Port input 7 interrupt level	0 to 7	X	R/W			
		D5	PP7L1							
		D4	PP7L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PP6L2	Port input 6 interrupt level	0 to 7	X	R/W			
		D1	PP6L1							
		D0	PP6L0							
Key input, port input 0–3 interrupt enable register	0040270 (B)	D7–6	–	reserved	–	–	–	0 when being read.		
		D5	EK1	Key input 1	1	Enabled	0	Disabled	0	R/W
		D4	EK0	Key input 0					0	R/W
		D3	EP3	Port input 3					0	R/W
		D2	EP2	Port input 2					0	R/W
		D1	EP1	Port input 1					0	R/W
		D0	EP0	Port input 0					0	R/W
Port input 4–7, clock timer, A/D interrupt enable register	0040277 (B)	D7–6	–	reserved	–	–	–	–	0 when being read.	
		D5	EP7	Port input 7	1	Enabled	0	Disabled	0	R/W
		D4	EP6	Port input 6					0	R/W
		D3	EP5	Port input 5					0	R/W
		D2	EP4	Port input 4					0	R/W
		D1	ECTM	Clock timer					0	R/W
		D0	EADE	A/D converter					0	R/W
Key input, port input 0–3 interrupt factor flag register	0040280 (B)	D7–6	–	reserved	–	–	–	–	0 when being read.	
		D5	FK1	Key input 1	1	Factor is generated	0	No factor is generated	X	R/W
		D4	FK0	Key input 0					X	R/W
		D3	FP3	Port input 3					X	R/W
		D2	FP2	Port input 2					X	R/W
		D1	FP1	Port input 1					X	R/W
		D0	FP0	Port input 0					X	R/W

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks			
Port input 4–7, clock timer, A/D interrupt factor flag register	0040287 (B)	D7–6	–	reserved		–				–	–	0 when being read.		
		D5	FP7	Port input 7	1	Factor is generated	0	No factor is generated	X	R/W				
		D4	FP6	Port input 6	X				R/W					
		D3	FP5	Port input 5	X				R/W					
		D2	FP4	Port input 4	X				R/W					
		D1	FCTM	Clock timer	X				R/W					
		D0	FADE	A/D converter	X				R/W					
Port input 0–3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA request register	0040290 (B)	D7	R16TC0	16-bit timer 0 comparison A	1				IDMA request	0	Interrupt request	0	R/W	
		D6	R16TU0	16-bit timer 0 comparison B	0	R/W								
		D5	RHDM1	High-speed DMA Ch.1	0	R/W								
		D4	RHDM0	High-speed DMA Ch.0	0	R/W								
		D3	RP3	Port input 3	0	R/W								
		D2	RP2	Port input 2	0	R/W								
		D1	RP1	Port input 1	0	R/W								
		D0	RP0	Port input 0	0	R/W								
Serial I/F Ch.1, A/D, port input 4–7 IDMA request register	0040293 (B)	D7	RP7	Port input 7	1	IDMA request	0	Interrupt request	0	R/W				
		D6	RP6	Port input 6	0				R/W					
		D5	RP5	Port input 5	0				R/W					
		D4	RP4	Port input 4	0				R/W					
		D3	–	reserved					–	–	0 when being read.			
		D2	RADE	A/D converter	1				IDMA request	0	Interrupt request	0	R/W	
		D1	RSTX1	SIF Ch.1 transmit buffer empty	0							R/W		
D0	RSRX1	SIF Ch.1 receive buffer full	0	R/W										
Port input 0–3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA enable register	0040294 (B)	D7	DE16TC0	16-bit timer 0 comparison A	1	IDMA enabled	0	IDMA disabled	0	R/W				
		D6	DE16TU0	16-bit timer 0 comparison B	0				R/W					
		D5	DEHDM1	High-speed DMA Ch.1	0				R/W					
		D4	DEHDM0	High-speed DMA Ch.0	0				R/W					
		D3	DEP3	Port input 3	0				R/W					
		D2	DEP2	Port input 2	0				R/W					
		D1	DEP1	Port input 1	0				R/W					
		D0	DEP0	Port input 0	0				R/W					
Serial I/F Ch.1, A/D, port input 4–7 IDMA enable register	0040297 (B)	D7	DEP7	Port input 7	1	IDMA enabled	0	IDMA disabled	0	R/W				
		D6	DEP6	Port input 6	0				R/W					
		D5	DEP5	Port input 5	0				R/W					
		D4	DEP4	Port input 4	0				R/W					
		D3	–	reserved					–	–	0 when being read.			
		D2	DEADE	A/D converter	1				IDMA enabled	0	IDMA disabled	0	R/W	
		D1	DESTX1	SIF Ch.1 transmit buffer empty	0							R/W		
D0	DESRX1	SIF Ch.1 receive buffer full	0	R/W										
Port input interrupt select register 1	00402C6 (B)	D7	SPT31	FPT3 interrupt input port selection	11	10	01	00	0	R/W				
		D6	SPT30		P23	P03	K53	K63	0					
		D5	SPT21	FPT2 interrupt input port selection	11	10	01	00	0	R/W				
		D4	SPT20		P22	P02	K52	K62	0					
		D3	SPT11	FPT1 interrupt input port selection	11	10	01	00	0	R/W				
		D2	SPT10		P21	P01	K51	K61	0					
		D1	SPT01	FPT0 interrupt input port selection	11	10	01	00	0	R/W				
		D0	SPT00		P20	P00	K50	K60	0					
Port input interrupt select register 2	00402C7 (B)	D7	SPT71	FPT7 interrupt input port selection	11	10	01	00	0	R/W				
		D6	SPT70		P27	P07	P33	K67	0					
		D5	SPT61	FPT6 interrupt input port selection	11	10	01	00	0	R/W				
		D4	SPT60		P26	P06	P32	K66	0					
		D3	SPT51	FPT5 interrupt input port selection	11	10	01	00	0	R/W				
		D2	SPT50		P25	P05	P31	K65	0					
		D1	SPT41	FPT4 interrupt input port selection	11	10	01	00	0	R/W				
		D0	SPT40		P24	P04	K54	K64	0					
Port input interrupt input polarity select register	00402C8 (B)	D7	SPPT7	FPT7 input polarity selection	1	High level or Rising edge	0	Low level or Falling edge	1	R/W				
		D6	SPPT6	FPT6 input polarity selection	1				R/W					
		D5	SPPT5	FPT5 input polarity selection	1				R/W					
		D4	SPPT4	FPT4 input polarity selection	1				R/W					
		D3	SPPT3	FPT3 input polarity selection	1				R/W					
		D2	SPPT2	FPT2 input polarity selection	1				R/W					
		D1	SPPT1	FPT1 input polarity selection	1				R/W					
		D0	SPPT0	FPT0 input polarity selection	1				R/W					
Port input interrupt edge/level select register	00402C9 (B)	D7	SEPT7	FPT7 edge/level selection	1	Edge	0	Level	1	R/W				
		D6	SEPT6	FPT6 edge/level selection	1				R/W					
		D5	SEPT5	FPT5 edge/level selection	1				R/W					
		D4	SEPT4	FPT4 edge/level selection	1				R/W					
		D3	SEPT3	FPT3 edge/level selection	1				R/W					
		D2	SEPT2	FPT2 edge/level selection	1				R/W					
		D1	SEPT1	FPT1 edge/level selection	1				R/W					
		D0	SEPT0	FPT0 edge/level selection	1				R/W					

### III-11 PERIPHERAL BLOCK: INPUT/OUTPUT PORTS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Key input interrupt select register	00402CA (B)	D7-4	–	reserved		–	–	0 when being read.
		D3	<b>SPPK11</b>	FPK1 interrupt input port selection	11 10 01 00	0	R/W	
		D2	<b>SPPK10</b>	FPK0 interrupt input port selection	P2[7:4] P0[7:4] K6[7:4] K6[3:0]	0	R/W	
		D1	<b>SPPK01</b>	FPK0 interrupt input port selection	11 10 01 00	0	R/W	
D0	<b>SPPK00</b>	FPK0 interrupt input port selection	P2[4:0] P0[4:0] K6[4:0] K5[4:0]	0	R/W			
Key input interrupt (FPK0) input comparison register	00402CC (B)	D7-5	–	reserved		–	–	0 when being read.
		D4	<b>SCPK04</b>	FPK04 input comparison	1 High 0 Low	0 Low	0	R/W
		D3	<b>SCPK03</b>	FPK03 input comparison			0	R/W
		D2	<b>SCPK02</b>	FPK02 input comparison			0	R/W
		D1	<b>SCPK01</b>	FPK01 input comparison			0	R/W
		D0	<b>SCPK00</b>	FPK00 input comparison			0	R/W
Key input interrupt (FPK1) input comparison register	00402CD (B)	D7-4	–	reserved				–
		D3	<b>SCPK13</b>	FPK13 input comparison	1 High 0 Low	0 Low	0	R/W
		D2	<b>SCPK12</b>	FPK12 input comparison			0	R/W
		D1	<b>SCPK11</b>	FPK11 input comparison			0	R/W
		D0	<b>SCPK10</b>	FPK10 input comparison			0	R/W
Key input interrupt (FPK0) input mask register	00402CE (B)	D7-5	–	reserved				–
		D4	<b>SMPK04</b>	FPK04 input mask	1 Interrupt enabled 0 Interrupt disabled	0 Interrupt disabled	0	R/W
		D3	<b>SMPK03</b>	FPK03 input mask			0	R/W
		D2	<b>SMPK02</b>	FPK02 input mask			0	R/W
		D1	<b>SMPK01</b>	FPK01 input mask			0	R/W
		D0	<b>SMPK00</b>	FPK00 input mask			0	R/W
Key input interrupt (FPK1) input mask register	00402CF (B)	D7-4	–	reserved				–
		D3	<b>SMPK13</b>	FPK13 input mask	1 Interrupt enabled 0 Interrupt disabled	0 Interrupt disabled	0	R/W
		D2	<b>SMPK12</b>	FPK12 input mask			0	R/W
		D1	<b>SMPK11</b>	FPK11 input mask			0	R/W
		D0	<b>SMPK10</b>	FPK10 input mask			0	R/W

**SPT71–SPT70:** FPT7 interrupt input port selection (D[7:6]) / Port input interrupt select register 2 (0x402C7)

**SPT61–SPT60:** FPT6 interrupt input port selection (D[5:4]) / Port input interrupt select register 2 (0x402C7)

**SPT51–SPT50:** FPT5 interrupt input port selection (D[3:2]) / Port input interrupt select register 2 (0x402C7)

**SPT41–SPT40:** FPT4 interrupt input port selection (D[1:0]) / Port input interrupt select register 2 (0x402C7)

**SPT31–SPT30:** FPT3 interrupt input port selection (D[7:6]) / Port input interrupt select register 1 (0x402C6)

**SPT21–SPT20:** FPT2 interrupt input port selection (D[5:4]) / Port input interrupt select register 1 (0x402C6)

**SPT11–SPT10:** FPT1 interrupt input port selection (D[3:2]) / Port input interrupt select register 1 (0x402C6)

**SPT01–SPT00:** FPT0 interrupt input port selection (D[1:0]) / Port input interrupt select register 1 (0x402C6)

Select an input pin for port interrupt generation.

Table III.11.11 Selecting Pins for Port Input Interrupts

Interrupt system	SPT settings			
	11	10	01	00
FPT7	P27	P07	P33	K67
FPT6	P26	P06	P32	K66
FPT5	P25	P05	P31	K65
FPT4	P24	P04	K54	K64
FPT3	P23	P03	K53	K63
FPT2	P22	P02	K52	K62
FPT1	P21	P01	K51	K61
FPT0	P20	P00	K50	K60

At cold start, SPT is set to "00". At hot start, SPT retains its state from prior to the initial reset.

**SPPT7–SPPT0:** Input polarity selection (D[7:0]) / Port interrupt input polarity select register (0x402C8)

Selects input signal polarity for port interrupt generation.

Write "1": High level or Rising edge

Write "0": Low level or Falling edge

Read: Valid

SPPTx is the input polarity select bit corresponding to the FPTx interrupt. When SPPTx is set to "1", the FPTx interrupt will be generated by a high level input or at the rising edge. When SPPTx is set to "0", the interrupt will be generated by a low level input or at the falling edge. An edge or a level interrupt is selected by the SEPTx bit.

At cold start, SPPT is set to "0" (low level). At hot start, SPPT retains its state from prior to the initial reset.



**SEPT7–SEPT0:** Edge/level selection (D[7:0]) / Port interrupt edge/level select register (0x402C9)

Selects an edge trigger or a level trigger for port interrupt generation.

Write "1": Edge

Write "0": Level

Read: Valid

SEPTx is the edge/level select bit corresponding to the FPTx interrupt. When SEPTx is set to "1", the FPTx interrupt will be generated at the signal edge. Either falling edge or rising edge can be selected by the SPPTx bit. When SEPTx is set to "0", the interrupt will be generated by the level (high or low) specified with the SPPTx bit. At cold start, SEPT is set to "0" (level). At hot start, SEPT retains its state from prior to the initial reset.

**SPPK11–SPPK10:** FPK1 interrupt input port selection (D[3:2]) / Key input interrupt select register (0x402CA)**SPPK01–SPPK00:** FPK0 interrupt input port selection (D[1:0]) / Key input interrupt select register (0x402CA)

Select an input-pin group for key interrupt generation.

Table III.11.12 Selecting Pins for Key Input Interrupts

Interrupt system	SPPK settings			
	11	10	01	00
FPK1	P2[7:4]	P0[7:4]	K6[7:4]	K6[3:0]
FPK0	P2[4:0]	P0[4:0]	K6[4:0]	K5[4:0]

At cold start, SPPK is set to "00". At hot start, SPPK retains its state from prior to the initial reset.

**SCP13–SCP10:** FPK1 input comparison (D[3:0]) / FPK1 input comparison register (0x402CD)**SCP04–SCP00:** FPK0 input comparison (D[4:0]) / FPK0 input comparison register (0x402CC)

Sets the conditions for key-input interrupt generation (timing of interrupt generation).

Write "1": Generated at falling edge

Write "0": Generated at rising edge

Read: Valid

SCP0k[4:0] is compared with the input state of five bits of the FPK0 input ports, and SCP1k[3:0] is compared with the input state of four bits of the FPK1 input ports, and when a change in states from a matched to an unmatched state occurs in either, an interrupt is generated (except for the inputs disabled from interrupt by the SMPK register). At cold start, SCPK is set to "0" (rising edge). At hot start, SCPK retains its state from prior to the initial reset.

**SMPK13–SMPK10:** FPK1 input mask (D[3:0]) / FPK1 input mask register (0x402CF)**SMPK04–SMPK00:** FPK0 input mask (D[4:0]) / FPK0 input mask register (0x402CE)

Sets conditions for key-input interrupt generation (interrupt enabled/disabled).

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

SMPK is an input mask register for each key-input interrupt system. Interrupts for bits set to "1" are enabled, and interrupts for bits set to "0" are disabled. A change in the state of an input pin that is disabled from interrupt does not affect interrupt generation.

At cold start, SMPK is set to "0" (interrupt disabled). At hot start, SMPK retains its state from prior to the initial reset.

---

**PP0L2–PP0L0:** Port input 0 interrupt level (D[2:0]) / Port input 0/1 interrupt priority register (0x40260)

**PP1L2–PP1L0:** Port input 1 interrupt level (D[6:4]) / Port input 0/1 interrupt priority register (0x40260)

**PP2L2–PP2L0:** Port input 2 interrupt level (D[2:0]) / Port input 2/3 interrupt priority register (0x40261)

**PP3L2–PP3L0:** Port input 3 interrupt level (D[6:4]) / Port input 2/3 interrupt priority register (0x40261)

**PP4L2–PP4L0:** Port input 4 interrupt level (D[2:0]) / Port input 4/5 interrupt priority register (0x4026C)

**PP5L2–PP5L0:** Port input 5 interrupt level (D[6:4]) / Port input 4/5 interrupt priority register (0x4026C)

**PP6L2–PP6L0:** Port input 6 interrupt level (D[2:0]) / Port input 6/7 interrupt priority register (0x4026D)

**PP7L2–PP7L0:** Port input 7 interrupt level (D[6:4]) / Port input 6/7 interrupt priority register (0x4026D)

**PK0L2–PK0L0:** Key input 0 interrupt level (D[2:0]) / Key input interrupt priority register (0x40262)

**PK1L2–PK1L0:** Key input 1 interrupt level (D[6:4]) / Key input interrupt priority register (0x40262)

---

Sets the priority level of the input interrupt.

PPxL and PKxL are interrupt priority registers corresponding to each port-input interrupt and key-input interrupt, respectively.

The priority level can be set for each interrupt group in the range of 0 to 7.

At initial reset, these registers becomes indeterminate.

---

**EP3–EP0:** Port input 3–0 interrupt enable (D[3:0]) / Key input, port input 0–3 interrupt enable register (0x40270)

**EP7–EP4:** Port input 7–4 interrupt enable (D[5:2]) / Port input 4–7, clock timer, A/D interrupt enable register (0x40277)

**EK1, EK0:** Key input 1, 0 interrupt enable (D[5:4]) / Key input, port input 0–3 interrupt enable register (0x40270)

---

Enables or disables the generation of an interrupt to the CPU.

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

EP and EK are interrupt enable bits corresponding to the port-input interrupt and the key-input interrupt, respectively. Interrupts for input systems set to "1" are enabled, and interrupts for input systems set to "0" are disabled.

At initial reset, these bits are set to "0" (interrupt disabled).

---

**FP3–FP0:** Port input 3–0 interrupt factor flag (D[3:0]) / Key input, port input 0–3 interrupt factor flag register (0x40280)

**FP7–FP4:** Port input 7–4 interrupt factor flag (D[5:2]) / Port input 4–7, clock timer, A/D interrupt factor flag register (0x40287)

**FK1, FK0:** Key input 1, 0 interrupt factor flag (D[5:4]) / Key input, port input 0–3 interrupt factor flag register (0x40280)

---

Indicates the status of an input interrupt factor generated.

#### When read

Read "1": Interrupt factor has occurred

Read "0": No interrupt factor has occurred

#### When written using the reset-only method (default)

Write "1": Interrupt factor flag is reset

Write "0": Invalid

#### When written using the read/write method

Write "1": Interrupt flag is set

Write "0": Interrupt flag is reset

FP and FK are an interrupt factor flags corresponding to the port-input interrupt and the key-input interrupt, respectively. The flag is set to "1" when interrupt generation conditions are met.

At this time, if the following conditions are met, an interrupt to the CPU is generated:

1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The IE bit of the PSR is set to "1" (interrupts enabled).
4. The value set in the corresponding interrupt priority register is higher than the interrupt level (IL) of the CPU.

When using the interrupt factor of the port-input to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed. The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of how the interrupt enable and interrupt priority registers are set.

If the next interrupt is to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).

The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.

At initial reset, all the flags become indeterminate, so be sure to reset them in the software.

---

**RP3–RP0:** Port input 3–0 IDMA request (D[3:0]) / Port input 0–3, high-speed DMA, 16-bit timer 0 IDMA request register (0x40290)

**RP7–RP4:** Port input 7–4 IDMA request (D[7:4]) / Serial I/F Ch.1, A/D, Port input 4–7 IDMA request register (0x40293)

---

Specifies whether to invoke IDMA when an interrupt factor occurs.

#### When using the set-only method (default)

Write "1": IDMA request

Write "0": Not changed

Read: Valid

#### When using the read/write method

Write "1": IDMA request

Write "0": Interrupt request

Read: Valid

RP7 to RP0 are IDMA request bits corresponding to the port-input 7 to 0 interrupts, respectively. If the bit is set to "1", IDMA is invoked when an interrupt factor occurs, thereby performing a programmed data transfer. If the bit is set to "0", normal interrupt processing is performed, without invoking IDMA.

For details on IDMA, refer to "IDMA (Intelligent DMA)".

At initial reset, RP is set to "0" (interrupt request).

---

**DEP3–DEP0:** Port input 3–0 IDMA enable (D[3:0]) / Port input 0–3, high-speed DMA, 16-bit timer 0 IDMA enable register (0x40294)

**DEP7–DEP4:** Port input 7–4 IDMA enable (D[7:4]) / Serial I/F Ch.1, A/D, Port input 4–7 IDMA enable register (0x40297)

---

Enables IDMA transfer by means of an interrupt factor.

#### When using the set-only method (default)

Write "1": IDMA enabled

Write "0": Not changed

Read: Valid

#### When using the read/write method

Write "1": IDMA enabled

Write "0": IDMA disabled

Read: Valid

If DEP is set to "1", the IDMA request by the interrupt factor is enabled. If the register bit is set to "0", the IDMA request is disabled.

After an initial reset, DEP is set to "0" (IDMA disabled).

## ***Programming Notes***

---

- (1) After an initial reset, the interrupt factor flags become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset the flags in a program.
- (2) To prevent regeneration of interrupts due to the same factor following the occurrence of an interrupt, always be sure to reset the interrupt factor flag before resetting the PSR or executing the reti instruction.
- (3) The input/output ports operate only when the prescaler is operating.
- (4) When using an port input interrupt as the trigger to restart from the HALT2 or SLEEP mode, an interrupt will occur due to the input signal level even if edge interrupt is specified as an interrupt condition. The signal level to restart the CPU is as follows according to the signal edge selected:  
If a rising-edge interrupt is set, the CPU restarts when the input signal goes to a high level.  
If a falling-edge interrupt is set, the CPU restarts when the input signal goes to a low level.

When a falling edge interrupt is selected to restart after the slp instruction is executed, the operation is as follows.

If the interrupt port is already at a low level when the slp instruction is executed, the CPU enters SLEEP mode instantaneously and restarts immediately afterward.

If the interrupt port is at a high level when the slp instruction is executed, the SLEEP mode continues until the port goes low.

Therefore, design the system assuming that the CPU can restart normally due to the signal level at the interrupt port, not an edge interrupt, when restarting the CPU from HALT2 or SLEEP mode using a port input interrupt.

## III-12 EXTENDED PORTS

The S1C33301 has 18 extended I/O ports in addition to the ports in the C33 Peripheral Block. This section explains the functions and how to control the extended ports when they are used as general-purpose I/O ports.

### ***Structure of Extended I/O Port***

In the S1C33301, I/O ports that can be directed for input or output through the use of a program are extended into 47 bits from 29 bits of the standard I/O ports in the C33 Peripheral Block. The extended ports are P40 to P47, P50 to P55 and P60 to P63.

Figure III.12.1 shows the structure of an extended I/O port.

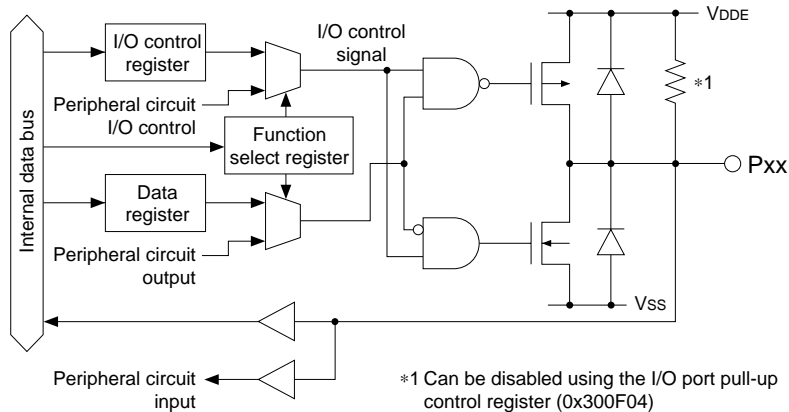


Figure III.12.1 Structure of Extended I/O Port

A pull-up resistor is provided for each I/O pin and it can be enabled/disabled by software control. At initial reset, the pull-up resistor is enabled and each pin is pulled up. Refer to Section III-2, "Chip ID / Pin Status Control Registers" for how to disconnect the pull-up resistor.

## Extended I/O Port Pins

The extended I/O ports concurrently serve as the input/output pins for the BCU, as shown in Table III.12.1.

Table III.12.1 I/O Pins

Pin name	I/O	Function	Function select bit
#HCAS(P40/A25)	I/O	DRAM column address strobe (high byte) / I/O port / Address bus (A25)	EFP40[1:0](D[1:0])/P40–P43 port function extension register(0x300048)
#LCAS(P41/A24)	I/O	DRAM column address strobe (low byte) / I/O port / Address bus (A24)	EFP41[1:0](D[3:2])/P40–P43 port function extension register(0x300048)
A23(P42)	I/O	Address bus (A23) / I/O port	EFP42[1:0](D[5:4])/P40–P43 port function extension register(0x300048)
A22(P43)	I/O	Address bus (A22) / I/O port	EFP43[1:0](D[7:6])/P40–P43 port function extension register(0x300048)
A21(P44)	I/O	Address bus (A21) / I/O port	EFP44[1:0](D[1:0])/P44–P47 port function extension register(0x300049)
A20(P45)	I/O	Address bus (A20) / I/O port	EFP45[1:0](D[3:2])/P44–P47 port function extension register(0x300049)
A19(P46)	I/O	Address bus (A19) / I/O port	EFP46[1:0](D[5:4])/P44–P47 port function extension register(0x300049)
A18(P47)	I/O	Address bus (A18) / I/O port	EFP47[1:0](D[7:6])/P44–P47 port function extension register(0x300049)
#CE4(#CE11/#CE11&12/P50)	I/O	Area 4 chip enable / Area 11 chip enable / Areas 11&12 chip enable / I/O port	EFP50[1:0](D[1:0])/P50–P53 port function extension register(0x30004A)
#CE5(#CE15/#CE15&16/P51)	I/O	Area 5 chip enable / Area 15 chip enable / Areas 15&16 chip enable / I/O port	EFP51[1:0](D[3:2])/P50–P53 port function extension register(0x30004A)
#CE6(#CE7&8/P52)	I/O	Area 6 chip enable / Areas 7&8 chip enable / I/O port	EFP52[1:0](D[5:4])/P50–P53 port function extension register(0x30004A)
#CE7(#RAS0/#CE13/#RAS2/P53)	I/O	Area 7 chip enable / Area 7 DRAM row strobe / Area 13 chip enable / Area 13 DRAM row strobe / I/O port	EFP53[1:0](D[7:6])/P50–P53 port function extension register(0x30004A)
#CE8(#RAS1/#CE14/#RAS3/P54)	I/O	Area 8 chip enable / Area 8 DRAM row strobe / Area 14 chip enable / Area 14 DRAM row strobe / I/O port	EFP54[1:0](D[1:0])/P54–P55 port function extension register(0x30004B)
#CE9(#CE17/#CE17&18/P55)	I/O	Area 9 chip enable / Area 17 chip enable / Areas 17&18 chip enable / I/O port	EFP55[1:0](D[3:2])/P54–P55 port function extension register(0x30004B)
BCLK(P60/FOSC1)	I/O	Bus clock output / I/O port / OSC1 clock output	EFP60[1:0](D[1:0])/P60 port function extension register(0x30004C)
P61	I/O	I/O port	–
P62	I/O	I/O port	–
P63	I/O	I/O port	–

At cold start, P40–P47, P50–P55 and P60 are configured as the bus signal output pins.

Pins that are not used for bus signal outputs can be used as general-purpose I/O port pins by setting the function extension bit EFP.

Refer to the "BCU (Bus Control Unit)" section for controlling the bus signals and "Low-speed (OSC1) Oscillation Circuit" section for the FOSC1 output.

At hot start, the pins retain their state from prior to the reset.

P61–P63 are exclusively used for I/O ports, note, however that these pins are not available in the PFBGA-121pin package.

## ***I/O Control Register and I/O Modes***

The extended I/O ports are directed for input or output modes by writing data to an I/O control register corresponding to each port bit.

P47–P40 I/O control: IOC4[7:0] (D[7:0]) / P4 I/O control register (0x300021)

P55–P50 I/O control: IOC5[5:0] (D[5:0]) / P5 I/O control register (0x300023)

P63–P60 I/O control: IOC6[3:0] (D[3:0]) / P6 I/O control register (0x300025)

To set an I/O port for input, write "0" to the I/O control bit. I/O ports set for input mode are placed in the high-impedance state, and thus function as input ports. In the input mode, the state of the input pin is read directly, so the data is "1" when the pin state is high (VDD level) or "0" when the pin state is low (VSS level).

Even in the input mode, data can be written to the data register without affecting the pin state.

To set an I/O port for output, write "1" to the I/O control bit. I/O port set for output function as output ports. When the port output data is "1", the port outputs a high level (VDD level); when the data is "0", the port outputs a low level (VSS level).

At cold start, the I/O control register is set to "0" (input mode).

At hot start, the pins retain their state from prior to the reset.

## ***Setting BCU to Access to the P4–P6 and EFP Registers***

The data and I/O control registers of the P40–P47, P50–P55 and P60–P63 ports, and the EFP registers for selecting extended port functions are mapped into Area 6 addresses 0x300020 to 0x30004C. Therefore, in order for the registers to be accessed, the BCU register for Area 6 must be set up in accordance with the procedure described below.

1. A6IO (D9) / Access control register (0x48132) = "1"  
This sets Area 6 so that the internal device can be accessed.
2. A6EC (D1) / Access control register (0x48132) = "0"  
This sets Area 6 so that it can be accessed in the little endian format.
3. A6WT[2:0] (D[A:8]) / Areas 6–4 setup register (0x4812A) = "001"  
This sets Area 6 so that it can be accessed with 1 wait state inserted.

## I/O Memory of Extended I/O Ports

Table III.12.2 shows the control bits of the extended I/O ports.

Table III.12.2 Control Bits of Extended I/O Ports

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks						
P4 I/O port data register	0300020 (B)	D7	P47D	P47 I/O port data	1	High	0	Low	0	R/W					
		D6	P46D	P46 I/O port data					0	R/W					
		D5	P45D	P45 I/O port data					0	R/W					
		D4	P44D	P44 I/O port data					0	R/W					
		D3	P43D	P43 I/O port data					0	R/W					
		D2	P42D	P42 I/O port data					0	R/W					
		D1	P41D	P41 I/O port data					0	R/W					
		D0	P40D	P40 I/O port data					0	R/W					
		P4 I/O control register	0300021 (B)	D7					IOC47	P47 I/O control		1	Output	0	Input
D6	IOC46			P46 I/O control	0	R/W									
D5	IOC45			P45 I/O control	0	R/W									
D4	IOC44			P44 I/O control	0	R/W									
D3	IOC43			P43 I/O control	0	R/W									
D2	IOC42			P42 I/O control	0	R/W									
D1	IOC41			P41 I/O control	0	R/W									
D0	IOC40			P40 I/O control	0	R/W									
P5 I/O port data register	0300022 (B)			D7-6	–	reserved	1	High	0	Low	–				
		D5	P55D	P55 I/O port data	0	R/W									
		D4	P54D	P54 I/O port data	0	R/W									
		D3	P53D	P53 I/O port data	0	R/W									
		D2	P52D	P52 I/O port data	0	R/W									
		D1	P51D	P51 I/O port data	0	R/W									
		D0	P50D	P50 I/O port data	0	R/W									
		P5 I/O control register	0300023 (B)	D7-6	–	reserved					1	Output	0	Input	–
D5	IOC55			P55 I/O control	0	R/W									
D4	IOC54			P54 I/O control	0	R/W									
D3	IOC53			P53 I/O control	0	R/W									
D2	IOC52			P52 I/O control	0	R/W									
D1	IOC51			P51 I/O control	0	R/W									
D0	IOC50			P50 I/O control	0	R/W									
P6 I/O port data register	0300024 (B)			D7-4	–	reserved	1	High	0	Low					–
		D3	P63D	P63 I/O port data	0	R/W									
		D2	P62D	P62 I/O port data	0	R/W									
		D1	P61D	P61 I/O port data	0	R/W									
		D0	P60D	P60 I/O port data	0	R/W									
P6 I/O control register	0300025 (B)	D7-4	–	reserved	1	Output	0	Input	–	–	0 when being read.				
		D3	IOC63	P63 I/O control					0	R/W					
		D2	IOC62	P62 I/O control					0	R/W					
		D1	IOC61	P61 I/O control					0	R/W					
		D0	IOC60	P60 I/O control					0	R/W					
P00–P03 port function extension register	0300040 (B)	D7 D6	EFP031 EFP030	P03 port extended function	1	*	EFP03[1:0]		Function	0	R/W				
							reserved		0						
							#FSRDY0 P03/#SRDY0		0						
		D5 D4	EFP021 EFP020	P02 port extended function	1	*	EFP02[1:0]		Function	0	R/W				
							reserved		0						
							#FSCLK0 P02/#SCLK0		0						
		D3 D2	EFP011 EFP010	P01 port extended function	1	*	EFP01[1:0]		Function	0	R/W				
							reserved		0						
							FSOUT0 P01/SOUT0		0						
		D1 D0	EFP001 EFP000	P00 port extended function	1	*	EFP00[1:0]		Function	0	R/W				
							reserved		0						
							FSIN0 P00/SIN0		0						
P34–P35 port function extension register	0300047 (B)	D7-4	–	reserved	1	*	–		–	–	0 when being read.				
							D3 D2	EFP351 EFP350	P35 port extended function	EFP35[1:0]		Function	0	R/W	
										reserved		0			
		#SMRE P35, etc.		0											
		D1 D0	EFP341 EFP340	P34 port extended function			1	*	EFP34[1:0]			Function	0	R/W	
									reserved			0			
									#SMWE P34, etc.			0			
					0										



Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
P40–P43 port function extension register	0300048 (B)	D7 D6	EFP431	P43 port extended function	EFP43[1:0]		Function	0	R/W		
			EFP430		1	*	reserved	0			
					0	1	P43				
					0	0	A22				
		D5 D4	EFP421	P42 port extended function	EFP42[1:0]		Function	0	R/W		
			EFP420		1	*	reserved	0			
					0	1	P42				
					0	0	A23				
		D3 D2	EFP411	P41 port extended function	EFP41[1:0]		Function	0	R/W		
			EFP410		1	1	reserved	0			
					1	0	A24				
					0	1	P41				
					0	0	#LCAS				
		D1 D0	EFP401	P40 port extended function	EFP40[1:0]		Function	0	R/W		
			EFP400		1	1	reserved	0			
					1	0	A25				
	0		1		P40						
			0	0	#HCAS						
P44–P47 port function extension register	0300049 (B)	D7 D6	EFP471	P47 port extended function	EFP47[1:0]		Function	0	R/W		
			EFP470		1	*	reserved	0			
					0	1	P47				
					0	0	A18				
		D5 D4	EFP461	P46 port extended function	EFP46[1:0]		Function	0	R/W		
			EFP460		1	*	reserved	0			
					0	1	P46				
					0	0	A19				
		D3 D2	EFP451	P45 port extended function	EFP45[1:0]		Function	0	R/W		
			EFP450		1	*	reserved	0			
					0	1	P45				
					0	0	A20				
		D1 D0	EFP441	P44 port extended function	EFP44[1:0]		Function	0	R/W		
			EFP440		1	*	reserved	0			
					0	1	P44				
					0	0	A21				
P50–P53 port function extension register	030004A (B)	D7 D6	EFP531	P53 port extended function	EFP53[1:0]		Function	0	R/W		
			EFP530		1	*	reserved	0			
					0	1	P53				
					0	0	#CE7, etc.				
		D5 D4	EFP521	P52 port extended function	EFP52[1:0]		Function	0	R/W		
			EFP520		1	*	reserved	0			
					0	1	P52				
					0	0	#CE6, etc.				
		D3 D2	EFP511	P51 port extended function	EFP51[1:0]		Function	0	R/W		
			EFP510		1	*	reserved	0			
					0	1	P51				
					0	0	#CE5, etc.				
		D1 D0	EFP501	P50 port extended function	EFP50[1:0]		Function	0	R/W		
			EFP500		1	*	reserved	0			
					0	1	P50				
					0	0	#CE4, etc.				
P54–P55 port function extension register	030004B (B)	D7–4	–	reserved	–		–	–	0 when being read.		
			D3 D2	EFP551 EFP550	P55 port extended function	EFP55[1:0]		Function		0	R/W
						1	*	reserved	0		
						0	1	P55			
						0	0	#CE9, etc.			
		D1 D0	EFP541 EFP540	P54 port extended function	EFP54[1:0]		Function	0	R/W		
							1	*		reserved	0
							0	1		P54	
					0	0	#CE8, etc.				
P60 port function extension register	030004C (B)	D7–2	–	reserved	–		–	–	0 when being read.		
			D1 D0	EFP601 EFP600	P60 port extended function	EFP60[1:0]		Function		0	R/W
						1	1	reserved	0		
						1	0	FOSC1			
						0	1	P60			
			0	0	BCLK						

---

**P47D–P40D:** P4[7:0] I/O port data (D[7:0]) / P4 I/O port data register (0x300020)

**P55D–P50D:** P5[5:0] I/O port data (D[5:0]) / P5 I/O port data register (0x300022)

**P63D–P60D:** P6[3:0] I/O port data (D[3:0]) / P6 I/O port data register (0x300024)

---

This register reads data from I/O-port pins or sets output data.

**When writing data**

Write "1": High level

Write "0": Low level

When an I/O port is set for output, the data written to it is directly output to the I/O port pin. If the data written to the port is "1", the port pin is set high (VDD and VDDE level); if the data is "0", the port pin is set low (VSS level). Even in the input mode, data can be written to the port data register.

**When reading data**

Read "1": High level

Read "0": Low level

The voltage level on the port pin is read out regardless of whether an I/O port is set for input or output mode. If the pin voltage is high (VDD and VDDE level), "1" is read out as input data; if the pin voltage is low (VSS level), "0" is read out as input data.

At cold start, all data bits are set to "0". At hot start, they retain their state from prior to the initial reset.

---

**IOC47–IOC40:** P4[7:0] port I/O control (D[7:0]) / P4 port I/O control register (0x300021)

**IOC55–IOC50:** P5[5:0] port I/O control (D[5:0]) / P5 port I/O control register (0x300023)

**IOC63–IOC60:** P6[3:0] port I/O control (D[3:0]) / P6 port I/O control register (0x300025)

---

Directs an I/O port for input or output and indicates the I/O control signal value of the port.

**When writing data**

Write "1": Output mode

Write "0": Input mode

This I/O control register corresponds bit-for-bit to each I/O port. When an IOC bit is set to "1", the corresponding I/O port is directed for output; if it is set to "0", the I/O port is directed for input.

At cold start, all IOC bits are set to "0" (input). At hot start, IOC retains its state from prior to the initial reset.

**When reading data**

Read "1": I/O control signal (output)

Read "0": I/O control signal (input)

The I/O control signal value for the port pin is read from this register. When I/O port function is selected using the EFP register, the value written to the IOC register is read out as is. When a bus signal is selected, the read value may not indicate the value written to the IOC register.

**EFP001–EFP000:** P00 port extended function (D[1:0]) / P00–P03 port function extension register (0x300040)  
**EFP011–EFP010:** P01 port extended function (D[3:2]) / P00–P03 port function extension register (0x300040)  
**EFP021–EFP020:** P02 port extended function (D[5:4]) / P00–P03 port function extension register (0x300040)  
**EFP031–EFP030:** P03 port extended function (D[7:6]) / P00–P03 port function extension register (0x300040)  
**EFP341–EFP340:** P34 port extended function (D[1:0]) / P34–P35 port function extension register (0x300047)  
**EFP351–EFP350:** P35 port extended function (D[3:2]) / P34–P35 port function extension register (0x300047)  
**EFP401–EFP400:** P40 port extended function (D[1:0]) / P40–P43 port function extension register (0x300048)  
**EFP411–EFP410:** P41 port extended function (D[3:2]) / P40–P43 port function extension register (0x300048)  
**EFP421–EFP420:** P42 port extended function (D[5:4]) / P40–P43 port function extension register (0x300048)  
**EFP431–EFP430:** P43 port extended function (D[7:6]) / P40–P43 port function extension register (0x300048)  
**EFP441–EFP440:** P44 port extended function (D[1:0]) / P44–P47 port function extension register (0x300049)  
**EFP451–EFP450:** P45 port extended function (D[3:2]) / P44–P47 port function extension register (0x300049)  
**EFP461–EFP460:** P46 port extended function (D[5:4]) / P44–P47 port function extension register (0x300049)  
**EFP471–EFP470:** P47 port extended function (D[7:6]) / P44–P47 port function extension register (0x300049)  
**EFP501–EFP500:** P50 port extended function (D[1:0]) / P50–P53 port function extension register (0x30004A)  
**EFP511–EFP510:** P51 port extended function (D[3:2]) / P50–P53 port function extension register (0x30004A)  
**EFP521–EFP520:** P52 port extended function (D[5:4]) / P50–P53 port function extension register (0x30004A)  
**EFP531–EFP530:** P53 port extended function (D[7:6]) / P50–P53 port function extension register (0x30004A)  
**EFP541–EFP540:** P54 port extended function (D[1:0]) / P54–P55 port function extension register (0x30004B)  
**EFP551–EFP550:** P55 port extended function (D[3:2]) / P54–P55 port function extension register (0x30004B)  
**EFP601–EFP600:** P60 port extended function (D[1:0]) / P60 port function extension register (0x30004C)

Selects the port extended function. See Table III.12.3 for the port functions that can be selected.

At cold start, EFPxx[1:0] is set to "00" (bus signal output). At hot start, EFPxx[1:0] retains its state from prior to the initial reset.

Table III.12.3 List of Port Functions

Port	Function extension bit	EFPxx[1:0] = "00"	EFPxx[1:0] = "01"	EFPxx[1:0] = "10"	EFPxx[1:0] = "11"
P00	EFP00[1:0]	<b>P00</b> SIN0	FSIN0	–	–
P01	EFP01[1:0]	<b>P01</b> SOUT0	FSOUT0	–	–
P02	EFP02[1:0]	<b>P02</b> #SCLK0	#FSCLK0	–	–
P03	EFP03[1:0]	<b>P03</b> #SRDY0	#FSRDY0	–	–
P34	EFP34[1:0]	<b>P34</b> #BUSREQ #CE6	#SMWE	–	–
P35	EFP35[1:0]	<b>P35</b> #BUSACK	#SMRE	–	–
P40	EFP40[1:0]	<b>#HCAS</b>	P40	A25	–
P41	EFP41[1:0]	<b>#LCAS</b>	P41	A24	–
P42	EFP42[1:0]	<b>A23</b>	P42	–	–
P43	EFP43[1:0]	<b>A22</b>	P43	–	–
P44	EFP44[1:0]	<b>A21</b>	P44	–	–
P45	EFP45[1:0]	<b>A20</b>	P45	–	–
P46	EFP46[1:0]	<b>A19</b>	P46	–	–
P47	EFP47[1:0]	<b>A18</b>	P47	–	–
P50	EFP50[1:0]	<b>#CE4</b> #CE11 #CE11&12	P50	–	–
P51	EFP51[1:0]	<b>#CE5</b> #CE15 #CE15&16	P51	–	–
P52	EFP52[1:0]	<b>#CE6</b> #CE7&8	P52	–	–
P53	EFP53[1:0]	<b>#CE7</b> #RAS0 #CE13 #RAS2	P53	–	–
P54	EFP54[1:0]	<b>#CE8</b> #RAS1 #CE14 #RAS3	P54	–	–
P55	EFP55[1:0]	<b>#CE9</b> #CE17 #CE17&18	P55	–	–
P60	EFP60[1:0]	<b>BCLK</b>	P60	FOSC1	–

Default: EFPxx[1:0] = "00"

THIS PAGE IS BLANK.

# III-13 SMARTMEDIA INTERFACE

## Configuration of SmartMedia Interface

The S1C33301 contains a SmartMedia Interface as an extended peripheral circuit. This circuit generates the following signals for connecting SmartMedia cards to the bus:

#SMWE (write signal for SmartMedia)

#SMRE (read signal for SmartMedia)

A 3.3 V SmartMedia Card can be directly connected to the S1C33301 bus using these signals as the -WE and -RE signals for SmartMedia.

Figure III.13.1 shows the structure of the SmartMedia interface.

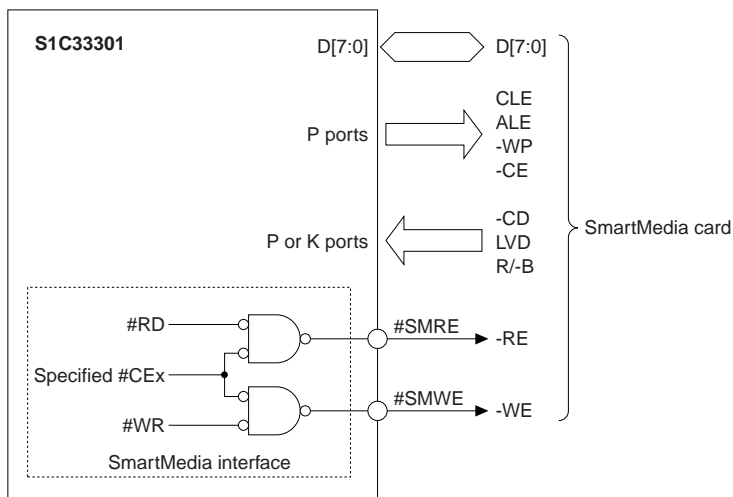


Figure III.13.1 Structure of SmartMedia Interface

## Output Pins of SmartMedia Interface

Table III.13.1 shows the pins that are used to output the #SMWE and #SMRE signals.

Table III.13.1 Output Pins of SmartMedia Interface

Pin name	I/O	Function	Function select bit
P34(#BUSREQ/#CE6/#SMWE)	I/O	I/O port / Bus release request / Area 6 chip enable / SmartMedia write signal	EFP34[1:0](D[1:0])/P34–P35 port function extension register(0x300047)
P35(#BUSACK/#SMRE)	I/O	I/O port / Bus request acknowledge / SmartMedia read signal	EFP35[1:0](D[3:2])/P34–P35 port function extension register(0x300047)

The #SMWE and #SMRE pins are shared with the P34 and P35 I/O ports, respectively. At cold start, these pins are both set for the I/O port. When using the SmartMedia interface, set EFP34[1:0] (D[1:0]) and EFP35[1:0] (D[3:2]) / P34–P35 port function extension register (0x300047) to "01".

These pins are also shared with the #BUSREQ and #BUSACK I/O signals, therefore the bus request function cannot be used when they are used as the SmartMedia control signal outputs.

At hot start, the register retains its status from prior to the reset.

## Selecting a #CE Area

Use SMCES[1:0] (D[1:0]) / SmartMedia I/F #CE area select register (0x300100) to select which #CE signal is assigned for accessing SmartMedia. As shown in Table III.13.2, there are four choices, however, the area to be allocated is changed according to the CEFUNC[1:0] (D[A:9]/0x48130) setting.

Table III.13.2 Selecting a #CE Area

SMCES1	SMCES0	#CE area		
		CEFUNC = "00"	CEFUNC = "01"	CEFUNC = "1x"
1	1	#CE9	#CE17	#CE17+#CE18
1	0	#CE8	#CE14	#CE14
0	1	#CE7	#CE13	#CE13
0	0	#CE4	#CE11	#CE11+#CE12

(Default: SMCES = CEFUNC = "00")

The selected #CE signal is ORed with the #WR signal to generate the #SMWE signal and is ORed with the #RD signal to generate the #SMRE signal.

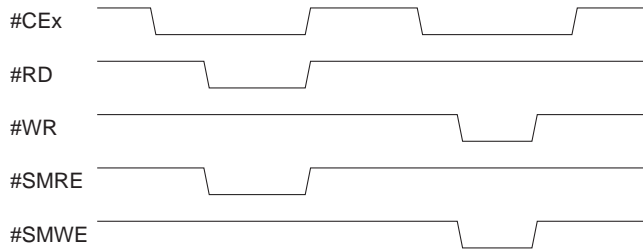


Figure III.13.2 Timing Chart of the #SMWE and #SMRE Signals

## Setting BCU to Access to the SmartMedia I/F Registers

The P34–P35 function extension register used for pin configuration and the SmartMedia I/F #CE area select register are mapped into Area 6 addresses 0x300047 and 0x300100. Therefore, in order for the registers to be accessed, the BCU register for Area 6 must be set up in accordance with the procedure described below.

1. A6IO (D9) / Access control register (0x48132) = "1"  
This sets Area 6 so that the internal device will be accessed.
2. A6EC (D1) / Access control register (0x48132) = "0"  
This sets Area 6 so that it will be accessed in the little endian format.
3. A6WT[2:0] (D[A:8]) / Areas 6–4 setup register (0x4812A) = "001"  
This sets Area 6 so that it can be accessed with 1 wait state inserted.

## I/O Memory of SmartMedia Interface

Table III.13.3 shows the SmartMedia interface control bits.

Table III.13.3 SmartMedia Interface Control Bits

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P34–P35 port function extension register	0300047 (B)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	EFP351	P35 port extended function	EFP35[1:0] Function	0	R/W	
		D2	EFP350		1 * reserved	0		
					0 1 #SMRE			
					0 0 P35, etc.			
		D1	EFP341	P34 port extended function	EFP34[1:0] Function	0	R/W	
D0	EFP340	1 * reserved	0					
		0 1 #SMWE						
			0 0 P34, etc.					
SmartMedia I/F #CE area select register	0300100 (B)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	SMCES1	SmartMedia I/F #CE area selection (according of CEFUNC)	SMCES[1:0] #CE area	0	R/W	
		D0	SMCES0		1 1 #CE9/#CE17(+18)	0		
					1 0 #CE8/#CE14			
					0 1 #CE7/#CE13			
					0 0 #CE4/#CE11(+12)			

**EFP341–EFP340:** P34 port extended function (D[1:0]) / P34–P35 port function extension register (0x300047)

**EFP351–EFP350:** P35 port extended function (D[3:2]) / P34–P35 port function extension register (0x300047)

Switches the P34 and P35 port functions.

Table III.13.4 P34 and P35 port extended functions

EFP3x1	EFP3x0	P34	P35
1	*	reserved	reserved
0	1	#SMWE	#SMRE
0	0	P34/#BUSREQ/#CE6	P35/#BUSACK

Set EFP34[1:0] and EFP35[1:0] to "01" when using P34 and P35 for the SmartMedia interface. The P34 and P35 pins are configured as the #SMWE and #SMRE pins, respectively.

At initial reset, EFP is set to "00".

**SMCES1–SMCES0:** #CE area selection (D[1:0]) / SmartMedia I/F #CE area select register (0x300100)

Selects a #CE area to be assigned to SmartMedia.

Table III.13.5 Selecting a #CE area

SMCES1	SMCES0	#CE area		
		CEFUNC = "00"	CEFUNC = "01"	CEFUNC = "1x"
1	1	#CE9	#CE17	#CE17+#CE18
1	0	#CE8	#CE14	#CE14
0	1	#CE7	#CE13	#CE13
0	0	#CE4	#CE11	#CE11+#CE12

(Default: SMCES = CEFUNC = "00")

The selected #CE signal is ORed with the #WR signal to generate the #SMWE signal and is ORed with the #RD signal to generate the #SMRE signal.

At initial reset, SMCES is set to "00" (#CE4).

THIS PAGE IS BLANK.



S1C33301 Technical Manual  
**IV ANALOG BLOCK**



# IV-1 INTRODUCTION

The analog block consists of an A/D converter with 8 input channels.

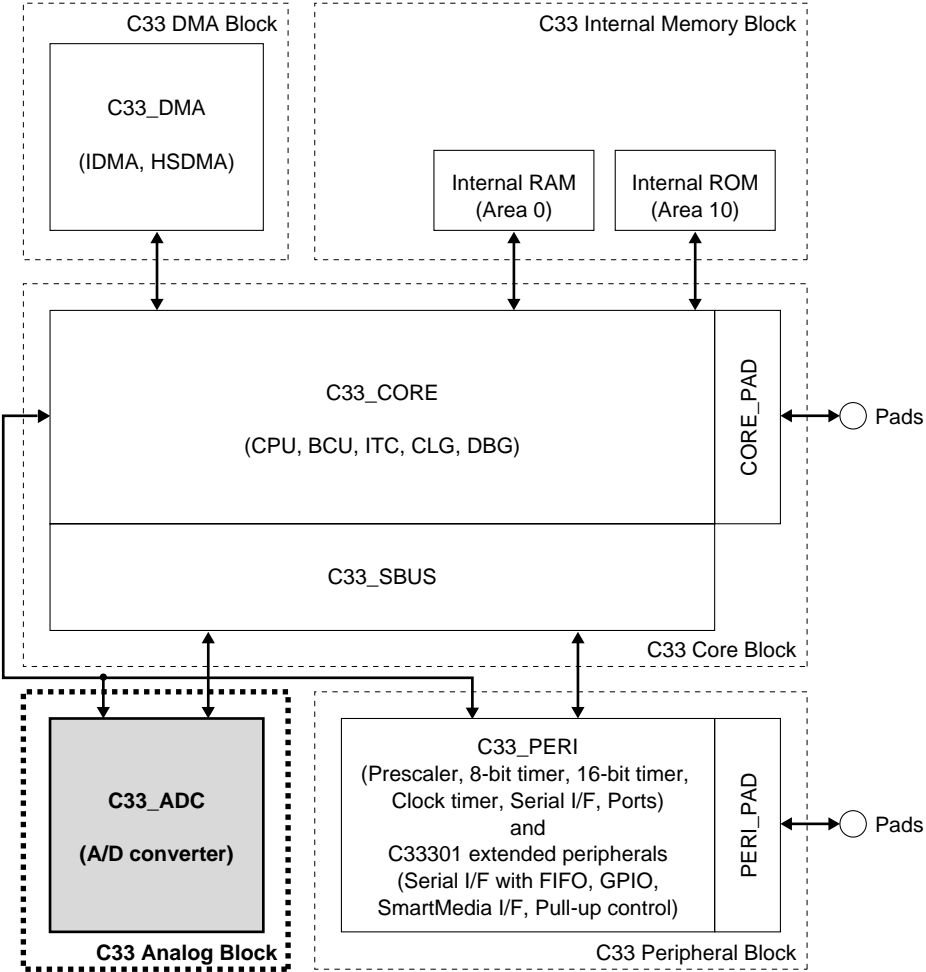


Figure IV.1.1 Analog Block

**Note:** Internal ROM is not provided in the S1C33301.

THIS PAGE IS BLANK.

## IV-2 A/D CONVERTER

### Features and Structure of A/D Converter

The C33 STD contains an A/D converter with the following features:

- Conversion method: Successive comparison
- Resolution: 10 bits
- Input channels: Maximum of 8
- A/D converter input clock: Maximum of 2 MHz, minimum of 16 kHz
- Conversion time: Minimum of 10  $\mu$ s (when a 2-MHz input clock is selected)  
Maximum of 1250  $\mu$ s (when a 16-kHz input clock is selected)
- Conversion range: Between  $V_{SS}$  and  $AV_{DDE}$
- Two conversion modes can be selected:
  - Normal mode: Conversion is completed in one operation.
  - Continuous mode: Conversion is continuous and terminated through software control.
- Continuous conversion of multiple channels can be performed in each mode.
- Four types of A/D-conversion start triggers can be selected:
  - Triggered by the external pin (#ADTRG)
  - Triggered by the compare match B of the 16-bit programmable timer 0
  - Triggered by the underflow of the 8-bit programmable timer 0
  - Triggered by the software
- A/D conversion results can be read out from the 10-bit data register or the conversion result buffer\* for each channel.
- An interrupt is generated upon completion of A/D conversion or when the conversion result is out of the specified range (upper and lower-limit values can be specified)\*.

\* These functions can be used in the advanced mode. The A/D converter of the C33 STD has two operating modes, 33209 compatible mode of which functions are compatible with the C33 analog block for the existing models and an advanced mode allowing use of the extended functions.

Figure IV.2.1 shows the structure of the A/D converter.

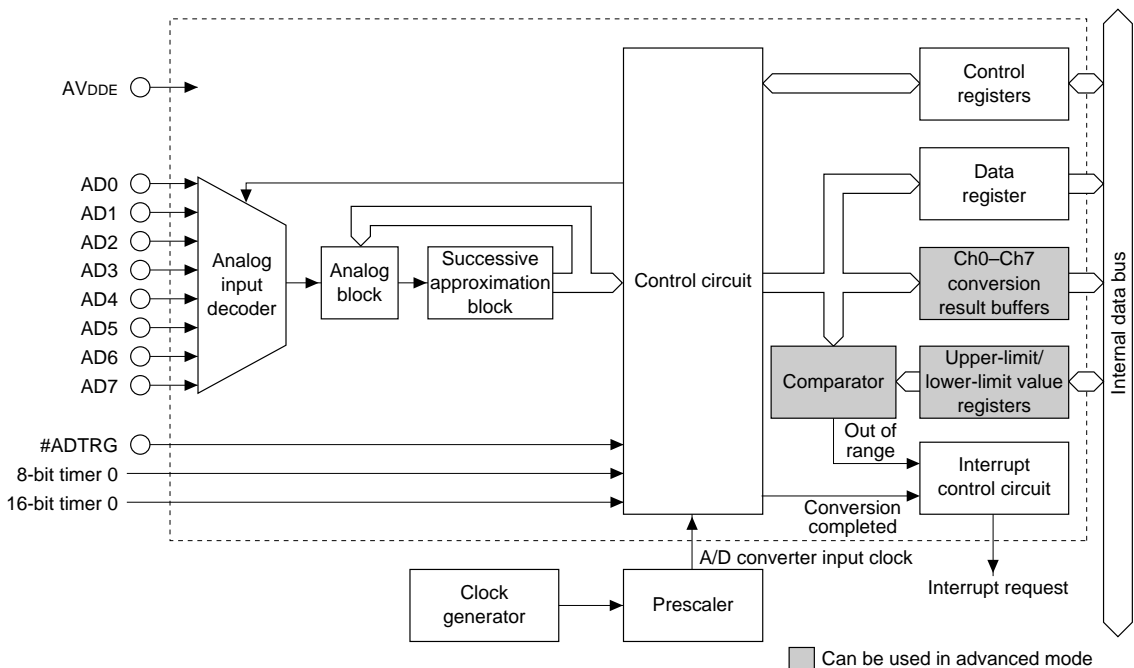


Figure IV.2.1 Structure of A/D Converter

## I/O Pins of A/D Converter

Table IV.2.1 shows the pins used by the A/D converter.

Table IV.2.1 I/O Pins of A/D Converter

Pin name	I/O	Function	Function select bit
K52( <b>#ADTRG</b> )	I	Input port / AD trigger	CFK52(D2)/K5 function select register(0x402C0)
K60( <b>AD0</b> )	I	Input port / AD converter input 0	CFK60(D0)/K6 function select register(0x402C3)
K61( <b>AD1</b> )	I	Input port / AD converter input 1	CFK61(D1)/K6 function select register(0x402C3)
K62( <b>AD2</b> )	I	Input port / AD converter input 2	CFK62(D2)/K6 function select register(0x402C3)
K63( <b>AD3</b> )	I	Input port / AD converter input 3	CFK63(D3)/K6 function select register(0x402C3)
K64( <b>AD4</b> )	I	Input port / AD converter input 4	CFK64(D4)/K6 function select register(0x402C3)
K65( <b>AD5</b> )	I	Input port / AD converter input 5	CFK65(D5)/K6 function select register(0x402C3)
K66( <b>AD6</b> )	I	Input port / AD converter input 6	CFK66(D6)/K6 function select register(0x402C3)
K67( <b>AD7</b> )	I	Input port / AD converter input 7	CFK67(D7)/K6 function select register(0x402C3)
<b>AVDDE</b>	–	Analog reference voltage (+)	–

### AVDDE (analog power-supply pin)

AVDDE is the power-supply pin for the analog circuit.

**Note:** When the A/D converter is set to enabled state, a current flows between AVDDE and VSS, and power is consumed, even when A/D operations are not performed. Therefore, when the A/D converter is not used, it must be set to the disabled state (default "0" setting of ADE (D2) in the A/D enable register (0x40244)).

### AD[7:0] (analog-signal input pins)

The analog input pins AD7 (Ch.7) through AD0 (Ch.0) are shared with input port pins K67 through K60.

Therefore, when these pins are used for analog input, they must be set for use with the A/D converter in the software. This setting can be made individually for each pin. At cold start, all these pins are set for input ports.

The analog input voltage AVIN can be input in the range of  $V_{SS} \leq AV_{IN} \leq AV_{DDE}$ .

### #ADTRG (external-trigger input pin)

This pin is used to input a trigger signal to start A/D conversion from an external source. Since this pin is shared with input port K52, it must be set for use with the A/D converter in the software before an external trigger can be applied to the pin. At cold start, this pin is set for an input port.

### Method for setting A/D-converter input pins

At cold start, the #ADTRG and AD[7:0] pins all are set for input ports Kxx (function select bit CFKxx = "0").

When using these pins for the A/D converter, write "1" to the function select bit CFKxx.

At hot start, these pins retain their state prior to the reset.

## Setting A/D Converter

When the A/D converter is used, the following settings must be made before an A/D conversion can be performed:

1. Setting analog input pins
2. Setting the operating mode (33209 compatible mode/advanced mode)
3. Setting the input clock
4. Selecting the analog-conversion start and end channels
5. Setting the A/D conversion mode
6. Selecting a trigger
7. Setting the sampling time
8. Setting the upper-limit and lower-limit values (advanced mode)
9. Setting the interrupt mode (advanced mode)
10. Setting interrupt/IDMA/HSDMA

The following describes how to set each item. For details on how to set the analog input pins, refer to the preceding section. For details on how to set interrupt/DMA, refer to "A/D Converter Interrupt and DMA".

**Note:** Before making these settings, make sure the A/D converter is disabled (ADE (D2) / A/D enable register (0x40244) = "0"). Changing the settings while the A/D converter is enabled could cause a malfunction.

### Setting the operating mode (33209 compatible mode / advanced mode)

The A/D converter of the C33 STD has two operating modes, 33209 compatible mode of which functions are compatible with the C33 analog block for the existing models and an advanced mode allowing use of the extended functions. Table IV.2.2 shows differences between the 33209 compatible mode and the advanced mode.

Table IV.2.2 Differences Between 33209 Compatible Mode and Advanced Mode

Function	33209 compatible mode	Advanced mode
Reading conversion results	The conversion results are read from the A/D conversion result register common to all channels. When converting for multiple channels, the A/D conversion result register must be read before conversion for the next channel has completed.	The conversion results can be read from the conversion result buffer provided for each channel. Thus the conversion result for the current channel will not be lost even when the conversion for the next channel is completed during a multiple channel conversion.
Conversion-complete flag, overwrite error flag	One bit is assigned for the flag and is commonly used in all channels.	Different flags are provided for each channel.
Comparison with upper/lower-limit values	Not supported.	An upper-limit value and a lower-limit value can be set and conversion results of the specified channel can be checked whether they are within the specified range or not.
Interrupts	Conversion-complete interrupt only can be generated. The interrupts cannot be masked in channel units.	Conversion-complete interrupts and out-of-range interrupts can be generated. Conversion complete interrupts for the specified channels can be masked.

To configure the A/D converter in the advanced mode, set ADCADV (D0) / A/D converter mode select register (0x4025F) to "1". The control bits for the extended functions can be accessed after this setting. At initial reset, ADCADV is set to "0" and the A/D converter enters the 33209 compatible mode.

The following descriptions unless otherwise specified are common contents for both modes.

The extended functions in the advanced mode are explained assuming that ADCADV has been set to "1".

## Setting the input clock

As explained in "Prescaler", the A/D conversion clock can be selected from among the eight types shown in Table IV.2.3 below. Use PSAD[2:0] (D[2:0]) / A/D clock control register (0x4014F) for this selection.

Table IV.2.3 Input Clock Selection

PSAD2	PSAD1	PSAD0	Division ratio
1	1	1	fPSCIN/256
1	1	0	fPSCIN/128
1	0	1	fPSCIN/64
1	0	0	fPSCIN/32
0	1	1	fPSCIN/16
0	1	0	fPSCIN/8
0	0	1	fPSCIN/4
0	0	0	fPSCIN/2

fPSCIN: Prescaler input clock frequency

The selected clock is output from the prescaler to the A/D converter by writing "1" to PSONAD (D3) / A/D clock control register (0x4014F).

- Notes:**
- The A/D converter operates only when the prescaler is operating.
  - The recommended input clock frequency is a maximum of 2 MHz and a minimum of 16 kHz.
  - Do not start an A/D conversion when the clock output from the prescaler to the A/D converter is turned off, and do not turn off the prescaler's clock output when an A/D conversion is underway. This could cause the A/D converter to operate erratically.

## Selecting analog-conversion start and end channels

Select the channel in which the A/D conversion is to be performed from among the pins (channels) that have been set for analog input. To enable A/D conversions in multiple channels to be performed successively through one convert operation, specify the conversion start and conversion end channels.

Conversion start channel: CS[2:0] (D[2:0]) / A/D channel register (0x40243)

Conversion end channel: CE[2:0] (D[5:3]) / A/D channel register (0x40243)

Table IV.2.4 Relationship between CS/CE and Input Channel

CS2/CE2	CS1/CE1	CS0/CE0	Channel selected
1	1	1	AD7
1	1	0	AD6
1	0	1	AD5
1	0	0	AD4
0	1	1	AD3
0	1	0	AD2
0	0	1	AD1
0	0	0	AD0

Example: Operation of one A/D conversion

CS[2:0] = "0", CE[2:0] = "0": Converted only in AD0

CS[2:0] = "0", CE[2:0] = "3": Converted in the following order: AD0→AD1→AD2→AD3

CS[2:0] = "5", CE[2:0] = "1": Converted in the following order: AD5→AD6→AD7→AD0→AD1

**Note:** Only conversion-channel input pins that have been set for use with the A/D converter can be set using the CS and CE bits.

## Setting the A/D conversion mode

The A/D converter can operate in one of the following two modes. This operation mode is selected using MS (D5) / A/D trigger register (0x40242).

### 1. Normal mode (MS = "0")

All inputs in the range of channels set using the CS and CE bits are A/D converted once and then stopped.

### 2. Continuous mode (MS = "1")

A/D conversions in the range of channels set using the CS and CE bits are executed successively until stopped by the software.

At initial reset, the normal mode is selected.



## Selecting a trigger

Use TS[1:0] (D[4:3]) / A/D trigger register (0x40242) to select a trigger to start A/D conversion from among the four types shown in Table IV.2.5.

Table IV.2.5 Trigger Selection

TS1	TS0	Trigger
1	1	External trigger (K52/#ADTRG)
1	0	8-bit programmable timer 0
0	1	16-bit programmable timer 0
0	0	Software

### 1. External trigger

The signal input to the #ADTRG pin is used as a trigger. When this trigger is used, the K52 pin must be set for #ADTRG in advance by writing "1" to CFK52 (D2) / K5 function select register (0x402C0). A/D conversion is started when a low level of the #ADTRG signal is detected.

### 2. Programmable timer

The underflow signal of 8-bit programmable timer 0 or the comparison match B signal of the 16-bit programmable timer 0 is used as a trigger. Since the cycle can be programmed using each timer, this trigger is effective when cyclic A/D conversions are required.

For details on how to set a timer, refer to the explanation of each programmable timer in this manual.

### 3. Software trigger

Writing "1" to ADST (D1) / A/D enable register (0x40244) in the software serves as a trigger to start A/D conversion.

## Setting the sampling time

The A/D converter contains ST[1:0] (D[1:0]) / A/D sampling register (0x40245) that allows the analog-signal input sampling time to be set in four steps (3, 5, 7, or 9 times the input clock period).

However, this register should be used as set by default (ST = "11"; x9 clock periods).

## Setting the upper-limit and lower-limit values (advanced mode)

The advanced mode allows a range check of the conversion results by setting the upper-limit and lower-limit values. Setup the A/D converter according to the procedure shown below to use this function.

### 1. Selecting the channel

Select the channel to compare the A/D conversion results and the upper-limit and lower-limit value using the ADCMP[2:0] (D[6:4]) / A/D sampling register (0x40245).

Table IV.2.6 Selecting the Channel for Checking Conversion Results

ADCMP2	ADCMP1	ADCMP0	Channel selected
1	1	1	AD7
1	1	0	AD6
1	0	1	AD5
1	0	0	AD4
0	1	1	AD3
0	1	0	AD2
0	0	1	AD1
0	0	0	AD0

### 2. Setting upper-limit and lower-limit values

Set the upper-limit value to the A/D upper-limit value register ADUPR[9:0] (ADUPR[9:8] = D[1:0]/0x40259, ADUPR[7:0] = D[7:0]/0x40258) and the lower-limit value to the A/D lower-limit value register ADLWR[9:0] (ADLWR[9:8] = D[1:0]/0x4025B, ADLWR[7:0] = D[7:0]/0x4025A).

When the conversion result exceeds the upper-limit value set or is lower than the lower-limit value, it is determined as out of range. If the conversion result is the same value as the upper-limit or lower-limit value, it is determined as within the range.

### 3. Enabling comparison with the upper-limit and lower-limit values

Set the ADCMPE (D7) / A/D sampling register (0x40245) to "1" to enable the range check function.

## Setting the interrupt mode (advanced mode)

The interrupt functions are extended in the advanced mode, so the following configuration is necessary.

### 1. Enabling/disabling the conversion-complete interrupt

The conversion-complete interrupt can be enabled/disabled using the CNVINTEN (D4) / A/D enable register (0x40244). Set CNVINTEN to "1" when using the conversion-complete interrupt, or to "0" when it is not used. At initial reset, CNVINTEN is set to "1", so the conversion-complete interrupt function is enabled.

### 2. Enabling/disabling the out-of-range interrupt

The out-of-range interrupt can be enabled/disabled using the CMPINTEN (D5) / A/D enable register (0x40244). Set CMPINTEN to "1" when using the out-of range interrupt, or to "0" when it is not used. At initial reset, CMPINTEN is set to "0", so the conversion-complete interrupt function is disabled.

### 3. Setting the interrupt signal mode

The A/D converter uses one signal line for interrupt requests to the interrupt controller (ITC). In the initial setting, only the conversion-complete interrupt signal is output to the ITC. When using the out-of-range interrupt, the out-of-range interrupt signal is ORed with the above signal to send to the ITC. This signal mode can be selected by setting the INTMODE (D6) / A/D enable register (0x40244) to "1".

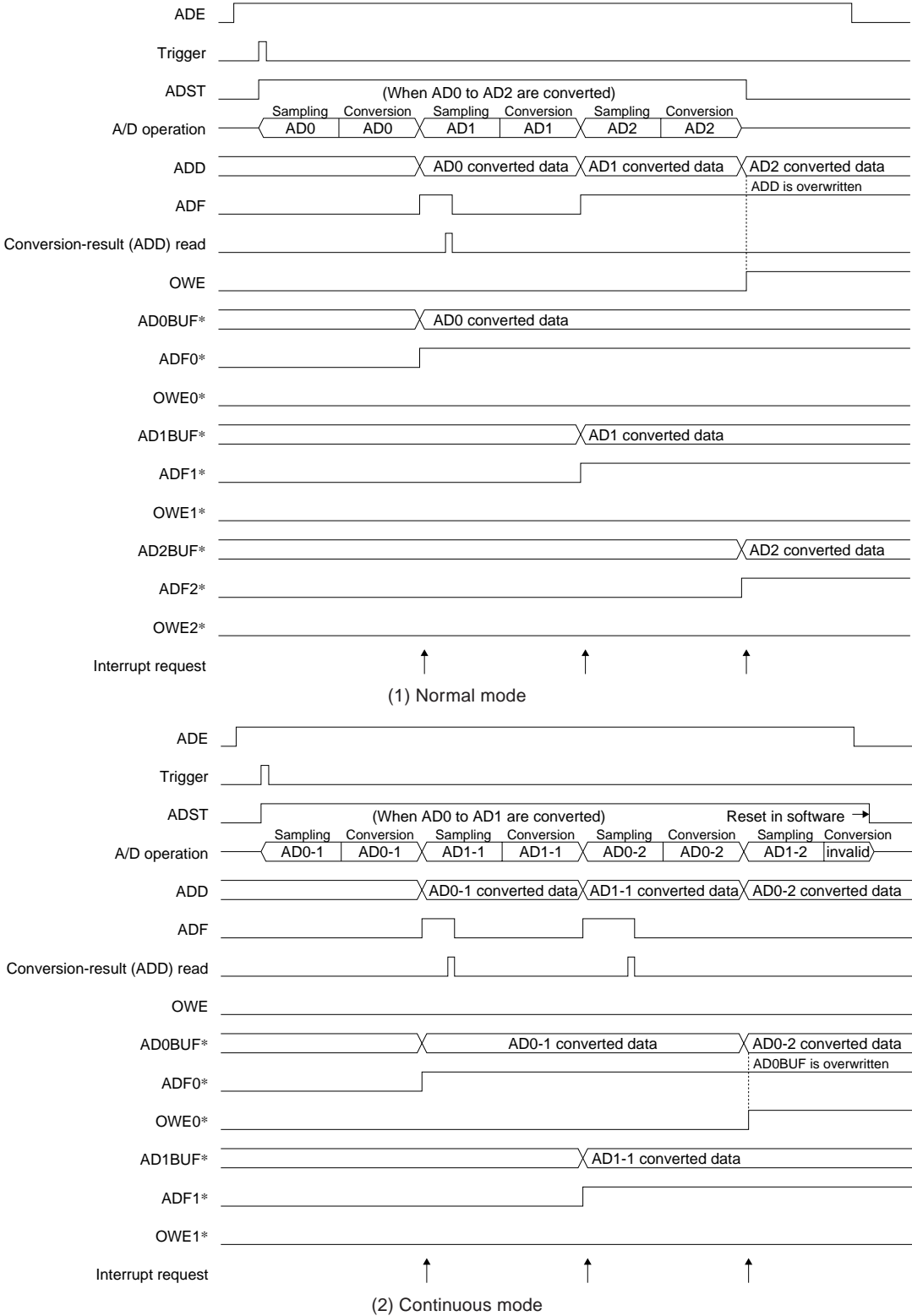
### 4. Masking conversion-complete interrupt for the specified channels

The A/D conversion-complete interrupt mask register (0x4025C) is used to mask the conversion-complete interrupts of the specified channels. When the INTMASKx bit (Dx) for channel x in the register is set to "0", channel x does not generate conversion-complete interrupts. For instance, by masking the conversion-complete interrupt of the channel used for range checking, it is possible to generate out-of range interrupts only.

At initial reset, the INTMASK bits are all set to "1" to enable conversion-complete interrupts.

# Control and Operation of A/D Conversion

Figure IV.2.2 shows the operation of the A/D converter.



\* Extended functions that can be used when ADCADV = "1"

Figure IV.2.2 Operation of A/D Converter

## Starting up the A/D converter circuit

After the settings specified in the preceding section have been made, write "1" to ADE (D2) / A/D enable register (0x40244) to enable the A/D converter. The A/D converter is thereby readied to accept a trigger to start A/D conversion. To set the A/D converter again, or if it is not be used, set ADE to "0".

## Starting A/D conversion

When a trigger is input while ADE = "1", A/D conversion is started. If a software trigger has been selected, A/D conversion is started by writing "1" to ADST (D1) / A/D enable register (0x40244).

Only the trigger selected using TS[1:0] (D[4:3]) / A/D trigger register (0x40242) are valid; no other trigger is accepted.

When a trigger is input, the A/D converter samples and A/D-converts the analog input signal, beginning with the conversion start channel selected by CS[2:0].

The ADST used for the software trigger is set to "1" during A/D conversion, even when it is started by some other trigger, so it can be used as an A/D-conversion status bit.

The channel in which conversion is underway can be identified by reading CH[2:0] (D[2:0]) / A/D trigger register (0x40242).

## Reading out A/D conversion results

### 33209 compatible mode

Upon completion of the A/D conversion in the start channel, the A/D converter stores the conversion result, in 10-bit data registers ADD[9:0] (ADD[9:8] = D[1:0]/0x40241, ADD[7:0] = D[7:0]/0x40240), and sets the conversion-complete flag ADF (D3) / A/D enable register (0x40244) and interrupt factor flag FADE (D0) / Port input 4–7, clock timer and A/D interrupt factor flag register (0x40287). If multiple channels are specified using CS[2:0] and CE[2:0], A/D conversions in the subsequent channels are performed in succession.

The results of A/D conversion are stored in the ADD[9:0] register each time conversion in one channel is completed. Since an interrupt can be generated simultaneously, this interrupt is normally used to read out the converted data. In addition, be sure to reset the interrupt factor flag (by writing "0") to prepare the A/D converter for the next operation.

Since the interrupt factor of the A/D converter can also be used to invoke DMA, the conversion results can automatically be transferred to a specified memory location.

If multiple A/D conversion channels are specified, the conversion results in one channel must be read out prior to completion of conversion in the next channel. If the A/D conversion currently under way is completed before the previous conversion results are read out, the ADD[9:0] register is overwritten with the new conversion results.

If ADD[9:0] is updated when the conversion-complete flag ADF = "1" (before the converted data is read out), the overwrite-error flag OWE (D0) / A/D enable register (0x40244) is set to "1". The conversion-complete flag ADF is reset to "0" when the converted data is read out. If ADD[9:0] is updated when ADF = "0", OWE remains at "0", indicating that the operation has been completed normally. When reading out data, also read the OWE flag also to make sure the data is valid. Once OWE is set, it remains set until it is reset to "0" in the software. Note also that if OWE is set, ADF also is set. In this case, read out the converted data and reset ADF.

### Advanced mode

Upon completion of the A/D conversion in the start channel (Ch.x), the A/D converter stores the conversion result to the 10-bit Ch.x conversion result buffer ADxBUF[9:0] and sets the Ch.x conversion-complete flag ADFx (Dx) / A/D conversion-complete flag register (0x40246) and the interrupt factor flag FADE (D0/0x40287). If multiple channels are specified using CS[2:0] and CE[2:0], A/D conversions in the subsequent channels are performed in succession.

The results of A/D conversion are stored in the A/D conversion result buffer for each channel each time conversion in one channel is completed. Since an interrupt can be generated simultaneously, this interrupt is normally used to read out the converted data. In addition, be sure to reset the interrupt factor flag (by writing "0") to prepare the A/D converter for the next operation.

Since the interrupt factor of the A/D converter can also be used to invoke DMA, the conversion results can automatically be transferred to a specified memory location.

In the advanced mode, each channel has a conversion result buffer, so it is not necessary to read the conversion results prior to completion of conversion in the next channel. However, if the next A/D conversion in the same channel is completed before the previous conversion results are read out, the conversion result buffer is overwritten with the new conversion results. If ADxBUF[9:0] is updated when the conversion-complete flag ADFx = "1" (before the converted data is read out), the overwrite-error flag OWEx (Dx) / A/D overwrite error flag register (0x40247) is set to "1". The conversion-complete flag ADFx is reset to "0" when the converted data is read out. If ADxBUF[9:0] is updated when ADFx = "0", OWEx remains at "0", indicating that the operation has been completed normally. When reading out data, read the OWEx flag also to make sure the data is valid. Once OWEx is set, it remains set until it is reset to "0" by writing "0" in the software. Note also that if OWEx is set, ADFx is also set. In this case, read out the converted data and reset ADFx.

ADD, ADF and OWE used in the 33209 compatible mode are also effective in the advanced mode as well. The functions and actions of the register/bits are the same as those of the 33209 compatible mode. The OWE flag is set during conversion in multiple-channels, but it is not necessary to reset it.

### Range check (comparison with upper-limit/lower-limit values in advanced mode)

When the range check function is enabled (ADCMPE = "1") and an A/D conversion in the channel specified using ADCMP[2:0] has completed, the conversion results are compared with the contents of the A/D upper-limit value register ADUPR[9:0] and A/D lower-limit value register ADLWR[9:0].

If the conversion results exceed the upper-limit value, the upper-limit comparison status bit ADUPRST (D3) / A/D sampling register (0x40245) is set to "1". If the results are less than the lower-limit value, the lower-limit comparison status bit ADLWRST (D2) / A/D sampling register (0x40245) is set to "1". When the out-of range interrupt is enabled, an interrupt occurs if one of the status bits has been set. This interrupt request sets the same interrupt factor flag FADE as the conversion-complete interrupt to "1". Therefore, the above status bits must be read out to check whether the out-of-range interrupt is generated or not if the conversion-complete interrupt is also enabled.

When the conversion results are the same as the upper-limit or lower-limit values, it is assumed within the range and an interrupt is not generated.

### Terminating A/D conversion

- **For normal mode (MS = "1")**

In the normal mode, A/D conversion is performed successively from the conversion start channel specified using CS[2:0] to the conversion end channel specified using CE[2:0], and is completed after these conversions are executed in one operation. ADST is reset to "0" upon completion of the conversion.

- **For continuous mode (MS = "0")**

In the continuous mode, A/D conversion from the conversion-start to the conversion-end channels is executed repeatedly, without being stopped in the hardware. To terminate conversion, therefore, ADST must be reset to "0" in the software. However, the A/D conversion being executed will be completed normally or forcibly stopped depending on the timing of writing "0" to ADST. When the A/D conversion has completed normally, ADF is set to "1" and the conversion results can be obtained. If it is forcibly stopped, ADF maintains its previous status, therefore, conversion results cannot be obtained.

- **Forced termination**

A/D conversion is immediately terminated by writing "0" to ADST. The results of the conversion then underway cannot be obtained.

## ***A/D Converter Interrupt and DMA***

---

Upon completion of A/D conversion in each channel, the A/D converter generates an interrupt and invokes the IDMA if necessary. In the advanced mode, the A/D converter can generate an interrupt when the conversion results are out of the range specified with the upper-limit and lower-limit registers.

### **Control registers of the interrupt controller**

The following shows the interrupt control registers available for the A/D converter:

Interrupt factor flag: FADE (D0) / Port input 4–7, clock timer, A/D interrupt factor flag register (0x40287)

Interrupt enable: EADE (D0) / Port input 4–7, clock timer, A/D interrupt enable register (0x40277)

Interrupt level: PAD[2:0] (D[6:4]) / Serial I/F Ch.1, A/D interrupt priority register (0x4026A)

The A/D converter sets the interrupt factor flag to "1" when A/D conversion in one channel is completed, and the conversion results are stored in the ADD and ADxBUF (advanced mode) registers.

If the out-of-range interrupt is enabled in the advanced mode, the interrupt factor flag is set to "1" when the conversion results in the specified channel are out of range. Both the conversion-complete and out-of-range interrupts use the same interrupt factor flag.

At this time, if the interrupt enable register bit has been set to "1", an interrupt request is generated. Interrupts can be disabled by leaving the interrupt enable register bit set to "0". The interrupt factor flag is set to "1" upon completion of A/D conversion in each channel, regardless of the setting of the interrupt enable register (even when it is set to "0"). The interrupt priority register sets the priority level (0 to 7) of an interrupt. An interrupt request to the CPU is accepted no other interrupt request of a higher priority has been generated. In addition, it is only when the PSR's IE bit = "1" (interrupts enabled) and the set value of the IL is smaller than the A/D-converter interrupt level set by the interrupt priority register, that the A/D converter's interrupt request is actually accepted by the CPU. For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to "ITC (Interrupt Controller)".

### **Intelligent DMA**

The A/D converter can invoke the intelligent DMA (IDMA) through the use of its interrupt factor. This allows the conversion results to be transferred to a specified memory location with no need to execute an interrupt processing routine. The IDMA channel number assigned to the A/D converter is 0x1B.

Before IDMA can be invoked, the IDMA request and IDMA enable bits must be set to "1". Transfer conditions on the IDMA side must also be set in advance.

IDMA request: RADE (D2) / Serial I/F Ch.1, A/D, Port input 4–7 IDMA request register (0x40293)

IDMA enable: DEADE (D2) / Serial I/F Ch.1, A/D, Port input 4–7 IDMA enable register (0x40297)

If an interrupt factor occurs when the IDMA request and IDMA enable bits are set to "1", IDMA is invoked. No interrupt request is generated at that point. An interrupt request is generated upon completion of the DMA transfer. Otherwise, the bit can be set so as not to generate an interrupt, with only a DMA transfer performed.

For details on DMA transfers and how to control interrupts upon completion of a DMA transfer, refer to "IDMA (Intelligent DMA)".

## High-speed DMA

The A/D interrupt factor can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit:

Table IV.2.7 HSDMA Trigger Set-up Bits

HSDMA channel	Trigger set-up bits
0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0/1 trigger set-up register (0x40298)
1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298)
2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299)
3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299)

For HSDMA to be invoked, the trigger set-up bits should be set to "1100" in advance. Transfer conditions, etc. must also be set on the HSDMA side. If the A/D interrupt factor is selected as the HSDMA trigger, the HSDMA channel is invoked through generation of the interrupt factor.

For details on HSDMA transfer, refer to "HSDMA (High-Speed DMA)".

## Trap vector

The A/D converter's interrupt trap-vector default address is set to 0x0C00100.

The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

## I/O Memory of A/D Converter

Table IV.2.8 shows the control bits of the A/D converter.

For details on the I/O memory of the prescaler used to set clocks, refer to "Prescaler". For details on the I/O memory of the programmable timers used for a trigger, refer to "8-Bit Programmable Timers" or "16-Bit Programmable Timers".

Table IV.2.8 Control Bits of A/D Converter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
A/D conversion result (low-order) register	0040240 (B)	D7	ADD7	A/D converted data (low-order 8 bits) ADD0 = LSB	0x0 to 0x3FF (low-order 8 bits)		0	R			
		D6	ADD6								
		D5	ADD5								
		D4	ADD4								
		D3	ADD3								
		D2	ADD2								
		D1	ADD1								
		D0	ADD0								
		A/D conversion result (high-order) register	0040241 (B)				D7-2			–	–
D1	ADD9			A/D converted data (high-order 2 bits) ADD9 = MSB	0x0 to 0x3FF (high-order 2 bits)		0	R			
D0	ADD8										
A/D trigger register	0040242 (B)	D7-6	–	–	–		–	–	0 when being read.		
		D5	MS	A/D conversion mode selection	1	Continuous	0	Normal	0	R/W	
		D4	TS1	A/D conversion trigger selection	TS[1:0]		Trigger		0	R/W	
		D3	TS0		1	1	#ADTRG pin				
					1	0	8-bit timer 0				
				0	1	16-bit timer 0					
				0	0	Software					
		D2	CH2	A/D conversion channel status	CH[2:0]		Channel		0	R	
		D1	CH1		1	1	1	AD7	0		
		D0	CH0		1	1	0	AD6	0		
					1	0	1	AD5			
					1	0	0	AD4			
		0	1		1	AD3					
		0	1		0	AD2					
		0	0	1	AD1						
		0	0	0	AD0						
A/D channel register	0040243 (B)	D7-6	–	–	–		–	–	0 when being read.		
		D5	CE2	A/D converter end channel selection	CE[2:0]		End channel		0	R/W	
		D4	CE1		1	1	1	AD7	0		
		D3	CE0		1	1	0	AD6	0		
					1	0	1	AD5			
					1	0	0	AD4			
					0	1	1	AD3			
					0	1	0	AD2			
				0	0	1	AD1				
				0	0	0	AD0				
		D2	CS2	A/D converter start channel selection	CS[2:0]		Start channel		0	R/W	
		D1	CS1		1	1	1	AD7	0		
		D0	CS0		1	1	0	AD6	0		
1	0				1	AD5					
		1	0		0	AD4					
		0	1		1	AD3					
		0	1		0	AD2					
		0	0	1	AD1						
		0	0	0	AD0						
A/D enable register	0040244 (B)	D7	–	reserved	–		–	–	0 when being read.		
		D6	INTMODE	Interrupt signal mode	1	Complete only	0	OR	0	R/W	Can be used when
		D5	CMPINTEN	Out-of-range int. enable	1	Enabled	0	Disabled	0	R/W	ADCADV = "1".
		D4	CNVINTEN	Conversion-complete int. enable	1	Enabled	0	Disabled	1	R/W	
		D3	ADF	Conversion-complete flag	1	Completed	0	Run/Standby	0	R	Reset when ADD is read.
		D2	ADE	A/D enable	1	Enabled	0	Disabled	0	R/W	
		D1	ADST	A/D conversion control/status	1	Start/Run	0	Stop	0	R/W	
		D0	OWE	Overwrite error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.



Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
A/D sampling register	0040245 (B)	D7	ADCMPE	Upper/lower-limit comparison enable	1	Enabled	0 Disabled	0	R/W	Can be used when ADCADV = "1".
		D6	ADCMP2	Upper/lower-limit comparison channel selection	0 to 7			0	R/W	
		D5	ADCMP1		0					
		D4	ADCMP0		0					
		D3	ADUPRST	Upper-limit comparison status	1	Out of range	0 Within range	0	R	
		D2	ADLWRST	Lower-limit comparison status	1	Out of range	0 Within range	0	R	
		D1	ST1	Input signal sampling time setup	ST[1:0]		Sampling time	1	R/W	Use with 9 clocks.
		D0	ST0		1	1	9 clocks	1		
		1	0		7 clocks					
		0	1		5 clocks					
			0	0	3 clocks					
A/D conversion complete flag register	0040246 (B)	D7	ADF7	Ch.7 conversion-complete flag	1	Completed	0 Run/Standby	0	R	Can be used when ADCADV = "1". Reset when ADBUFx is read.
		D6	ADF6	Ch.6 conversion-complete flag				0	R	
		D5	ADF5	Ch.5 conversion-complete flag				0	R	
		D4	ADF4	Ch.4 conversion-complete flag				0	R	
		D3	ADF3	Ch.3 conversion-complete flag				0	R	
		D2	ADF2	Ch.2 conversion-complete flag				0	R	
		D1	ADF1	Ch.1 conversion-complete flag				0	R	
		D0	ADF0	Ch.0 conversion-complete flag				0	R	
A/D overwrite error flag	0040247 (B)	D7	OWE7	Ch.7 overwrite error flag	1	Error	0 Normal	0	R/W	Can be used when ADCADV = "1". Reset by writing 0.
		D6	OWE6	Ch.6 overwrite error flag				0	R/W	
		D5	OWE5	Ch.5 overwrite error flag				0	R/W	
		D4	OWE4	Ch.4 overwrite error flag				0	R/W	
		D3	OWE3	Ch.3 overwrite error flag				0	R/W	
		D2	OWE2	Ch.2 overwrite error flag				0	R/W	
		D1	OWE1	Ch.1 overwrite error flag				0	R/W	
		D0	OWE0	Ch.0 overwrite error flag				0	R/W	
A/D Ch.0 conversion result (low-order) buffer	0040248 (B)	D7	AD0BUF7	A/D Ch.0 converted data (low-order 8 bits) AD0BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)		0	R	Can be used when ADCADV = "1".	
		D6	AD0BUF6		0					
		D5	AD0BUF5		0					
		D4	AD0BUF4		0					
		D3	AD0BUF3		0					
		D2	AD0BUF2		0					
		D1	AD0BUF1		0					
		D0	AD0BUF0		0					
A/D Ch.0 conversion result (high-order) buffer	0040249 (B)	D7-2	–	reserved	–		–	–	0 when being read.	
		D1	AD0BUF9	A/D Ch.0 converted data (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)		0	R	Can be used when ADCADV = "1".	
		D0	AD0BUF8	AD0BUF9 = MSB			0			
A/D Ch.1 conversion result (low-order) buffer	004024A (B)	D7	AD1BUF7	A/D Ch.1 converted data (low-order 8 bits) AD1BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)		0	R	Can be used when ADCADV = "1".	
		D6	AD1BUF6		0					
		D5	AD1BUF5		0					
		D4	AD1BUF4		0					
		D3	AD1BUF3		0					
		D2	AD1BUF2		0					
		D1	AD1BUF1		0					
		D0	AD1BUF0		0					
A/D Ch.1 conversion result (high-order) buffer	004024B (B)	D7-2	–	reserved	–		–	–	0 when being read.	
		D1	AD1BUF9	A/D Ch.1 converted data (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)		0	R	Can be used when ADCADV = "1".	
		D0	AD1BUF8	AD1BUF9 = MSB			0			
A/D Ch.2 conversion result (low-order) buffer	004024C (B)	D7	AD2BUF7	A/D Ch.2 converted data (low-order 8 bits) AD2BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)		0	R	Can be used when ADCADV = "1".	
		D6	AD2BUF6		0					
		D5	AD2BUF5		0					
		D4	AD2BUF4		0					
		D3	AD2BUF3		0					
		D2	AD2BUF2		0					
		D1	AD2BUF1		0					
		D0	AD2BUF0		0					
A/D Ch.2 conversion result (high-order) buffer	004024D (B)	D7-2	–	reserved	–		–	–	0 when being read.	
		D1	AD2BUF9	A/D Ch.2 converted data (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)		0	R	Can be used when ADCADV = "1".	
		D0	AD2BUF8	AD2BUF9 = MSB			0			

IV-2 ANALOG BLOCK: A/D CONVERTER

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Ch.3 conversion result (low-order) buffer	004024E (B)	D7	AD3BUF7	A/D Ch.3 converted data (low-order 8 bits) AD3BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD3BUF6					
		D5	AD3BUF5					
		D4	AD3BUF4					
		D3	AD3BUF3					
		D2	AD3BUF2					
		D1	AD3BUF1					
		D0	AD3BUF0					
A/D Ch.3 conversion result (high-order) buffer	004024F (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	AD3BUF9	A/D Ch.3 converted data (high-order 2 bits) AD3BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	AD3BUF8			0		
A/D Ch.4 conversion result (low-order) buffer	0040250 (B)	D7	AD4BUF7	A/D Ch.4 converted data (low-order 8 bits) AD4BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD4BUF6					
		D5	AD4BUF5					
		D4	AD4BUF4					
		D3	AD4BUF3					
		D2	AD4BUF2					
		D1	AD4BUF1					
		D0	AD4BUF0					
A/D Ch.4 conversion result (high-order) buffer	0040251 (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	AD4BUF9	A/D Ch.4 converted data (high-order 2 bits) AD4BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	AD4BUF8			0		
A/D Ch.5 conversion result (low-order) buffer	0040252 (B)	D7	AD5BUF7	A/D Ch.5 converted data (low-order 8 bits) AD5BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD5BUF6					
		D5	AD5BUF5					
		D4	AD5BUF4					
		D3	AD5BUF3					
		D2	AD5BUF2					
		D1	AD5BUF1					
		D0	AD5BUF0					
A/D Ch.5 conversion result (high-order) buffer	0040253 (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	AD5BUF9	A/D Ch.5 converted data (high-order 2 bits) AD5BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	AD5BUF8			0		
A/D Ch.6 conversion result (low-order) buffer	0040254 (B)	D7	AD6BUF7	A/D Ch.6 converted data (low-order 8 bits) AD6BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD6BUF6					
		D5	AD6BUF5					
		D4	AD6BUF4					
		D3	AD6BUF3					
		D2	AD6BUF2					
		D1	AD6BUF1					
		D0	AD6BUF0					
A/D Ch.6 conversion result (high-order) buffer	0040255 (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	AD6BUF9	A/D Ch.6 converted data (high-order 2 bits) AD6BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	AD6BUF8			0		
A/D Ch.7 conversion result (low-order) buffer	0040256 (B)	D7	AD7BUF7	A/D Ch.7 converted data (low-order 8 bits) AD7BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD7BUF6					
		D5	AD7BUF5					
		D4	AD7BUF4					
		D3	AD7BUF3					
		D2	AD7BUF2					
		D1	AD7BUF1					
		D0	AD7BUF0					
A/D Ch.7 conversion result (high-order) buffer	0040257 (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	AD7BUF9	A/D Ch.7 converted data (high-order 2 bits) AD7BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	AD7BUF8			0		
A/D upper limit value (low-order) register	0040258 (B)	D7	ADUPR7	A/D conversion upper limit value (low-order 8 bits) ADUPR0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R/W	Can be used when ADCADV = "1".
		D6	ADUPR6					
		D5	ADUPR5					
		D4	ADUPR4					
		D3	ADUPR3					
		D2	ADUPR2					
		D1	ADUPR1					
		D0	ADUPR0					

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
A/D upper limit value (high-order) register	0040259 (B)	D7-2	–	reserved	–	–	–	0 when being read.	
		D1	ADUPR9	A/D conversion upper limit value (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)	0	R/W	Can be used when ADCADV = "1".	
		D0	ADUPR8	ADUPR9 = MSB		0			
A/D lower limit value (low-order) register	004025A (B)	D7	ADLWR7	A/D conversion lower limit value (low-order 8 bits)	0x0 to 0x3FF (low-order 8 bits)	0	R/W	Can be used when ADCADV = "1".	
		D6	ADLWR6			0			
		D5	ADLWR5	ADLWR0 = LSB		0			
		D4	ADLWR4			0			
		D3	ADLWR3			0			
		D2	ADLWR2			0			
		D1	ADLWR1			0			
		D0	ADLWR0			0			
		A/D lower limit value (high-order) register	004025B (B)	D7-2	–	reserved	–		–
D1	ADLWR9			A/D conversion lower limit value (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)	0	R/W	Can be used when ADCADV = "1".	
D0	ADLWR8			ADLWR9 = MSB		0			
A/D conversion complete interrupt mask register	004025C (B)	D7	INTMASK7	Ch.7 conversion-complete int. mask	1 Interrupt enabled	0 Interrupt mask	1	R/W	Can be used when ADCADV = "1".
		D6	INTMASK6	Ch.6 conversion-complete int. mask			1	R/W	
		D5	INTMASK5	Ch.5 conversion-complete int. mask			1	R/W	
		D4	INTMASK4	Ch.4 conversion-complete int. mask			1	R/W	
		D3	INTMASK3	Ch.3 conversion-complete int. mask			1	R/W	
		D2	INTMASK2	Ch.2 conversion-complete int. mask			1	R/W	
		D1	INTMASK1	Ch.1 conversion-complete int. mask			1	R/W	
		D0	INTMASK0	Ch.0 conversion-complete int. mask			1	R/W	
A/D converter mode select register	004025F (B)	D7-1	–	reserved	–	–	–	0 when being read.	
		D0	ADCADV	33209 compatible mode/advanced mode selection	1 Advanced mode	0 33209 compatible mode	0	R/W	
Serial I/F Ch.1, A/D interrupt priority register	004026A (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PAD2	A/D converter interrupt level	0 to 7	X	R/W		
		D5	PAD1						
		D4	PAD0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PSIO12	Serial interface Ch.1 interrupt level	0 to 7	X	R/W		
D1	PSIO11								
D0	PSIO10								
Port input 4-7, clock timer, A/D interrupt enable register	0040277 (B)	D7-6	–	reserved	–	–	–	0 when being read.	
		D5	EP7	Port input 7	1 Enabled	0 Disabled	0	R/W	
		D4	EP6	Port input 6			0	R/W	
		D3	EP5	Port input 5			0	R/W	
		D2	EP4	Port input 4			0	R/W	
		D1	ECTM	Clock timer			0	R/W	
		D0	EADE	A/D converter			0	R/W	
Port input 4-7, clock timer, A/D interrupt factor flag register	0040287 (B)	D7-6	–	reserved			–	–	–
		D5	FP7	Port input 7	1 Factor is generated	0 No factor is generated	X	R/W	
		D4	FP6	Port input 6			X	R/W	
		D3	FP5	Port input 5			X	R/W	
		D2	FP4	Port input 4			X	R/W	
		D1	FCTM	Clock timer			X	R/W	
		D0	FADE	A/D converter			X	R/W	
Serial I/F Ch.1, A/D, port input 4-7 IDMA request register	0040293 (B)	D7	RP7	Port input 7			1 IDMA request	0 Interrupt request	0
		D6	RP6	Port input 6	0	R/W			
		D5	RP5	Port input 5	0	R/W			
		D4	RP4	Port input 4	0	R/W			
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	RADE	A/D converter	1 IDMA request	0 Interrupt request	0	R/W	
		D1	RSTX1	SIF Ch.1 transmit buffer empty			0	R/W	
		D0	RSRX1	SIF Ch.1 receive buffer full			0	R/W	
Serial I/F Ch.1, A/D, port input 4-7 IDMA enable register	0040297 (B)	D7	DEP7	Port input 7	1 IDMA enabled	0 IDMA disabled	0	R/W	
		D6	DEP6	Port input 6			0	R/W	
		D5	DEP5	Port input 5			0	R/W	
		D4	DEP4	Port input 4			0	R/W	
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	DEADE	A/D converter	1 IDMA enabled	0 IDMA disabled	0	R/W	
		D1	DESTX1	SIF Ch.1 transmit buffer empty			0	R/W	
		D0	DESRX1	SIF Ch.1 receive buffer full			0	R/W	

## IV-2 ANALOG BLOCK: A/D CONVERTER

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
K5 function select register	00402C0 (B)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	<b>CFK54</b>	K54 function selection	1 #DMAREQ3	0 K54	0 R/W	
		D3	<b>CFK53</b>	K53 function selection	1 #DMAREQ2	0 K53	0 R/W	
		D2	<b>CFK52</b>	K52 function selection	1 #ADTRG	0 K52	0 R/W	
		D1	<b>CFK51</b>	K51 function selection	1 #DMAREQ1	0 K51	0 R/W	
		D0	<b>CFK50</b>	K50 function selection	1 #DMAREQ0	0 K50	0 R/W	
K6 function select register	00402C3 (B)	D7	<b>CFK67</b>	K67 function selection	1 AD7	0 K67	0 R/W	
		D6	<b>CFK66</b>	K66 function selection	1 AD6	0 K66	0 R/W	
		D5	<b>CFK65</b>	K65 function selection	1 AD5	0 K65	0 R/W	
		D4	<b>CFK64</b>	K64 function selection	1 AD4	0 K64	0 R/W	
		D3	<b>CFK63</b>	K63 function selection	1 AD3	0 K63	0 R/W	
		D2	<b>CFK62</b>	K62 function selection	1 AD2	0 K62	0 R/W	
		D1	<b>CFK61</b>	K61 function selection	1 AD1	0 K61	0 R/W	
		D0	<b>CFK60</b>	K60 function selection	1 AD0	0 K60	0 R/W	

### Control bits for 33209 compatible mode

**CFK52:** K52 pin function selection (D2) / K5 function select register (0x402C0)

**CFK67–CFK60:** K6[7:0] pin function selection (D[7:0]) / K6 function select register (0x402C3)

Selects the pins used by the A/D converter.

Write "1": A/D converter

Write "0": Input port

Read: Valid

When an external trigger is used, write "1" to CFK52 to set the K52 pin for external trigger input #ADTRG. Select the pin used for analog input from among K60 (AD0) through K67 (AD7) by writing "1" to CFK60 through CFK67. If the function select bit for a pin is set to "0", the pin is set for an input port.

At cold start, CFK is set to "0" (input port). At hot start, CFK retains its state from prior to the initial reset.

**ADD9–ADD0:** A/D converted data (D[1:0]) / A/D conversion result (high-order) register (0x40241)  
(D[7:0]) / A/D conversion result (low-order) register (0x40240)

Stores the results of A/D conversion.

The LSB is stored in ADD0, and the MSB is stored in ADD9. ADD0 and ADD1 are mapped to bits D0 and D1 at the address 0x40241, but bits D2 through D7 are always 0 when read.

This is a read-only register, so writing to this register is ignored.

At initial reset, the data in this register is cleared to "0".

**MS:** A/D conversion mode selection (D5) / A/D trigger register (0x40242)

Selects an A/D conversion mode.

Write "1": Continuous mode

Write "0": Normal mode

Read: Valid

The A/D converter is set for the continuous mode by writing "1" to MS. In this mode, A/D conversions in the range of the channels selected using CS and CE are executed continuously until stopped in the software.

When MS = "0", the A/D converter operates in the normal mode. In this mode, A/D conversion is completed after all inputs in the range of the channels selected by CS and CE are converted in one operation.

At initial reset, MS is set to "0" (normal mode).

**TS1–TS0: Trigger selection (D[4:3]) / A/D trigger register (0x40242)**

Selects a trigger to start A/D conversion.

Table IV.2.9 Trigger Selection

TS1	TS0	Trigger
1	1	External trigger (K52/#ADTRG)
1	0	8-bit programmable timer 0
0	1	16-bit programmable timer 0
0	0	Software

When an external trigger is used, use the CFK52 bit to set the K52 pin for #ADTRG.

When a programmable timer is used, since its underflow signal (8-bit timer) or comparison match B signal (16-bit timer) serves as a trigger, set the cycle and other parameters for the programmable timer.

At initial reset, TS is set to "0" (software trigger).

**CH2–CH0: Conversion channel status (D[2:0]) / A/D trigger register (0x40242)**

Indicates the channel number (0 to 7) currently being A/D-converted.

When A/D conversion is performed in multiple channels, read this bit to identify the channel in which conversion is underway.

At initial reset, CH is set to "0" (AD0).

**CE2–CE0: Conversion end-channel setup (D[5:3]) / A/D channel register (0x40243)**

Sets the conversion end channel by selecting a channel number from 0 to 7.

Analog inputs can be A/D-converted successively from the channel set using CS to the channel set using this bit in one operation. If only one channel is to be A/D converted, set the same channel number in both the CS and CE bits.

At initial reset, CE is set to "0" (AD0).

**CS2–CS0: Conversion start-channel setup (D[2:0]) / A/D channel register (0x40243)**

Sets the conversion start channel by selecting a channel number from 0 to 7.

Analog inputs can be A/D-converted successively from the channel set using this bit to the channel set using CE in one operation. If only one channel is to be A/D converted, set the same channel number in both the CS and CE bits.

At initial reset, CS is set to "0" (AD0).

**ADF: Conversion-complete flag (D3) / A/D enable register (0x40244)**

Indicates that A/D conversion has been completed.

Read "1": Conversion completed

Read "0": Being converted or standing by

Write: Invalid

This flag is set to "1" when A/D conversion is completed, and the converted data is stored in the data register and is reset to "0" when the converted data is read out. When A/D conversion is performed in multiple channels, if the next A/D conversion is completed while ADF = "1" (before the converted data is read out), the data register is overwritten with the new conversion results, causing an overrun error to occur. Therefore, ADF must be reset by reading out the converted data before the next A/D conversion is completed.

At initial reset, ADF is set to "0" (being converted or standing by).

**ADE: A/D enable (D2) / A/D enable register (0x40244)**

Enables the A/D converter (readied for conversion).

Write "1": Enabled

Write "0": Disabled

Read: Valid

When ADE is set to "1", the A/D converter is enabled, meaning it is ready to start A/D conversion (i.e., ready to accept a trigger). When ADE = "0", the A/D converter is disabled, meaning it is unable to accept a trigger.

Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to reset ADE to "0". This helps to prevent the A/D converter from operating erratically.

At initial reset, ADE is set to "0" (disabled).

---

**ADST:** A/D conversion control/status (D1) / A/D enable register (0x40244)

---

Controls A/D conversion.

- Write "1": Software trigger
- Write "0": A/D conversion is stopped
- Read: Valid

If A/D conversion is to be started by a software trigger, set ADST to "1". If any other trigger is used, ADST is automatically set to "1" by the hardware.

ADST remains set while A/D conversion is underway.

In normal mode, upon completion of A/D conversion in selected channels, ADST is reset to "0" and the A/D conversion circuit is turned off. To stop A/D conversion during operation in continuous mode, reset ADST by writing "0".

When ADE = "0" (A/D conversion disabled), ADST is fixed to "0", with no trigger accepted.

At initial reset, ADST is set to "0" (A/D conversion stopped).

---

**OWE:** Overwrite-error flag (D0) / A/D enable register (0x40244)

---

Indicates that the converted data has been overwritten.

- Read "1": Overwritten
- Read "0": Normal
- Write "1": Invalid
- Write "0": Flag is set

During A/D conversion in multiple channels, if the conversion results for the next channel are written to the converted-data register (overwritten) before the converted data is read out to reset the conversion-complete flag ADF that has been set through conversion of the preceding channel, OWE is set to "1". When ADF is reset, because this means that the converted data has been read out, OWE is not set.

Once OWE is set to "1", it remains set until it is reset by writing "0" in the software.

At initial reset, OWE is set to "0" (normal).

---

**ST1–ST0:** Sampling-time setup (D[1:0]) / A/D sampling register (0x40245)

---

Sets the analog input sampling time.

Table IV.2.10 Sampling Time

ST1	ST0	Sampling Time
1	1	9-clock period
1	0	7-clock period
0	1	5-clock period
0	0	3-clock period

The A/D converter input clock is used for counting.

At initial reset, ST is set to "11" (9-clock period).

To maintain the conversion accuracy, use ST as set by default (9-clock period).

---

**PAD2–PAD0:** A/D converter interrupt level (D[6:4]) / Serial I/F Ch.1, A/D interrupt priority register (0x4026A)

---

Sets the priority level of the A/D-converter interrupt in the range of 0 to 7.

At initial reset, PAD becomes indeterminate.

---

**EADE:** A/D converter interrupt enable (D0) / Port input 4–7, clock timer, A/D interrupt enable register (0x40277)

---

Enables or disables an interrupt to the CPU generated by the A/D converter.

- Write "1": Interrupt enabled
- Write "0": Interrupt disabled
- Read: Valid

EADE is an interrupt enable bit to control the A/D converter interrupt.

When EADE is set to "1", the A/D converter interrupt is enabled. When EADE is set to "0", the A/D-converter interrupt is disabled.

At initial reset, EADE is set to "0" (interrupt disabled).

---

**FADE:** A/D converter interrupt factor flag (D0) / Port input 4–7, clock timer, A/D interrupt factor flag register (0x40287)

Indicates the status of an A/D-converter interrupt factor generated.

**When read**

Read "1": Interrupt factor has occurred

Read "0": No interrupt factor has occurred

**When written using the reset-only method (default)**

Write "1": Interrupt factor flag is reset

Write "0": Invalid

**When written using the read/write method**

Write "1": Interrupt flag is set

Write "0": Interrupt flag is reset

FADE is the A/D converter interrupt factor flag and is set to "1" when A/D conversion in each channel has completed (and when the conversion result is out of range in the advanced mode if it is enabled).

At this time, if the following conditions are met, an interrupt to the CPU is generated:

1. The corresponding interrupt enable register bit is set to "1".
2. No other interrupt request of a higher priority has been generated.
3. The IE bit of the PSR is set to "1" (interrupts enabled).
4. The value set in the corresponding interrupt priority register is higher than the interrupt level (IL) of the CPU.

When using the interrupt factor of the A/D converter to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of IDMA, an interrupt is generated under the above conditions after the data transfer by IDMA is completed.

The interrupt factor flag is set to "1" whenever interrupt generation conditions are met, regardless of how the interrupt enable and interrupt priority registers are set.

If the next interrupt is to be accepted after an interrupt has occurred, it is necessary that the interrupt factor flag be reset, and that the PSR be set again (by setting the IE bit to "1" after setting the IL to a value lower than the level indicated by the interrupt priority register, or by executing the reti instruction).

The interrupt factor flag can be reset only by writing to it in the software. Note that if the PSR is set again to accept interrupts generated (or if the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt occurs again. Note also that the value to be written to reset the flag is "1" when the reset-only method (RSTONLY = "1") is used, and "0" when the read/write method (RSTONLY = "0") is used.

At initial reset, the content of FADE becomes indeterminate, so be sure to reset it in the software.

---

**RADE:** A/D converter IDMA request (D2) / Serial I/F Ch.1, A/D, port input 4–7 IDMA request register (0x40293)

Specifies whether to invoke IDMA when an interrupt factor occurs.

**When using the set-only method (default)**

Write "1": IDMA request

Write "0": Not changed

Read: Valid

**When using the read/write method**

Write "1": IDMA request

Write "0": Interrupt request

Read: Valid

When RADE is set to "1", IDMA is invoked when an interrupt factor occurs, thereby performing a programmed data transfer. If RADE is set to "0", normal interrupt processing is performed, without invoking IDMA.

For details on IDMA, refer to "IDMA (Intelligent DMA)".

At initial reset, RADE is set to "0" (interrupt request).

---

**DEADE:** A/D converter IDMA enable (D2) / Serial I/F Ch.1, A/D, port input 4–7 IDMA enable register (0x40297)

Enables IDMA transfer by means of an interrupt factor.

**When using the set-only method (default)**

Write "1": IDMA enabled

Write "0": Not changed

Read: Valid

**When using the read/write method**

Write "1": IDMA enabled

Write "0": IDMA disabled

Read: Valid

If DEADE is set to "1", the IDMA request by the interrupt factor is enabled. If this bit is set to "0", the IDMA request is disabled.

After an initial reset, DEADE is set to "0" (IDMA disabled).



## Control bits for advanced mode

The following control bits can be used in the advanced mode as well as the control bit described above.

---

**ADCADV:** 33209 compatible mode/advanced mode selection (D0) / A/D converter mode select register (0x4025F)

Selects the A/D converter operating mode.

Write "1": Advanced mode

Write "0": 33209 compatible mode

Read: Valid

When ADCADV is set to "1", the A/D converter is set in the advanced mode, and the registers/bits for the extended function described below can be used.

When ADCADV is set to "0", only the standard C33 A/D converter functions implemented in the S1C33209, etc., can be used. In this mode, the extended registers/bits for advanced mode become read only and writing operation is disabled.

At initial reset, ADCADV is set to "0" (33209 compatible mode).

---

**AD0BUF9–AD0BUF0:** A/D Ch.0 conversion result / A/D Ch.0 conversion result buffer (D[1:0] /0x40249, D[7:0]/0x40248)

**AD1BUF9–AD1BUF0:** A/D Ch.1 conversion result / A/D Ch.1 conversion result buffer (D[1:0] /0x4024B, D[7:0]/0x4024A)

**AD2BUF9–AD2BUF0:** A/D Ch.2 conversion result / A/D Ch.2 conversion result buffer (D[1:0] /0x4024D, D[7:0]/0x4024C)

**AD3BUF9–AD3BUF0:** A/D Ch.3 conversion result / A/D Ch.3 conversion result buffer (D[1:0] /0x4024F, D[7:0]/0x4024E)

**AD4BUF9–AD4BUF0:** A/D Ch.4 conversion result / A/D Ch.4 conversion result buffer (D[1:0] /0x40251, D[7:0]/0x40250)

**AD5BUF9–AD5BUF0:** A/D Ch.5 conversion result / A/D Ch.5 conversion result buffer (D[1:0] /0x40253, D[7:0]/0x40252)

**AD6BUF9–AD6BUF0:** A/D Ch.6 conversion result / A/D Ch.6 conversion result buffer (D[1:0] /0x40255, D[7:0]/0x40254)

**AD7BUF9–AD7BUF0:** A/D Ch.7 conversion result / A/D Ch.7 conversion result buffer (D[1:0] /0x40257, D[7:0]/0x40256)

The conversion results in each channel are stored.

This is a read-only register, so writing to this register is ignored.

At initial reset, the buffer is cleared to "0".

---

**ADF0:** Ch.0 conversion-complete flag (D0) / A/D conversion-complete flag register (0x40246)

**ADF1:** Ch.1 conversion-complete flag (D1) / A/D conversion-complete flag register (0x40246)

**ADF2:** Ch.2 conversion-complete flag (D2) / A/D conversion-complete flag register (0x40246)

**ADF3:** Ch.3 conversion-complete flag (D3) / A/D conversion-complete flag register (0x40246)

**ADF4:** Ch.4 conversion-complete flag (D4) / A/D conversion-complete flag register (0x40246)

**ADF5:** Ch.5 conversion-complete flag (D5) / A/D conversion-complete flag register (0x40246)

**ADF6:** Ch.6 conversion-complete flag (D6) / A/D conversion-complete flag register (0x40246)

**ADF7:** Ch.7 conversion-complete flag (D7) / A/D conversion-complete flag register (0x40246)

Indicates that A/D conversion in each channel has been completed.

Read "1": Conversion completed

Read "0": Being converted or standing by

Write: Invalid

This flag is set to "1" when A/D conversion is completed, and the converted data is stored in the conversion result buffer and is reset to "0" when the conversion result buffer is read out. When A/D conversion is performed in continuous mode, if the next A/D conversion of the same channel is completed while ADF<sub>x</sub> = "1" (before the conversion result buffer is read out), the buffer is overwritten with the new conversion results, causing an overrun error to occur. Therefore, ADF<sub>x</sub> must be reset by reading out the converted data before the next A/D conversion is completed.

At initial reset, ADF<sub>x</sub> is set to "0" (being converted or standing by).

---

**OWE0:** Ch.0 overwrite error flag (D0) / A/D overwrite error flag register (0x40247)  
**OWE1:** Ch.1 overwrite error flag (D1) / A/D overwrite error flag register (0x40247)  
**OWE2:** Ch.2 overwrite error flag (D2) / A/D overwrite error flag register (0x40247)  
**OWE3:** Ch.3 overwrite error flag (D3) / A/D overwrite error flag register (0x40247)  
**OWE4:** Ch.4 overwrite error flag (D4) / A/D overwrite error flag register (0x40247)  
**OWE5:** Ch.5 overwrite error flag (D5) / A/D overwrite error flag register (0x40247)  
**OWE6:** Ch.6 overwrite error flag (D6) / A/D overwrite error flag register (0x40247)  
**OWE7:** Ch.7 overwrite error flag (D7) / A/D overwrite error flag register (0x40247)

---

Indicates that the conversion result buffer has been overwritten.

Read "1": Overwritten  
 Read "0": Normal  
 Write "1": Invalid  
 Write "0": Flag is set

During A/D conversion in continuous mode, if the new conversion results in the same channel are written to the conversion result buffer (overwritten) before the converted data is read out to reset the ADFx conversion-complete flag that has been set through the previous conversion, OWEx is set to "1". When ADFx is reset, because this means that the converted data has been read out, OWEx is not set.

Once OWEx is set to "1", it remains set until it is reset by writing "0" in the software.

At initial reset, OWEx is set to "0" (normal).

---

**INTMASK0:** Ch.0 conversion-complete interrupt mask (D0) / A/D conversion-complete interrupt mask register (0x4025C)  
**INTMASK1:** Ch.1 conversion-complete interrupt mask (D1) / A/D conversion-complete interrupt mask register (0x4025C)  
**INTMASK2:** Ch.2 conversion-complete interrupt mask (D2) / A/D conversion-complete interrupt mask register (0x4025C)  
**INTMASK3:** Ch.3 conversion-complete interrupt mask (D3) / A/D conversion-complete interrupt mask register (0x4025C)  
**INTMASK4:** Ch.4 conversion-complete interrupt mask (D4) / A/D conversion-complete interrupt mask register (0x4025C)  
**INTMASK5:** Ch.5 conversion-complete interrupt mask (D5) / A/D conversion-complete interrupt mask register (0x4025C)  
**INTMASK6:** Ch.6 conversion-complete interrupt mask (D6) / A/D conversion-complete interrupt mask register (0x4025C)  
**INTMASK7:** Ch.7 conversion-complete interrupt mask (D7) / A/D conversion-complete interrupt mask register (0x4025C)

---

Masks the A/D conversion-complete interrupt for each channel individually.

Write "1": Interrupt is enabled  
 Write "0": Interrupt is masked  
 Read: Valid

When INTMASKx is set to "0", the conversion-completed interrupt request of the Ch.x is masked and the interrupt factor flag FADE will not be set to "1" even if A/D conversion is completed. When INTMASKx is "1", the A/D converter can generate an interrupt upon completion of A/D conversion in Ch.x.

At initial reset, INTMASKx is set to "1" (interrupt is enabled).

---

**CNVINTEN:** Conversion-complete interrupt enable (D4) / A/D enable register (0x40244)

---

Enables/disables the conversion-complete interrupt.

Write "1": Enabled  
 Write "0": Disabled  
 Read: Valid

When CNVINTEN is set to "1", completion of an A/D conversion becomes a cause of an interrupt. When it is set to "0", a conversion-complete interrupt is not generated.

At initial reset, CNVINTEN is set to "1" (enabled).

**CMPINTEN: Out-of-range interrupt enable (D5) / A/D enable register (0x40244)**

Enables/disables the out-of-range interrupt.

Write "1": Enabled  
 Write "0": Disabled  
 Read: Valid

When CMPINTEN is set to "1", upper and lower-limit comparison results become a cause of an interrupt. When it is set to "0", an out-of-range interrupt is not generated.

At initial reset, CMPINTEN is set to "0" (disabled).

**INTMODE: Interrupt signal mode (D6) / A/D enable register (0x40244)**

Configures the interrupt signal delivered to the ITC.

Write "1": Conversion-complete signal only  
 Write "0": OR between conversion-complete and out-of-range signals  
 Read: Valid

INTMODE selects whether the interrupt signal line connected to the ITC is used to send the conversion-complete signal only or used to send the signal of which the conversion-complete and out-of-range signal are ORed.

Set INTMODE to "1" if the out-of-range interrupt is not used. When using the out-of-range interrupt, set INTMODE to "0" and CMPINTEN to "1".

At initial reset, INTMODE is set to "0" (ORed signal).

**ADCMPE: Upper-limit and lower-limit comparison enable (D7) / A/D sampling register (0x40245)**

Enables/disables comparison between converted data and upper-/lower-limit values.

Write "1": Enabled  
 Write "0": Disabled  
 Read: Valid

ADCMPE selects whether the converted data is compared with the upper-/lower-limit values after A/D conversion of the channel specified using ADCMP[2:0]. Set ADCMPE to "1" when using the comparison function or set to "0" when not used.

At initial reset, ADCMPE is set to "0" (disabled).

**ADCMP2–ADCMP0: Upper/lower limit comparison channel selection (D6) / A/D sampling register (0x40245)**

Set the channel number (0–7) to compare its converted data with the upper-/ lower-limit values.

At initial reset, ADCMP is set to "0" (AD0).

**ADUPR9–ADUPR0: A/D conversion upper-limit value / A/D upper-limit register (D[1:0]/0x40259, D[7:0]/0x40258)**

Set the upper-limit value to be compared with the A/D conversion results.

The value set in this register is used for the range check of the A/D conversion results in the channel specified with ADCMP[2:0]. If the converted data exceeds the set value, an interrupt can be generated.

At initial reset, ADUPR is set to "0".

**ADLWR9–ADLWR0: A/D conversion lower-limit value / A/D lower-limit register (D[1:0]/0x4025B, D[7:0]/0x4025A)**

Set the lower-limit value to be compared with the A/D conversion results.

The value set in this register is used for the range check of the A/D conversion results in the channel specified with ADCMP[2:0]. If the converted data is less than the set value, an interrupt can be generated.

At initial reset, ADLWR is set to "0".

---

**ADUPRST:** Upper-limit value comparison status (D3) / A/D sampling register (0x40245)

---

Indicates the results of comparison between the A/D converted data and the upper-limit value.

- Read "1": Exceeded the upper limit
- Read "0": Within the range
- Write: Invalid

When the upper-/lower-limit comparison function is enabled (ADCMPE = "1"), the converted data is compared with the upper-/lower-limit values after A/D conversion of the channel specified using ADCMP[2:0] has completed. If the converted data exceeds the upper-limit value set in the ADUPR register, ADUPRST is set to "1". If the converted data is equal to or less than the upper-limit value, ADUPRST is set to "0". An interrupt occurs when ADUPRST is set to "1" if the out-of-range interrupt is enabled.

At initial reset, ADUPRST is set to "0" (within the range).

---

**ADLWRST:** Lower-limit value comparison status (D2) / A/D sampling register (0x40245)

---

Indicates the results of comparison between the A/D converted data and the lower-limit value.

- Read "1": Under the lower limit
- Read "0": Within the range
- Write: Invalid

When the upper-/lower-limit comparison function is enabled (ADCMPE = "1"), the converted data is compared with the upper-/lower-limit values after A/D conversion of the channel specified using ADCMP[2:0] has completed. If the converted data is less than the lower-limit value set in the ADLWR register, ADLWRST is set to "1". If the converted data is equal to or more than the upper-lower value, ADLWRST is set to "0". An interrupt occurs when ADLWRST is set to "1" if the out-of-range interrupt is enabled.

At initial reset, ADLWRST is set to "0" (within the range).

## Programming Notes

- (1) Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to disable the A/D converter (ADE (D2) / A/D enable register (0x40244) = "0"). A change in settings while the A/D converter is enabled could cause it to operate erratically.
- (2) The A/D converter operates only when the prescaler is operating.  
In consideration of the conversion accuracy, we recommend that the A/D converter operating clock be min. 16 kHz to max. 2 MHz.
- (3) Do not start an A/D conversion when the clock supplied from the prescaler to the A/D converter is turned off, and do not turn off the prescaler's clock output when an A/D conversion is underway, as doing so could cause the A/D converter to operate erratically.
- (4) After an initial reset, the interrupt factor flag (FADE) becomes indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in a program.
- (5) To prevent the regeneration of interrupts due to the same factor following the occurrence an interrupt, always be sure to reset the interrupt factor flag before setting the PSR again or executing the reti instruction.
- (6) When the A/D converter is set to enabled state, a current flows between AV<sub>DDE</sub> and V<sub>SS</sub>, and power is consumed, even when A/D operations are not performed. Therefore, when the A/D converter is not used, it must be set to the disabled state (default "0" setting of ADE (D2) in the A/D enable register (0x40244)).
- (7) When the 8-bit programmable timer 0 underflow signal or the 16-bit programmable timer 0 compare match B signal is used as a trigger factor, the division ratio of the prescaler used by the relevant timer must not be set to 0/1.
- (8) ADD[9:0] (A/D conversion results) is read twice, once in the low-order 8 bits and once in the high-order 2 bits. (The hardware loads the results in this manner even if the software reads the register in 16 bits.)  
In continuous mode or when two or more channels are converted successively in normal mode, ADD[9:0] may be overwritten with the new conversion results between reading of the low-order 8 bits and high-order 2 bits. In this case, correct conversion results cannot be obtained because the low-order 8 bits and the high-order 2 bits are not the results of the same conversion.  
At the 1st reading of the conversion results after an A/D conversion has completed (when the conversion-complete flag ADF is set to "1"), the overwrite-error flag OWE is set to "1" if ADD[9:0] is overwritten between reading of the low-order 8 bits and high-order 2 bits. Note, however, that OWE is not set to "1" even if ADD[9:0] is overwritten when the same conversion results have already been read (when ADF is reset to "0"). This may occur when the program reads the same results twice or more for verification or other purposes. This problem may also occur when the conversion result buffer ADxBUF[9:0] is read.
- (9) When using an external trigger to start A/D conversion, the low period of the trigger signal to be applied to the #ADTRG pin must be two or more CPU operating clock cycles. Furthermore, return the #ADTRG input level to high within 20 cycles of the A/D input clock set. Otherwise, it will be detected as the trigger for the next A/D conversion.
- (10) Depending on the model, software controllable pull-up resistors may be provided for the input ports. In this case, disable the pull-up resistors of the ports used for analog inputs.  
(Example of Models that support pull-up: S1C33301, S1C33L11)
- (11) When in break mode during ICD-based debugging, the operating clock for the A/D converter is turned off due to the internal chip design. Therefore, the A/D converter stops operating and registers cannot be accessed for write (but can be accessed for read).

THIS PAGE IS BLANK.

S1C33301 Technical Manual  
**V DMA BLOCK**





# V-1 INTRODUCTION

The DMA Block is configured with two types of DMA controllers: HSDMA (High-Speed DMA) that has on-chip registers for controlling DMA command information and IDMA (Intelligent DMA) that uses a memory area for storing DMA command information.

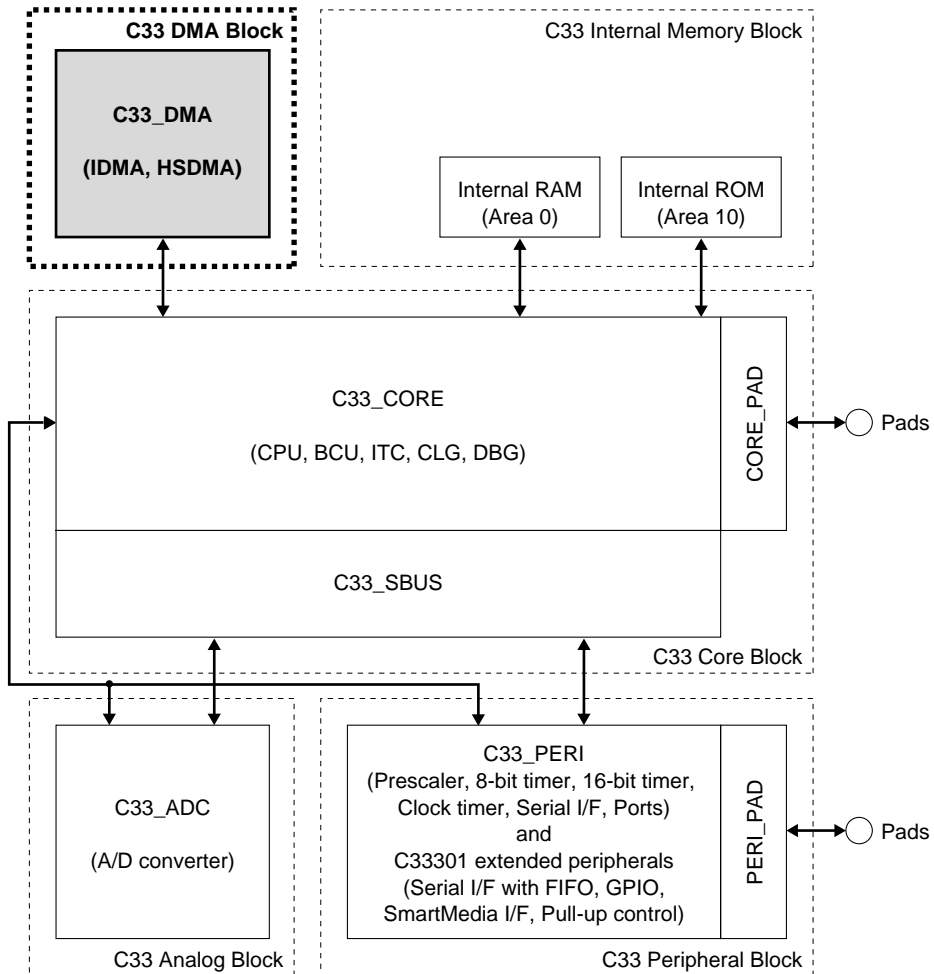


Figure V.1.1 DMA Block

**Note:** Internal ROM is not provided in the S1C33301.

THIS PAGE IS BLANK.

## V-2 HSDMA (High-Speed DMA)

### Functional Outline of HSDMA

The DMA Block contains four channels of HSDMA (High-Speed DMA) circuits that support dual-address transfer and single-address transfer methods.

Since the control registers required for the DMA function are built into the chip, DMA requests for data transfer can be responded to instantaneously.

#### Dual-address transfer

In this method, a source address and a destination address for DMA transfer can be specified and a DMA transfer is performed in two phases. The first phase reads data at the source address into the on-chip temporary register. The second phase writes the temporary register data to the destination address.

Unlike IDMA (Intelligent DMA), which has transfer information in memory, this DMA method does not support a DMA link function but allows high-speed data transfers because it is not necessary to read transfer information from a memory.

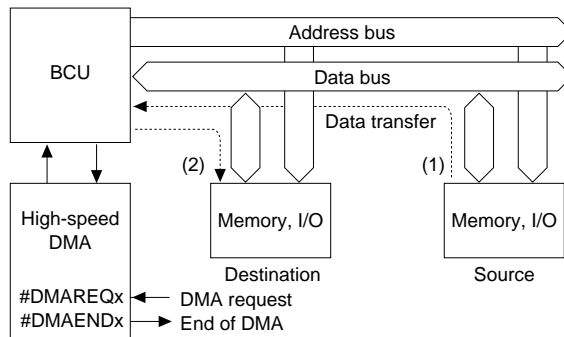


Figure V.2.1 Dual-Address Transfer Method

#### Single-address transfer

In this method, data transfers that are normally accomplished by executing data read and write operations back-to-back are executed on the external bus collectively at one time, thus further speeding up the transfer operation. The #DMAACKx and #DMAENDx signals are used to control data transfer.

Unlike dual-address transfer, this method does not allow memory to memory data transfer but data transfers can be performed in minimum cycles.

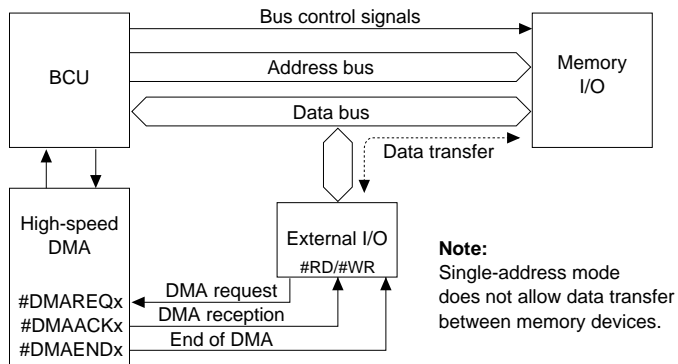


Figure V.2.2 Single-Address Transfer Method

**Note:** Channels 0 to 3 are configured in the same way and have the same functionality. Signal and control bit names are assigned channel numbers 0 to 3 to distinguish them from other channels. In this manual, however, channel numbers 0 to 3 are designated with an "x" except where they must be distinguished, as the explanation is the same for all channels.

## I/O Pins of HSDMA

Table V.2.1 lists the I/O pins used for HSDMA.

Table V.2.1 I/O Pins of HSDMA

Pin name	I/O	Function	Function select bit
K50(#DMAREQ0)	I	Input port / High-speed DMA request 0	CFK50(D0)/K5 function select register(0x402C0)
K51(#DMAREQ1)	I	Input port / High-speed DMA request 1	CFK51(D1)/K5 function select register(0x402C0)
K53(#DMAREQ2)	I	Input port / High-speed DMA request 2	CFK53(D3)/K5 function select register(0x402C0)
K54(#DMAREQ3)	I	Input port / High-speed DMA request 3	CFK54(D4)/K5 function select register(0x402C0)
P04(SIN1/#DMAACK2)	I/O	I/O port / Serial IF Ch.1 data input / #DMAACK2 output (Ex)	CFEX4(D4)/Port function extension register(0x402DF)
P05(SOUT1/#DMAEND2)	I/O	I/O port / Serial IF Ch.1 data output / #DMAEND2 output (Ex)	CFEX5(D5)/Port function extension register(0x402DF)
P06(#SCLK1/#DMAACK3)	I/O	I/O port / Serial IF Ch.1 clock input/output / #DMAACK3 output (Ex)	CFEX6(D6)/Port function extension register(0x402DF)
P07(#SRDY1/#DMAEND3)	I/O	I/O port / Serial IF Ch.1 ready input/output / #DMAEND3 output (Ex)	CFEX7(D7)/Port function extension register(0x402DF)
P15(EXCL4/#DMAEND0)	I/O	I/O port / 16-bit timer 4 event counter input (I) / #DMAEND0 output (O)	CFP15(D5)/P1 function select register(0x402D4)
P16(EXCL5/#DMAEND1)	I/O	I/O port / 16-bit timer 5 event counter input (I) / #DMAEND1 output (O)	CFP16(D6)/P1 function select register(0x402D4)
P32(#DMAACK0)	I/O	I/O port / #DMAACK0 output	CFP32(D2)/P3 function select register(0x402DC)
P33(#DMAACK1)	I/O	I/O port / #DMAACK1 output	CFP33(D3)/P3 function select register(0x402DC)

(I): Input mode, (O): Output mode, (Ex): Extended function

### #DMAREQx (DMA request input pin)

This pin is used to input a DMA request signal from an external peripheral circuit. One data transfer operation is performed by this trigger (either the rising edge or the falling edge of the signal can be selected). The #DMAREQ0 to #DMAREQ3 pins correspond to channel 0 to channel 3, respectively.

In addition to this external input, software trigger or an interrupt factor can be selected for the HSDMA trigger factor using the register in the interrupt controller.

### #DMAACKx (DMA acknowledge signal output pin for single-address mode)

This signal is output to indicate that a DMA request has been acknowledged by the DMA controller.

In single-address mode, the I/O device that is the source or destination of transfer outputs data to the external bus or takes in data from the external data synchronously with this signal.

The #DMAACK0 to #DMAACK3 pins correspond to channel 0 to channel 3, respectively.

This signal is not output in dual-address mode.

### #DMAENDx (End-of-transfer signal output pin)

This signal is output to indicate that the number of data transfer operations that is set in the control register have been completed. The #DMAEND0 to #DMAEND3 pins correspond to channel 0 to channel 3, respectively.

### Method for setting HSDMA I/O pins

As shown in Table V.2.1, the pins used for HSDMA are shared with input ports and I/O ports. At cold start, all of these are set as input and I/O port pins (function select register = "0"). According to the signals to be used, set the corresponding pin function select bit by writing "1". At hot start, the register retains the previous status before a reset.

The #DMAEND3, #DMAACK3, #DMAEND2 and #DMAACK2 outputs are the extended functions of the P04 to P07 ports. When using these signals, the extended function bit (CFEX[7:4]) must be set to "1".

In addition, setup of the #DMAEND0 pin or #DMAEND1 pin further requires setting the I/O port's I/O control bit IOC15 (D5) or IOC16 (D6) / P1 I/O control register (0x402D6) by writing "1" in order to direct the pin for output. If this pin is directed for input, it functions as a 16-bit programmable timer's event counter input and cannot be used to output the #DMAENDx signal. At cold start, this pin is set for input. At hot start, it retains the previous status.

## Programming Control Information

---

The HSDMA operates according to the control information set in the registers.

Note that some control bits change their functions according to the address mode.

The following explains how to set the contents of control information. Before using HSDMA, make each the settings described below.

### Setting the Registers in Dual-Address Mode

Make sure that the HSDMA channel is disabled (HSx\_EN = "0") before setting the control information.

#### Address mode

The address mode select bit DUALMx should be set to "1" (dual-address mode). This bit is set to "0" (single-address mode) at initial reset.

DUALM0: Ch. 0 address mode selection (DF) / HSDMA Ch. 0 control register (0x48222)

DUALM1: Ch. 1 address mode selection (DF) / HSDMA Ch. 1 control register (0x48232)

DUALM2: Ch. 2 address mode selection (DF) / HSDMA Ch. 2 control register (0x48242)

DUALM3: Ch. 3 address mode selection (DF) / HSDMA Ch. 3 control register (0x48252)

#### Transfer mode

A transfer mode should be set using the DxMOD[1:0] bits.

D0MOD[1:0]: Ch. 0 transfer mode (D[F:E]) / HSDMA Ch. 0 high-order destination address set-up register (0x4822A)

D1MOD[1:0]: Ch. 1 transfer mode (D[F:E]) / HSDMA Ch. 1 high-order destination address set-up register (0x4823A)

D2MOD[1:0]: Ch. 2 transfer mode (D[F:E]) / HSDMA Ch. 2 high-order destination address set-up register (0x4824A)

D3MOD[1:0]: Ch. 3 transfer mode (D[F:E]) / HSDMA Ch. 3 high-order destination address set-up register (0x4825A)

The following three transfer modes are available:

##### Single transfer mode (DxMOD = "00", default)

In this mode, a transfer operation invoked by one trigger is completed after transferring one unit of data of the size set by DATSIZEx. If data transfer need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

##### Successive transfer mode (DxMOD = "01")

In this mode, data transfer operations are performed by one trigger a number of times as set by the transfer counter. The transfer counter is decremented to 0 each time data is transferred.

##### Block transfer mode (DxMOD = "10")

In this mode, a transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKLENx. If a block transfer need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

#### Transfer data size

The DATSIZEx bit is used to set the unit size of data to be transferred.

A half-word size (16 bits) is assumed if this bit is "1" and a byte size (8 bits) is assumed if this bit is "0" (default).

DATSIZE0: Ch. 0 transfer data size (DE) / HSDMA Ch. 0 high-order source address set-up register (0x48226)

DATSIZE1: Ch. 1 transfer data size (DE) / HSDMA Ch. 1 high-order source address set-up register (0x48236)

DATSIZE2: Ch. 2 transfer data size (DE) / HSDMA Ch. 2 high-order source address set-up register (0x48246)

DATSIZE3: Ch. 3 transfer data size (DE) / HSDMA Ch. 3 high-order source address set-up register (0x48256)

## Block length

When using block transfer mode (DxMOD = "10"), the data block length (in units of DATSIZE<sub>x</sub>) should be set using the BLKLEN<sub>x</sub>[7:0] bits.

BLKLEN0[7:0]: Ch. 0 block length (D[7:0]) / HSDMA Ch. 0 transfer counter register (0x48220)

BLKLEN1[7:0]: Ch. 1 block length (D[7:0]) / HSDMA Ch. 1 transfer counter register (0x48230)

BLKLEN2[7:0]: Ch. 2 block length (D[7:0]) / HSDMA Ch. 2 transfer counter register (0x48240)

BLKLEN3[7:0]: Ch. 3 block length (D[7:0]) / HSDMA Ch. 3 transfer counter register (0x48250)

**Note:** When performing data transfer in block transfer mode, the block size must not be set to "0".

In single transfer and successive transfer modes, these bits are used as the bits7–0 of the transfer counter.

## Transfer counter

### Block transfer mode

In block transfer mode, up to 16 bits of transfer count can be specified.

TC0\_L[7:0]: Ch. 0 transfer counter [7:0] (D[F:8]) / HSDMA Ch. 0 transfer counter register (0x48220)

TC1\_L[7:0]: Ch. 1 transfer counter [7:0] (D[F:8]) / HSDMA Ch. 1 transfer counter register (0x48230)

TC2\_L[7:0]: Ch. 2 transfer counter [7:0] (D[F:8]) / HSDMA Ch. 2 transfer counter register (0x48240)

TC3\_L[7:0]: Ch. 3 transfer counter [7:0] (D[F:8]) / HSDMA Ch. 3 transfer counter register (0x48250)

TC0\_H[7:0]: Ch. 0 transfer counter [15:8] (D[7:0]) / HSDMA Ch. 0 control register (0x48222)

TC1\_H[7:0]: Ch. 1 transfer counter [15:8] (D[7:0]) / HSDMA Ch. 1 control register (0x48232)

TC2\_H[7:0]: Ch. 2 transfer counter [15:8] (D[7:0]) / HSDMA Ch. 2 control register (0x48242)

TC3\_H[7:0]: Ch. 3 transfer counter [15:8] (D[7:0]) / HSDMA Ch. 3 control register (0x48252)

### Single transfer and successive transfer modes

In single transfer and successive transfer modes, up to 24 bits of transfer count can be specified.

BLKLEN0[7:0]: Ch. 0 transfer counter [7:0] (D[7:0]) / HSDMA Ch.0 transfer counter register (0x48220)

BLKLEN1[7:0]: Ch. 1 transfer counter [7:0] (D[7:0]) / HSDMA Ch.1 transfer counter register (0x48230)

BLKLEN2[7:0]: Ch. 2 transfer counter [7:0] (D[7:0]) / HSDMA Ch.2 transfer counter register (0x48240)

BLKLEN3[7:0]: Ch. 3 transfer counter [7:0] (D[7:0]) / HSDMA Ch.3 transfer counter register (0x48250)

TC0\_L[7:0]: Ch. 0 transfer counter [15:8] (D[F:8]) / HSDMA Ch. 0 transfer counter register (0x48220)

TC1\_L[7:0]: Ch. 1 transfer counter [15:8] (D[F:8]) / HSDMA Ch. 1 transfer counter register (0x48230)

TC2\_L[7:0]: Ch. 2 transfer counter [15:8] (D[F:8]) / HSDMA Ch. 2 transfer counter register (0x48240)

TC3\_L[7:0]: Ch. 3 transfer counter [15:8] (D[F:8]) / HSDMA Ch. 3 transfer counter register (0x48250)

TC0\_H[7:0]: Ch. 0 transfer counter [23:16] (D[7:0]) / HSDMA Ch. 0 control register (0x48222)

TC1\_H[7:0]: Ch. 1 transfer counter [23:16] (D[7:0]) / HSDMA Ch. 1 control register (0x48232)

TC2\_H[7:0]: Ch. 2 transfer counter [23:16] (D[7:0]) / HSDMA Ch. 2 control register (0x48242)

TC3\_H[7:0]: Ch. 3 transfer counter [23:16] (D[7:0]) / HSDMA Ch. 3 control register (0x48252)

**Note:** The transfer count thus set is decremented according to the transfers performed. If the transfer count is set to 0, it is decremented to all Fs by the first transfer performed. This means that you have set the maximum value that is determined by the number of bits available.

## Source and destination addresses

In dual-address mode, a source address and a destination address for DMA transfer can be specified.

S0ADRL[15:0]: Ch. 0 source address [15:0] (D[F:0]) / Ch. 0 low-order source address set-up register (0x48224)

S1ADRL[15:0]: Ch. 1 source address [15:0] (D[F:0]) / Ch. 1 low-order source address set-up register (0x48234)

S2ADRL[15:0]: Ch. 2 source address [15:0] (D[F:0]) / Ch. 2 low-order source address set-up register (0x48244)

S3ADRL[15:0]: Ch. 3 source address [15:0] (D[F:0]) / Ch. 3 low-order source address set-up register (0x48254)

S0ADRH[11:0]: Ch. 0 source address [27:16] (D[B:0]) / Ch. 0 high-order source address set-up register (0x48226)

S1ADRH[11:0]: Ch. 1 source address [27:16] (D[B:0]) / Ch. 1 high-order source address set-up register (0x48236)

S2ADRH[11:0]: Ch. 2 source address [27:16] (D[B:0]) / Ch. 2 high-order source address set-up register (0x48246)

S3ADRH[11:0]: Ch. 3 source address [27:16] (D[B:0]) / Ch. 3 high-order source address set-up register (0x48256)

D0ADRL[15:0]: Ch. 0 destination address [15:0] (D[F:0]) / Ch. 0 low-order destination address set-up register (0x48228)  
 D1ADRL[15:0]: Ch. 1 destination address [15:0] (D[F:0]) / Ch. 1 low-order destination address set-up register (0x48238)  
 D2ADRL[15:0]: Ch. 2 destination address [15:0] (D[F:0]) / Ch. 2 low-order destination address set-up register (0x48248)  
 D3ADRL[15:0]: Ch. 3 destination address [15:0] (D[F:0]) / Ch. 3 low-order destination address set-up register (0x48258)  
 D0ADRH[11:0]: Ch. 0 destination address [27:16] (D[B:0]) / Ch. 0 high-order destination address set-up register (0x4822A)  
 D1ADRH[11:0]: Ch. 1 destination address [27:16] (D[B:0]) / Ch. 1 high-order destination address set-up register (0x4823A)  
 D2ADRH[11:0]: Ch. 2 destination address [27:16] (D[B:0]) / Ch. 2 high-order destination address set-up register (0x4824A)  
 D3ADRH[11:0]: Ch. 3 destination address [27:16] (D[B:0]) / Ch. 3 high-order destination address set-up register (0x4825A)

### Address increment/decrement control

The source and/or destination addresses can be incremented or decremented when one data transfer is completed. The SxIN[1:0] bits (for source address) and DxIN[1:0] bits (for destination address) are used to set this function.

S0IN[1:0]: Ch. 0 source address control (D[D:C]) / Ch. 0 high-order source address set-up register (0x48226)

S1IN[1:0]: Ch. 1 source address control (D[D:C]) / Ch. 1 high-order source address set-up register (0x48236)

S2IN[1:0]: Ch. 2 source address control (D[D:C]) / Ch. 2 high-order source address set-up register (0x48246)

S3IN[1:0]: Ch. 3 source address control (D[D:C]) / Ch. 3 high-order source address set-up register (0x48256)

D0IN[1:0]: Ch. 0 destination address control (D[D:C]) / Ch. 0 high-order destination address set-up register (0x4822A)

D1IN[1:0]: Ch. 1 destination address control (D[D:C]) / Ch. 1 high-order destination address set-up register (0x4823A)

D2IN[1:0]: Ch. 2 destination address control (D[D:C]) / Ch. 2 high-order destination address set-up register (0x4824A)

D3IN[1:0]: Ch. 3 destination address control (D[D:C]) / Ch. 3 high-order destination address set-up register (0x4825A)

#### SxIN/DxIN = "00": address fixed (default)

The address is not changed by a data transfer performed. Even when transferring multiple data, the transfer data is always read/write from/to the same address.

#### SxIN/DxIN = "01": address decremented without initialization

The address is decremented by an amount equal to the data size set by DATSIZE<sub>x</sub> when one data transfer is completed. The address that has been decremented during transfer does not return to the initial value.

#### SxIN/DxIN = "10": address incremented with initialization

If this function is selected in single and successive transfer modes, the address is incremented by an amount equal to the data size set by DATSIZE<sub>x</sub> when one data transfer is completed. The address that has been incremented during transfer does not return to the initial value.

In block transfer mode too, the address is incremented when one data unit is transferred. However, the address that has been incremented during a block transfer recycles returns to the initial value when the block transfer is completed.

#### SxIN/DxIN = "11": address incremented without initialization

The address is incremented by an amount equal to the data size set by DATSIZE<sub>x</sub> when one data transfer is completed. The address that has been incremented during transfer does not return to the initial value.

## Setting the Registers in Single-Address Mode

Make sure that the HSDMA channel is disabled ( $HSx\_EN = "0"$ ) before setting the control information.

### Address mode

The address mode select bit  $DUALMx$  should be set to "0" (single-address mode). This bit is set to "0" at initial reset.

### Transfer mode

A transfer mode should be set using the  $DxMOD[1:0]$  bits.

- Single transfer mode ( $DxMOD = "00"$ , default)
- Successive transfer mode ( $DxMOD = "01"$ )
- Block transfer mode ( $DxMOD = "10"$ )

Refer to the explanation in "Setting the Registers in Dual-Address Mode".

### Direction of transfer

The direction of data transfer should be set using  $DxDIR$ .

D0DIR: Ch. 0 transfer direction control (DE) / HSDMA Ch. 0 control register (0x48222)

D1DIR: Ch. 1 transfer direction control (DE) / HSDMA Ch. 1 control register (0x48232)

D2DIR: Ch. 2 transfer direction control (DE) / HSDMA Ch. 2 control register (0x48242)

D3DIR: Ch. 3 transfer direction control (DE) / HSDMA Ch. 3 control register (0x48252)

Memory write operations (data transfer from I/O device to memory) are specified by writing "1" and memory read operations (data transfer from memory to I/O device) are specified by writing "0".

### Transfer data size

The  $DATSIZEx$  bit is used to set the unit size of data to be transferred.

A half-word size (16 bits) is assumed if this bit is "1" and a byte size (8 bits) is assumed if this bit is "0" (default).

### Block length

When using block transfer mode ( $DxMOD = "10"$ ), the data block length (in units of  $DATSIZEx$ ) should be set using the  $BLKLENx[7:0]$  bits.

In single transfer and successive transfer modes,  $BLKLENx[7:0]$  is used as the bits 7–0 of the transfer counter.

**Note:** When performing data transfer in block transfer mode, the block size must not be set to "0".

### Transfer counter

#### Block transfer mode

In block transfer mode, up to 16 bits of transfer count can be specified using  $TCx\_L[7:0]$  and  $TCx\_H[7:0]$ .

#### Single transfer and successive transfer modes

In single transfer and successive transfer modes, up to 24 bits of transfer count can be specified using  $BLKLENx[7:0]$ ,  $TCx\_L[7:0]$  and  $TCx\_H[7:0]$ .



## Memory address

In single-address mode, SxADRL[15:0] and SxADRH[11:0] are used to specify a memory address.

S0ADRL[15:0]: Ch. 0 memory address [15:0] (D[F:0]) / Ch. 0 low-order source address set-up register (0x48224)  
 S0ADRH[11:0]: Ch. 0 memory address [27:16] (D[B:0]) / Ch. 0 high-order source address set-up register (0x48226)  
 S1ADRL[15:0]: Ch. 1 memory address [15:0] (D[F:0]) / Ch. 1 low-order source address set-up register (0x48234)  
 S1ADRH[11:0]: Ch. 1 memory address [27:16] (D[B:0]) / Ch. 1 high-order source address set-up register (0x48236)  
 S2ADRL[15:0]: Ch. 2 memory address [15:0] (D[F:0]) / Ch. 2 low-order source address set-up register (0x48244)  
 S2ADRH[11:0]: Ch. 2 memory address [27:16] (D[B:0]) / Ch. 2 high-order source address set-up register (0x48246)  
 S3ADRL[15:0]: Ch. 3 memory address [15:0] (D[F:0]) / Ch. 3 low-order source address set-up register (0x48254)  
 S3ADRH[11:0]: Ch. 3 memory address [27:16] (D[B:0]) / Ch. 3 high-order source address set-up register (0x48256)

In single-address mode, data transfer is performed between the memory connected to the system interface and an external I/O device. The I/O device is accessed directly by the #DMAACKx signal, so it is unnecessary to specify an address. DxADRL[15:0] and DxADRH[11:0] are not used in single-address mode.

## Address increment/decrement control

The memory addresses can be incremented or decremented when one data transfer is completed. SxIN[1:0] is used to set this function.

S0IN[1:0]: Ch. 0 memory address control (D[D:C]) / Ch. 0 high-order source address set-up register (0x48226)  
 S1IN[1:0]: Ch. 1 memory address control (D[D:C]) / Ch. 1 high-order source address set-up register (0x48236)  
 S2IN[1:0]: Ch. 2 memory address control (D[D:C]) / Ch. 2 high-order source address set-up register (0x48246)  
 S3IN[1:0]: Ch. 3 memory address control (D[D:C]) / Ch. 3 high-order source address set-up register (0x48256)

SxIN = "00": address fixed (default)

SxIN = "01": address decremented without initialization

SxIN = "10": address incremented with initialization

SxIN = "11": address incremented without initialization

Refer to the explanation in "Setting the Registers in Dual-Address Mode".

DxIN[1:0] is not used in single-address mode.

## ***Enabling/Disabling DMA Transfer***

---

The HSDMA transfer is enabled by writing "1" to the enable bit HSx\_EN.

HS0\_EN: Ch. 0 enable (D0) / Ch. 0 enable register (0x4822C)

HS1\_EN: Ch. 1 enable (D0) / Ch. 1 enable register (0x4823C)

HS2\_EN: Ch. 2 enable (D0) / Ch. 2 enable register (0x4824C)

HS3\_EN: Ch. 3 enable (D0) / Ch. 3 enable register (0x4825C)

However, the control information must always be set correctly before enabling a DMA transfer.

Note that the control information cannot be set when HSx\_EN = "1".

When HSx\_EN is set to "0", HSDMA requests are no longer accepted.

When a DMA transfer is completed (transfer counter = 0), HSx\_EN is reset to "0" to disable the following trigger inputs.

## Trigger Factor

A HSDMA trigger factor can be selected from among 13 types using the HSDMA trigger set-up register for each channel. This function is supported by the interrupt controller.

HSD0S[3:0]: Ch. 0 trigger set-up (D[3:0]) / HSDMA Ch. 0/1 trigger set-up register (0x40298)

HSD1S[3:0]: Ch. 1 trigger set-up (D[7:4]) / HSDMA Ch. 0/1 trigger set-up register (0x40298)

HSD2S[3:0]: Ch. 2 trigger set-up (D[3:0]) / HSDMA Ch. 2/3 trigger set-up register (0x40299)

HSD3S[3:0]: Ch. 3 trigger set-up (D[7:4]) / HSDMA Ch. 2/3 trigger set-up register (0x40299)

Table V.2.2 shows the setting value and the corresponding trigger factor.

Table V.2.2 HSDMA Trigger Factor

Value	Ch.0 trigger factor	Ch.1 trigger factor	Ch.2 trigger factor	Ch.3 trigger factor
0000	Software trigger	Software trigger	Software trigger	Software trigger
0001	K50 port input (falling edge)	K51 port input (falling edge)	K53 port input (falling edge)	K54 port input (falling edge)
0010	K50 port input (rising edge)	K51 port input (rising edge)	K53 port input (rising edge)	K54 port input (rising edge)
0011	Port 0 input	Port 1 input	Port 2 input	Port 3 input
0100	Port 4 input	Port 5 input	Port 6 input	Port 7 input
0101	8-bit timer 0 underflow	8-bit timer 1 underflow	8-bit timer 2 underflow	8-bit timer 3 underflow
0110	16-bit timer 0 compare B	16-bit timer 1 compare B	16-bit timer 2 compare B	16-bit timer 3 compare B
0111	16-bit timer 0 compare A	16-bit timer 1 compare A	16-bit timer 2 compare A	16-bit timer 3 compare A
1000	16-bit timer 4 compare B	16-bit timer 5 compare B	16-bit timer 4 compare B	16-bit timer 5 compare B
1001	16-bit timer 4 compare A	16-bit timer 5 compare A	16-bit timer 4 compare A	16-bit timer 5 compare A
1010	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full
1011	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty
1100	A/D conversion completion	A/D conversion completion	A/D conversion completion	A/D conversion completion
1101	reserved	FIFO serial I/F Ch.0 Rx buffer full	reserved	FIFO serial I/F Ch.0 Rx buffer full
1110	reserved	FIFO serial I/F Ch.0 Tx buffer empty	reserved	FIFO serial I/F Ch.0 Tx buffer empty

By selecting an interrupt factor with the HSDMA trigger set-up register, the HSDMA channel is invoked when the selected interrupt factor occurs. The interrupt control bits (interrupt factor flag, interrupt enable register, IDMA request register, interrupt priority register) do not affect this invocation. The interrupt factor that invokes HSDMA sets the interrupt factor flag, and HSDMA does not reset the flag. Consequently, when the DMA transfer is completed (even if the transfer counter is not 0), an interrupt request to the CPU will be generated if the interrupt has been enabled. To generate an interrupt only when the transfer counter reaches 0, disable the interrupt by the interrupt factor that invokes HSDMA and use the HSDMA transfer completion interrupt.

When software trigger is selected, the HSDMA channel can be invoked by writing "1" to the HSTx bit.

HST0: Ch. 0 software trigger (D0) / HSDMA software trigger register (0x4029A)

HST1: Ch. 1 software trigger (D1) / HSDMA software trigger register (0x4029A)

HST2: Ch. 2 software trigger (D2) / HSDMA software trigger register (0x4029A)

HST3: Ch. 3 software trigger (D3) / HSDMA software trigger register (0x4029A)

When the selected trigger factor occurs, the trigger flag is set to "1" to invoke the HSDMA channel.

The HSDMA starts a DMA transfer if it has been enabled and the trigger flag is cleared by the hardware at the same time. This makes it possible to queue the HSDMA triggers that have been generated.

The trigger flag can be read and cleared using the HSx\_TF bit.

HS0\_TF: Ch. 0 trigger flag status/clear (D0) / Ch. 0 trigger flag register (0x4822E)

HS1\_TF: Ch. 1 trigger flag status/clear (D0) / Ch. 1 trigger flag register (0x4823E)

HS2\_TF: Ch. 2 trigger flag status/clear (D0) / Ch. 2 trigger flag register (0x4824E)

HS3\_TF: Ch. 3 trigger flag status/clear (D0) / Ch. 3 trigger flag register (0x4825E)

By writing "1" to this bit, the set trigger flag can be cleared if the DMA transfer has not been started.

When this bit is read, "1" indicates that the flag is set and "0" indicates that the flag is cleared.

## Operation of HSDMA

An HSDMA channel starts data transfer by the selected trigger factor.

Make sure that transfer conditions and a trigger factor are set and the HSDMA channel is enabled before starting a DMA transfer.

### Operation in Dual-Address Mode

In dual-address mode, both the source and destination addresses are accessed according to the bus condition set by the BCU.

HSDMA has three transfer modes, in each of which data transfer operates differently. The following describes the operation of HSDMA in each transfer mode.

#### Single transfer mode

The channel for which DxMOD in control information is set to "00" operates in single transfer mode. In this mode, a transfer operation invoked by one trigger is completed after transferring one data unit of the size set by DATSIZE<sub>x</sub>. If a data transfer needs to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

The operation of HSDMA in single transfer mode is shown by the flow chart in Figure V.2.3.

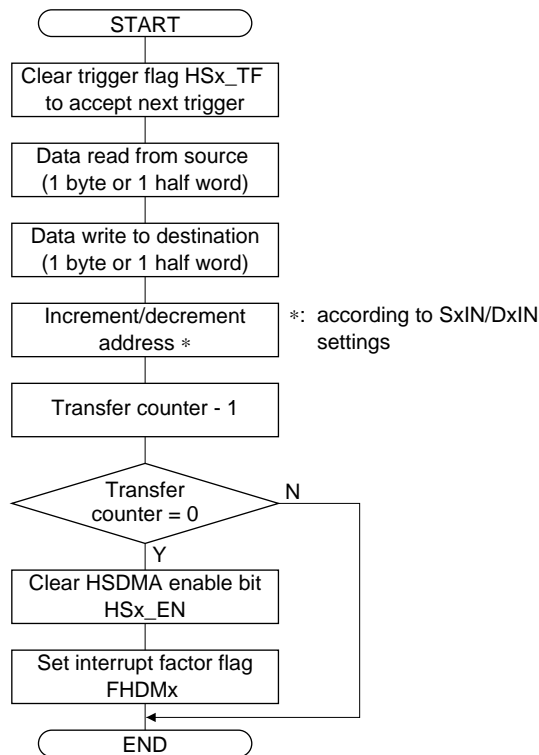


Figure V.2.3 Operation Flow in Single Transfer Mode

- (1) When a trigger is accepted, the trigger flag HS<sub>x</sub>\_TF is cleared and then data of the size set in the control information is read from the source address.
- (2) The read data is written to the destination address.
- (3) The addresses are incremented or decremented according to the S<sub>x</sub>IN/D<sub>x</sub>IN settings.
- (4) The transfer counter is decremented.
- (5) The HSDMA enable bit HS<sub>x</sub>\_EN is cleared and HSDMA interrupt factor flag in ITC is set when the transfer counter reaches 0 (when DINTEN<sub>x</sub> = "1").

## Successive transfer mode

The channel for which D<sub>x</sub>MOD in control information is set to "01" operates in successive transfer mode. In this mode, a data transfer is performed by one trigger a number of times as set by the transfer counter. The transfer counter is decremented to "0" by one transfer executed.

The operation of HSDMA in successive transfer mode is shown by the flow chart in Figure V.2.4.

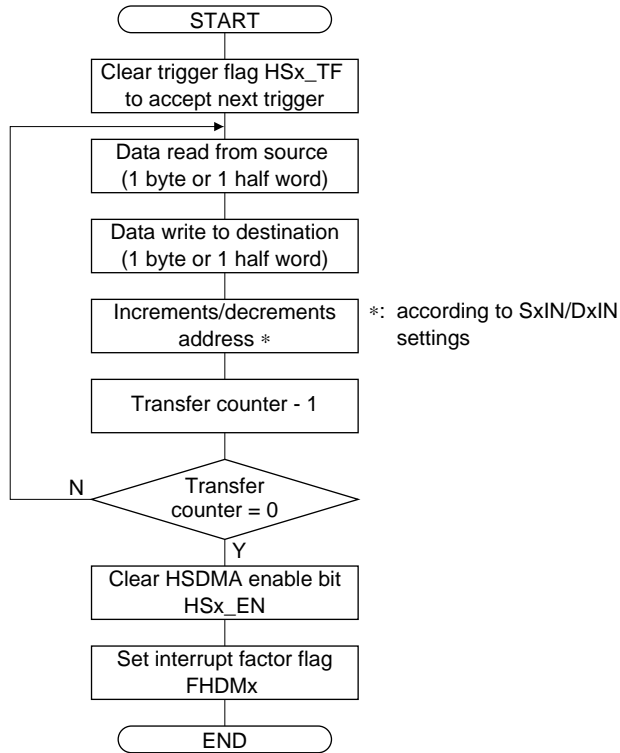


Figure V.2.4 Operation Flow in Successive Transfer Mode

- (1) When a trigger is accepted, the trigger flag HS<sub>x</sub>\_TF is cleared and then data of the size set in the control information is read from the source address.
- (2) The read data is written to the destination address.
- (3) The addresses are incremented or decremented according to the S<sub>x</sub>IN/D<sub>x</sub>IN settings.
- (4) The transfer counter is decremented.
- (5) Steps (1) to (4) are repeated until the transfer counter reaches 0.
- (6) The HSDMA enable bit HS<sub>x</sub>\_EN is cleared and HSDMA interrupt factor flag in ITC is set when the transfer counter reaches 0 (when DINTEN<sub>x</sub> = "1").

## Block transfer mode

The channel for which DxMOD in control information is set to "10" operates in block transfer mode. In this mode, a transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKLENx. If a block transfer needs to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

The operation of HSDMA in block transfer mode is shown by the flow chart in Figure V.2.5.

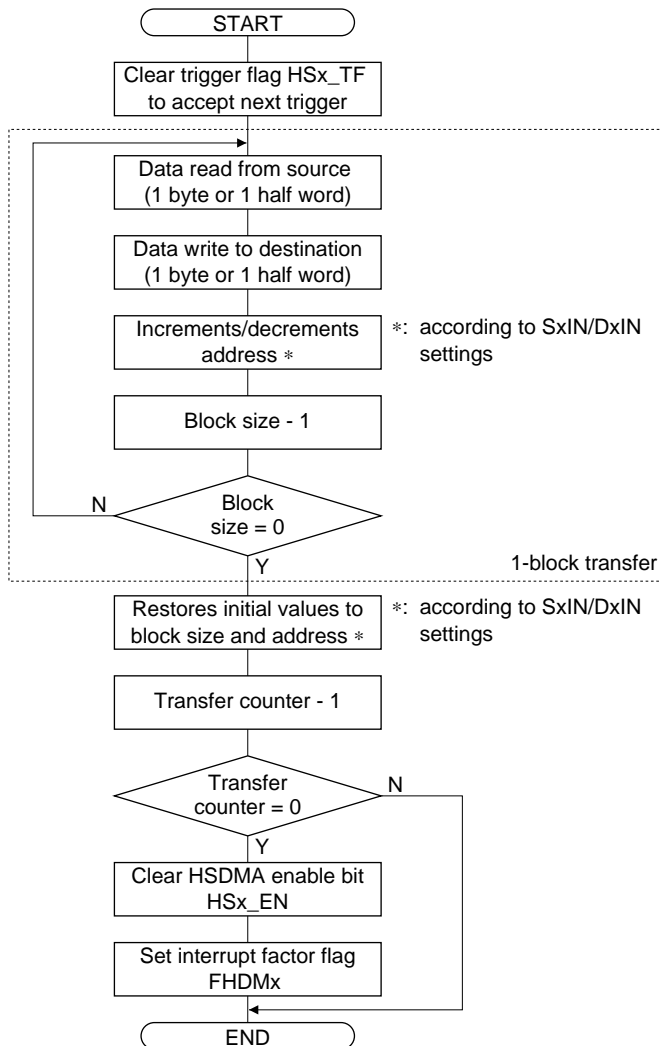


Figure V.2.5 Operation Flow in Block Transfer Mode

- (1) When a trigger is accepted, the trigger flag HSx\_TF is cleared and then data of the size set in the control information is read from the source address.
- (2) The read data is written to the destination address.
- (3) The address is incremented or decremented and BLKLENx is decremented.
- (4) Steps (1) to (3) are repeated until BLKLEN reaches 0.
- (5) If SxIN or DxIN is "10", the address is recycled to the initial value.
- (6) The transfer counter is decremented.
- (7) The HSDMA enable bit HSx\_EN is cleared and HSDMA interrupt factor flag in ITC is set when the transfer counter reaches 0 (when DINTENx = "1").

## Operation in Single-Address Mode

The operation of each transfer mode is almost the same as that of dual-address mode (see the previous section). However, data read/write operation is performed simultaneously in single-address mode. The following explains the data transfer operation different from dual-address mode.

### #DMAACKx signal output and bus operation

When the HSDMA circuit accepts the DMA request, it outputs a low-level pulse from the #DMAACKx pin and starts bus operation for the memory at the same time.

The contents of this bus operation are as follows:

- **Data transfer from I/O device to memory**

The address that has been set in the memory address register is output to the address bus.

A write operation is performed under the interface conditions set on the area to which the memory at the destination of transfer belongs. The data bus is left floating.

The external I/O device outputs the transfer data onto the data bus using the #DMAACKx signal as the read signal. The memory takes in this data using the write signal.

- **Data transfer from memory to an I/O device**

The address that has been set in the memory address register is output to the address bus.

A read operation is performed under the interface conditions set on the area to which the memory at the source of transfer belongs.

The memory outputs the transfer data onto the data bus using the read signal.

The external I/O device takes in the data from the data bus using the #DMAACKx signal as the write signal.

If the transfer data size is 16 bits and the I/O device is an 8-bit device, two bus operations are performed. Otherwise, transfer is completed in one bus operation.

### #DMAENDx signal output

When the transfer counter reaches 0, the end-of-transfer signal is output from the #DMAENDx pin indicating that a specified number of transfers has been completed. At the same time, the interrupt factor for the completion of HSDMA is generated.

## Timing Chart

### Dual-address mode

#### SRAM

Example: When 2 (RD)/1 (WR) wait cycles are inserted

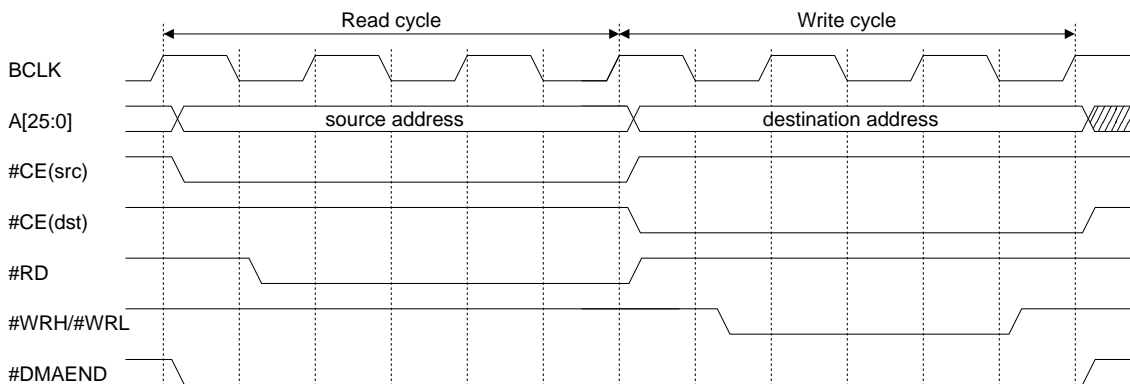


Figure V.2.6 #DMAEND Signal Output Timing (SRAM)

### Single-address mode

#### (1) SRAM

Example: When 2 (RD)/1 (WR) wait cycles are inserted

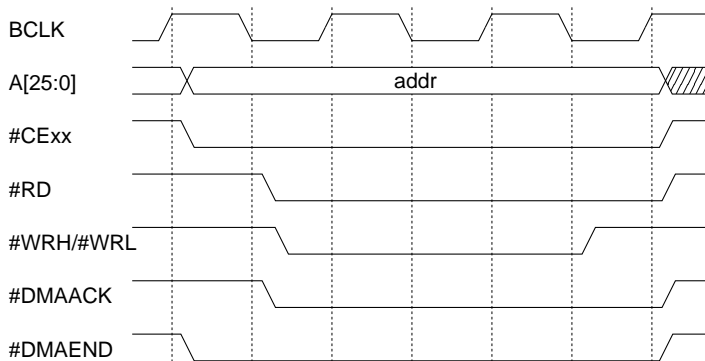


Figure V.2.7 #DMAACK/#DMAEND Signal Output Timing (SRAM)

#### (2) Burst ROM

Example: When 4-consecutive-burst and 2-wait cycles are set during the first access

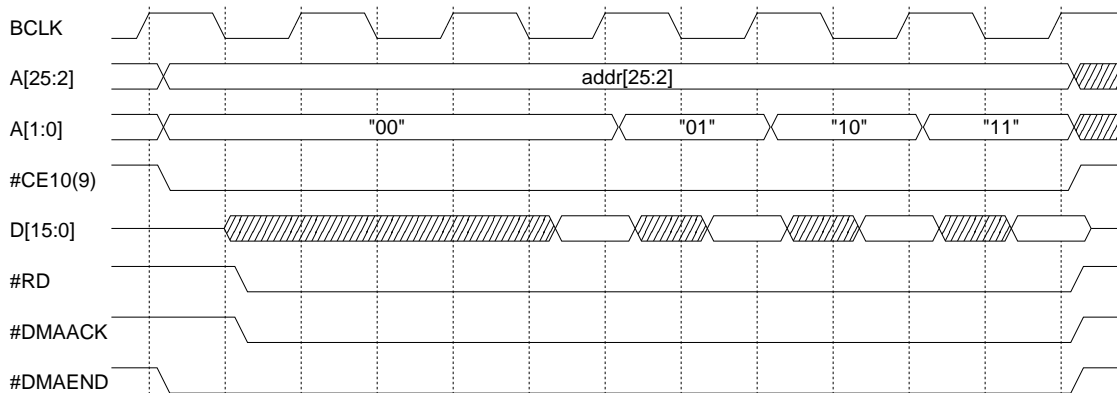


Figure V.2.8 #DMAACK/#DMAEND Signal Output Timing (Burst ROM)



## Interrupt Function of HSDMA

The DMA controller can generate an interrupt when the transfer counter in each HSDMA channel reaches 0. Furthermore, channels 0 and 1 can invoke IDMA using their interrupt factor.

### Control registers of the interrupt controller

Table V.2.3 shows the control registers of the interrupt controller that are provided for each channel.

Table V.2.3 Control Registers of Interrupt Controller

Channel	Interrupt factor flag	Interrupt enable register	Interrupt priority register
Ch. 0	FHDM0(D0/0x40281)	EHDM0(D0/0x40271)	PHSD0L[2:0](D[2:0]/0x40263)
Ch. 1	FHDM1(D1/0x40281)	EHDM1(D1/0x40271)	PHSD1L[2:0](D[6:4]/0x40263)
Ch. 2	FHDM2(D2/0x40281)	EHDM2(D2/0x40271)	PHSD2L[2:0](D[2:0]/0x40264)
Ch. 3	FHDM3(D3/0x40281)	EHDM3(D3/0x40271)	PHSD3L[2:0](D[6:4]/0x40264)

The HSDMA controller sets the HSDMA interrupt factor flag to "1" when the transfer counter reaches 0 after completing a series of HSDMA transfers. If the corresponding bit of the interrupt enable register is set to "1" at this time, an interrupt request is generated. Interrupts can be disabled by leaving the interrupt enable register bit set to "0". The HSDMA interrupt factor flag is always set to "1" when the data transfer in each channel is completed no matter what value the interrupt enable register bit is set to. (This is true even when it is set to "0".) The interrupt priority register sets an interrupt priority level (0 to 7). An interrupt request to the CPU is accepted only when there is no other interrupt request of higher priority. Furthermore, it is only when the PSR's IE bit = "1" (interrupt enable) and the set value of IL is smaller than the HSDMA interrupt level which is set in the interrupt priority register that the CPU actually accepts a HSDMA interrupt. For details about the interrupt control register and for the device operation when an interrupt occurs, refer to "ITC (Interrupt Controller)".

### Intelligent DMA

Intelligent DMA (IDMA) can be invoked by the end-of-transfer interrupt factor of channels 0 and 1 of HSDMA. The following shows the IDMA channels set in HSDMA:

IDMA channel

Channel 0 end-of-transfer interrupt: 0x05

Channel 1 end-of-transfer interrupt: 0x06

Before IDMA can be invoked, the corresponding bits of the IDMA request and IDMA enable registers must be set to "1". Settings of transfer conditions on the IDMA side are also required.

Table V.2.4 Control Bits for IDMA Transfer

Channel	IDMA request bit	IDMA enable bit
Ch. 0	RHDM0(D4/0x40290)	DEHDM0(D4/0x40294)
Ch. 1	RHDM1(D5/0x40290)	DEHDM1(D5/0x40294)

If the IDMA request and enable bits are set to "1", IDMA is invoked through generation of an interrupt factor. No interrupt request is generated at that point. An interrupt request is generated after the DMA transfer is completed. The registers can also be set so as not to generate an interrupt, with only a DMA transfer performed. For details on IDMA transfers and interrupt control upon completion of IDMA transfer, refer to "IDMA (Intelligent DMA)".

### Trap vector

The trap vector addresses for interrupt factors in each channel are set by default as follows:

Channel 0 end-of-transfer interrupt: 0x0C00058

Channel 1 end-of-transfer interrupt: 0x0C0005C

Channel 2 end-of-transfer interrupt: 0x0C00060

Channel 3 end-of-transfer interrupt: 0x0C00064

Note that the trap table base address can be modified using the TTBR registers (0x48134 to 0x48137).

## I/O Memory of HSDMA

Table V.2.5 shows the control bits of HSDMA.

Table V.2.5 Control Bits of HSDMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
High-speed DMA Ch.0/1 interrupt priority register	0040263 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PHSD1L2	High-speed DMA Ch.1 interrupt level	0 to 7	X	R/W		
		D5	PHSD1L1	High-speed DMA Ch.1 interrupt level		X			
		D4	PHSD1L0	High-speed DMA Ch.1 interrupt level		X			
		D3	–	reserved	–	–	–	0 when being read.	
		D2	PHSD0L2	High-speed DMA Ch.0 interrupt level	0 to 7	X	R/W		
		D1	PHSD0L1	High-speed DMA Ch.0 interrupt level		X			
		D0	PHSD0L0	High-speed DMA Ch.0 interrupt level		X			
High-speed DMA Ch.2/3 interrupt priority register	0040264 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PHSD3L2	High-speed DMA Ch.3 interrupt level	0 to 7	X	R/W		
		D5	PHSD3L1	High-speed DMA Ch.3 interrupt level		X			
		D4	PHSD3L0	High-speed DMA Ch.3 interrupt level		X			
		D3	–	reserved	–	–	–	0 when being read.	
		D2	PHSD2L2	High-speed DMA Ch.2 interrupt level	0 to 7	X	R/W		
		D1	PHSD2L1	High-speed DMA Ch.2 interrupt level		X			
		D0	PHSD2L0	High-speed DMA Ch.2 interrupt level		X			
DMA interrupt enable register	0040271 (B)	D7–5	–	reserved	–	–	–	0 when being read.	
		D4	EIDMA	IDMA	1 Enabled	0 Disabled	0	R/W	
		D3	EHDM3	High-speed DMA Ch.3			0	R/W	
		D2	EHDM2	High-speed DMA Ch.2			0	R/W	
		D1	EHDM1	High-speed DMA Ch.1			0	R/W	
		D0	EHDM0	High-speed DMA Ch.0			0	R/W	
DMA interrupt factor flag register	0040281 (B)	D7–5	–	reserved	–	–	–	0 when being read.	
		D4	FIDMA	IDMA	1 Factor is generated	0 No factor is generated	X	R/W	
		D3	FHDM3	High-speed DMA Ch.3			X	R/W	
		D2	FHDM2	High-speed DMA Ch.2			X	R/W	
		D1	FHDM1	High-speed DMA Ch.1			X	R/W	
		D0	FHDM0	High-speed DMA Ch.0			X	R/W	
Port input 0–3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA request register	0040290 (B)	D7	R16TC0	16-bit timer 0 comparison A	1 IDMA request	0 Interrupt request	0	R/W	
		D6	R16TU0	16-bit timer 0 comparison B			0	R/W	
		D5	RHDM1	High-speed DMA Ch.1			0	R/W	
		D4	RHDM0	High-speed DMA Ch.0			0	R/W	
		D3	RP3	Port input 3			0	R/W	
		D2	RP2	Port input 2			0	R/W	
		D1	RP1	Port input 1			0	R/W	
		D0	RP0	Port input 0			0	R/W	
Port input 0–3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA enable register	0040294 (B)	D7	DE16TC0	16-bit timer 0 comparison A	1 IDMA enabled	0 IDMA disabled	0	R/W	
		D6	DE16TU0	16-bit timer 0 comparison B			0	R/W	
		D5	DEHDM1	High-speed DMA Ch.1			0	R/W	
		D4	DEHDM0	High-speed DMA Ch.0			0	R/W	
		D3	DEP3	Port input 3			0	R/W	
		D2	DEP2	Port input 2			0	R/W	
		D1	DEP1	Port input 1			0	R/W	
		D0	DEP0	Port input 0			0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
High-speed DMA Ch.0/1 trigger set-up register	0040298 (B)	D7	HSD1S3	High-speed DMA Ch.1 trigger set-up	0	Software trigger	0	R/W			
		D6	HSD1S2		1	K51 input (falling edge)	0				
		D5	HSD1S1		2	K51 input (rising edge)	0				
		D4	HSD1S0		3	Port 1 input	0				
					4	Port 5 input					
					5	8-bit timer Ch.1 underflow					
					6	16-bit timer Ch.1 compare B					
					7	16-bit timer Ch.1 compare A					
					8	16-bit timer Ch.5 compare B					
					9	16-bit timer Ch.5 compare A					
					A	SI/F Ch.1 Rx buffer full					
					B	SI/F Ch.1 Tx buffer empty					
					C	A/D conversion completion					
					D	FSI/F Ch.0 Rx buffer full					
		E	FSI/F Ch.0 Tx buffer empty								
High-speed DMA Ch.2/3 trigger set-up register	0040299 (B)	D7	HSD3S3	High-speed DMA Ch.3 trigger set-up	0	Software trigger	0	R/W			
		D6	HSD3S2		1	K54 input (falling edge)	0				
		D5	HSD3S1		2	K54 input (rising edge)	0				
		D4	HSD3S0		3	Port 3 input	0				
					4	Port 7 input					
					5	8-bit timer Ch.3 underflow					
					6	16-bit timer Ch.3 compare B					
					7	16-bit timer Ch.3 compare A					
					8	16-bit timer Ch.5 compare B					
					9	16-bit timer Ch.5 compare A					
					A	SI/F Ch.1 Rx buffer full					
					B	SI/F Ch.1 Tx buffer empty					
					C	A/D conversion completion					
					D	FSI/F Ch.0 Rx buffer full					
		E	FSI/F Ch.0 Tx buffer empty								
High-speed DMA software trigger register	004029A (B)	D7-4	–	reserved	–	–	–	–	0 when being read.		
		D3	HST3	HSDMA Ch.3 software trigger	1	Trigger	0	Invalid		0	W
		D2	HST2	HSDMA Ch.2 software trigger						0	W
		D1	HST1	HSDMA Ch.1 software trigger						0	W
		D0	HST0	HSDMA Ch.0 software trigger						0	W

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
K5 function select register	00402C0 (B)	D7-5	–	reserved		–	–	0 when being read.		
		D4	CFK54	K54 function selection	1 #DMAREQ3	0 K54	0	R/W		
		D3	CFK53	K53 function selection	1 #DMAREQ2	0 K53	0	R/W		
		D2	CFK52	K52 function selection	1 #ADTRG	0 K52	0	R/W		
		D1	CFK51	K51 function selection	1 #DMAREQ1	0 K51	0	R/W		
		D0	CFK50	K50 function selection	1 #DMAREQ0	0 K50	0	R/W		
P1 function select register	00402D4 (B)	D7	–	reserved		–	–	0 when being read.		
		D6	CFP16	P16 function selection	1 EXCL5 #DMAEND1	0 P16	0	R/W	Extended functions (0x402D7)	
		D5	CFP15	P15 function selection	1 EXCL4 #DMAEND0	0 P15	0	R/W		
		D4	CFP14	P14 function selection	1 FOSC1	0 P14	0	R/W	Extended functions (0x402DF)	
		D3	CFP13	P13 function selection	1 EXCL3 T8UF3	0 P13	0	R/W		
		D2	CFP12	P12 function selection	1 EXCL2 T8UF2	0 P12	0	R/W		
		D1	CFP11	P11 function selection	1 EXCL1 T8UF1	0 P11	0	R/W		
		D0	CFP10	P10 function selection	1 EXCL0 T8UF0	0 P10	0	R/W		
P1 I/O control register	00402D6 (B)	D7	–	reserved		–	–	0 when being read.		
		D6	IOC16	P16 I/O control	1 Output	0 Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)	
		D5	IOC15	P15 I/O control			0	R/W		
		D4	IOC14	P14 I/O control			0	R/W		
		D3	IOC13	P13 I/O control			0	R/W		
		D2	IOC12	P12 I/O control			0	R/W		
		D1	IOC11	P11 I/O control			0	R/W		
		D0	IOC10	P10 I/O control			0	R/W		
P3 function select register	00402DC (B)	D7-6	–	reserved				–		–
		D5	CFP35	P35 function selection	1 #BUSACK	0 P35	0	R/W	Ext. func.(0x300047)	
		D4	CFP34	P34 function selection	1 #BUSREQ #CE6	0 P34	0	R/W		
		D3	CFP33	P33 function selection	1 #DMAACK1	0 P33	0	R/W	Ext. func.(0x402D7)	
		D2	CFP32	P32 function selection	1 #DMAACK0	0 P32	0	R/W	Ext. func.(0x402DF)	
		D1	CFP31	P31 function selection	1 #BUSGET	0 P31	0	R/W		
		D0	CFP30	P30 function selection	1 #WAIT #CE4/#CE5	0 P30	0	R/W		
Port function extension register	00402DF (B)	D7	CFEX7	P07 port extended function	1 #DMAEND3	0 P07, etc.	0	R/W		
		D6	CFEX6	P06 port extended function	1 #DMAACK3	0 P06, etc.	0	R/W		
		D5	CFEX5	P05 port extended function	1 #DMAEND2	0 P05, etc.	0	R/W		
		D4	CFEX4	P04 port extended function	1 #DMAACK2	0 P04, etc.	0	R/W		
		D3	CFEX3	P31 port extended function	1 #GARD	0 P31, etc.	0	R/W		
		D2	CFEX2	P21 port extended function	1 #GAAS	0 P21, etc.	0	R/W		
		D1	CFEX1	P10, P11, P13 port extended function	1 DST0 DST1 DPCO	0 P10, etc. P11, etc. P13, etc.	1	R/W		
		D0	CFEX0	P12, P14 port extended function	1 DST2 DCLK	0 P12, etc. P14, etc.	1	R/W		
High-speed DMA Ch.0 transfer counter register	0048220 (HW)	DF	TC0_L7	Ch.0 transfer counter[7:0] (block transfer mode)			X	R/W		
		DE	TC0_L6				X			
		DD	TC0_L5				X			
		DC	TC0_L4				X			
		DB	TC0_L3				X			
		DA	TC0_L2				X			
		D9	TC0_L1				X			
		D8	TC0_L0				X			
		D7	BLKLEN07	Ch.0 block length (block transfer mode)			X			
		D6	BLKLEN06				X			
		D5	BLKLEN05				X			
		D4	BLKLEN04				X			
		D3	BLKLEN03				Ch.0 transfer counter[7:0] (single/successive transfer mode)			X
		D2	BLKLEN02							X
		D1	BLKLEN01							X
		D0	BLKLEN00							X

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
High-speed DMA Ch.0 control register	0048222 (HW)	DF	DUALM0	Ch.0 address mode selection	1   Dual addr   0   Single addr	0	R/W		
		DE	D0DIR	D) Invalid	–	–	–		
				S) Ch.0 transfer direction control	1   Memory WR   0   Memory RD	0	R/W		
		DD-8	–	reserved	–	–	–	Undefined in read.	
		D7	TC0_H7	Ch.0 transfer counter[15:8]			X	R/W	
		D6	TC0_H6	(block transfer mode)			X		
		D5	TC0_H5				X		
		D4	TC0_H4	Ch.0 transfer counter[23:16]			X		
		D3	TC0_H3	(single/successive transfer mode)			X		
		D2	TC0_H2				X		
D1	TC0_H1				X				
D0	TC0_H0				X				
High-speed DMA Ch.0 low-order source address set-up register	0048224 (HW)	DF	S0ADRL15	D) Ch.0 source address[15:0]			X	R/W	
		DE	S0ADRL14	S) Ch.0 memory address[15:0]			X		
		DD	S0ADRL13				X		
		DC	S0ADRL12				X		
		DB	S0ADRL11				X		
		DA	S0ADRL10				X		
		D9	S0ADRL9				X		
		D8	S0ADRL8				X		
		D7	S0ADRL7				X		
		D6	S0ADRL6				X		
		D5	S0ADRL5				X		
		D4	S0ADRL4				X		
		D3	S0ADRL3				X		
		D2	S0ADRL2				X		
D1	S0ADRL1				X				
D0	S0ADRL0				X				
High-speed DMA Ch.0 high-order source address set-up register	0048226 (HW)	DF	–	reserved	–	–	–		
		DE	DATSIZE0	Ch.0 transfer data size	1   Half word   0   Byte	0	R/W		
		DD	S0IN1	D) Ch.0 source address control	S0IN[1:0]	Inc/dec	0	R/W	
		DC	S0IN0	S) Ch.0 memory address control	1   1   Inc.(no init)		0		
					1   0   Inc.(init)				
					0   1   Dec.(no init)				
					0   0   Fixed				
		DB	S0ADRH11	D) Ch.0 source address[27:16]			X	R/W	
		DA	S0ADRH10	S) Ch.0 memory address[27:16]			X		
		D9	S0ADRH9				X		
		D8	S0ADRH8				X		
		D7	S0ADRH7				X		
		D6	S0ADRH6				X		
		D5	S0ADRH5				X		
D4	S0ADRH4				X				
D3	S0ADRH3				X				
D2	S0ADRH2				X				
D1	S0ADRH1				X				
D0	S0ADRH0				X				
High-speed DMA Ch.0 low-order destination address set-up register	0048228 (HW)	DF	D0ADRL15	D) Ch.0 destination address[15:0]			X	R/W	
		DE	D0ADRL14	S) Invalid			X		
		DD	D0ADRL13				X		
		DC	D0ADRL12				X		
		DB	D0ADRL11				X		
		DA	D0ADRL10				X		
		D9	D0ADRL9				X		
		D8	D0ADRL8				X		
		D7	D0ADRL7				X		
		D6	D0ADRL6				X		
		D5	D0ADRL5				X		
		D4	D0ADRL4				X		
		D3	D0ADRL3				X		
		D2	D0ADRL2				X		
D1	D0ADRL1				X				
D0	D0ADRL0				X				

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
High-speed DMA Ch.0 high-order destination address set-up register  Note: D) Dual address mode S) Single address mode	004822A (HW)	DF	D0MOD1	Ch.0 transfer mode	D0MOD[1:0]		Mode	0	R/W			
			DE		D0MOD0	1	1	Invalid			0	
						1	0	Block				
						0	1	Successive				
					0	0	Single					
				DD	D0IN1	D) Ch.0 destination address control S) Invalid	D0IN[1:0]		Inc/dec	0	R/W	
			D0IN0		1		1	Inc.(no init)	0			
					1		0	Inc.(init)				
					0		1	Dec.(no init)				
					0	0	Fixed					
				DB	D0ADRH11	D) Ch.0 destination address[27:16] S) Invalid			X	R/W		
				DA	D0ADRH10				X			
				D9	D0ADRH9				X			
				D8	D0ADRH8				X			
		D7	D0ADRH7				X					
		D6	D0ADRH6				X					
		D5	D0ADRH5				X					
		D4	D0ADRH4				X					
		D3	D0ADRH3				X					
		D2	D0ADRH2				X					
		D1	D0ADRH1			X						
		D0	D0ADRH0			X						
High-speed DMA Ch.0 enable register	004822C (HW)	DF-1	–	reserved		–	–	–	–	Undefined in read.		
		D0	HS0_EN	Ch.0 enable	1	Enable	0	Disable	0	R/W		
High-speed DMA Ch.0 trigger flag register	004822E (HW)	DF-1	–	reserved		–	–	–	–	Undefined in read.		
		D0	HS0_TF	Ch.0 trigger flag clear (writing) Ch.0 trigger flag status (reading)	1	Clear	0	No operation	0	R/W		
					1	Set	0	Cleared				
High-speed DMA Ch.1 transfer counter register	0048230 (HW)	DF	TC1_L7	Ch.1 transfer counter[7:0] (block transfer mode)				X	R/W			
			DE		TC1_L6			X				
			DD		TC1_L5			X				
			DC		TC1_L4	Ch.1 transfer counter[15:8] (single/successive transfer mode)					X	
			DB		TC1_L3						X	
			DA		TC1_L2						X	
			D9		TC1_L1						X	
					D8	TC1_L0					X	
			D7	BLKLEN17	Ch.1 block length (block transfer mode)				X	R/W		
			D6	BLKLEN16				X				
			D5	BLKLEN15			X					
			D4	BLKLEN14	Ch.1 transfer counter[7:0] (single/successive transfer mode)				X	R/W		
			D3	BLKLEN13				X				
			D2	BLKLEN12				X				
	D1	BLKLEN11				X						
	D0	BLKLEN10				X						
						X						
High-speed DMA Ch.1 control register  Note: D) Dual address mode S) Single address mode	0048232 (HW)	DF	DUALM1	Ch.1 address mode selection	1	Dual addr	0	Single addr	0	R/W		
		DE	D1DIR	D) Invalid S) Ch.1 transfer direction control		–	–	–	–	–		
					1	Memory WR	0	Memory RD	0	R/W		
		DD-8	–	reserved		–	–	–	–	–	Undefined in read.	
			D7	TC1_H7	Ch.1 transfer counter[15:8] (block transfer mode)				X	R/W		
			D6	TC1_H6				X				
			D5	TC1_H5			X					
			D4	TC1_H4	Ch.1 transfer counter[23:16] (single/successive transfer mode)				X	R/W		
			D3	TC1_H3				X				
			D2	TC1_H2				X				
	D1	TC1_H1				X						
	D0	TC1_H0				X						
						X						

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
High-speed DMA Ch.1 low-order source address set-up register	0048234 (HW)	DF	S1ADRL15	D) Ch.1 source address[15:0]			X	R/W		
		DE	S1ADRL14	S) Ch.1 memory address[15:0]						
		DD	S1ADRL13							
		DC	S1ADRL12							
		DB	S1ADRL11							
		DA	S1ADRL10							
		D9	S1ADRL9							
		D8	S1ADRL8							
		D7	S1ADRL7							
		D6	S1ADRL6							
		D5	S1ADRL5							
		D4	S1ADRL4							
		D3	S1ADRL3							
		D2	S1ADRL2							
		D1	S1ADRL1							
		D0	S1ADRL0							
High-speed DMA Ch.1 high-order source address set-up register	0048236 (HW)	DF	–	reserved	–			–	–	
		DE	DATSIZE1	Ch.1 transfer data size	1	Half word	0	Byte	0	R/W
		DD	S1IN1	D) Ch.1 source address control	S1IN[1:0]			Inc/dec	0	R/W
		DC	S1IN0	S) Ch.1 memory address control	1	1	Inc.(no init)		0	
					1	0	Inc.(init)			
					0	1	Dec.(no init)			
					0	0	Fixed			
		DB	S1ADRH11	D) Ch.1 source address[27:16]				X	R/W	
		DA	S1ADRH10	S) Ch.1 memory address[27:16]						
		D9	S1ADRH9							
D8	S1ADRH8									
D7	S1ADRH7									
D6	S1ADRH6									
D5	S1ADRH5									
D4	S1ADRH4									
D3	S1ADRH3									
D2	S1ADRH2									
D1	S1ADRH1									
D0	S1ADRH0									
High-speed DMA Ch.1 low-order destination address set-up register	0048238 (HW)	DF	D1ADRL15	D) Ch.1 destination address[15:0]			X	R/W		
		DE	D1ADRL14	S) Invalid						
		DD	D1ADRL13							
		DC	D1ADRL12							
		DB	D1ADRL11							
		DA	D1ADRL10							
		D9	D1ADRL9							
		D8	D1ADRL8							
		D7	D1ADRL7							
		D6	D1ADRL6							
		D5	D1ADRL5							
		D4	D1ADRL4							
		D3	D1ADRL3							
		D2	D1ADRL2							
		D1	D1ADRL1							
		D0	D1ADRL0							

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
High-speed DMA Ch.1 high-order destination address set-up register  Note: D) Dual address mode S) Single address mode	004823A (HW)	DF DE	D1MOD1	Ch.1 transfer mode	D1MOD[1:0]	Mode	0	R/W			
			Invalid			Block Successive Single					
			1 0								
			0 1								
		DD DC	D1IN1 D1IN0	D) Ch.1 destination address control S) Invalid	D1IN[1:0]	Inc/dec	0	R/W			
										1 1	Inc.(no init)
		DB DA D9 D8 D7 D6 D5 D4 D3 D2 D1 D0	D1ADRH11 D1ADRH10 D1ADRH9 D1ADRH8 D1ADRH7 D1ADRH6 D1ADRH5 D1ADRH4 D1ADRH3 D1ADRH2 D1ADRH1 D1ADRH0	D) Ch.1 destination address[27:16] S) Invalid				X		R/W	
											X
											X
											X
											X
											X
											X
X											
X											
X											
X											
High-speed DMA Ch.1 enable register	004823C (HW)	DF-1	–	reserved	–	–	–	Undefined in read.			
		D0	HS1_EN	Ch.1 enable	1   Enable	0   Disable	0	R/W			
High-speed DMA Ch.1 trigger flag register	004823E (HW)	DF-1	–	reserved	–	–	–	Undefined in read.			
		D0	HS1_TF	Ch.1 trigger flag clear (writing)	1   Clear	0   No operation	0	R/W			
				Ch.1 trigger flag status (reading)	1   Set	0   Cleared					
High-speed DMA Ch.2 transfer counter register	0048240 (HW)	DF DE DD DC DB DA D9 D8	TC2_L7	Ch.2 transfer counter[7:0] (block transfer mode)			X	R/W			
			TC2_L6								
			TC2_L5								
			TC2_L4								
			TC2_L3								
			TC2_L2								
			TC2_L1								
			TC2_L0								
		D7 D6 D5 D4 D3 D2 D1 D0	BLKLEN27 BLKLEN26 BLKLEN25 BLKLEN24 BLKLEN23 BLKLEN22 BLKLEN21 BLKLEN20	Ch.2 block length (block transfer mode)	Ch.2 transfer counter[7:0] (single/successive transfer mode)			X	R/W		
										X	
										X	
										X	
										X	
										X	
High-speed DMA Ch.2 control register	0048242 (HW)	DF	DUALM2	Ch.2 address mode selection	1   Dual addr	0   Single addr	0	R/W			
		DE	D2DIR	D) Invalid S) Ch.2 transfer direction control	–	–	–	–			
		DD-8	–	reserved	–	–	–	–	Undefined in read.		
		D7 D6 D5 D4 D3 D2 D1 D0	TC2_H7 TC2_H6 TC2_H5 TC2_H4 TC2_H3 TC2_H2 TC2_H1 TC2_H0	Ch.2 transfer counter[15:8] (block transfer mode)	Ch.2 transfer counter[23:16] (single/successive transfer mode)			X	R/W		
										X	
										X	
X											
X											



Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
High-speed DMA Ch.2 low-order source address set-up register	0048244 (HW)	DF	S2ADRL15	D) Ch.2 source address[15:0]			X	R/W		
		DE	S2ADRL14	S) Ch.2 memory address[15:0]						
		DD	S2ADRL13							
		DC	S2ADRL12							
		DB	S2ADRL11							
		DA	S2ADRL10							
		D9	S2ADRL9							
		D8	S2ADRL8							
		D7	S2ADRL7							
		D6	S2ADRL6							
		D5	S2ADRL5							
		D4	S2ADRL4							
		D3	S2ADRL3							
		D2	S2ADRL2							
		D1	S2ADRL1							
		D0	S2ADRL0							
High-speed DMA Ch.2 high-order source address set-up register	0048246 (HW)	DF	–	reserved	–		–	–		
		DE	DATSIZE2	Ch.2 transfer data size	1	Half word	0	Byte	0	R/W
		DD	S2IN1	D) Ch.2 source address control	S2IN[1:0]		Inc/dec		0	R/W
		DC	S2IN0	S) Ch.2 memory address control	1	1	Inc.(no init)		0	
					1	0	Inc.(init)			
					0	1	Dec.(no init)			
					0	0	Fixed			
		DB	S2ADRH11	D) Ch.2 source address[27:16]				X	R/W	
		DA	S2ADRH10	S) Ch.2 memory address[27:16]						
		D9	S2ADRH9							
D8	S2ADRH8									
D7	S2ADRH7									
D6	S2ADRH6									
D5	S2ADRH5									
D4	S2ADRH4									
D3	S2ADRH3									
D2	S2ADRH2									
D1	S2ADRH1									
D0	S2ADRH0									
High-speed DMA Ch.2 low-order destination address set-up register	0048248 (HW)	DF	D2ADRL15	D) Ch.2 destination address[15:0]			X	R/W		
		DE	D2ADRL14	S) Invalid						
		DD	D2ADRL13							
		DC	D2ADRL12							
		DB	D2ADRL11							
		DA	D2ADRL10							
		D9	D2ADRL9							
		D8	D2ADRL8							
		D7	D2ADRL7							
		D6	D2ADRL6							
		D5	D2ADRL5							
		D4	D2ADRL4							
		D3	D2ADRL3							
		D2	D2ADRL2							
		D1	D2ADRL1							
		D0	D2ADRL0							

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
High-speed DMA Ch.2 high-order destination address set-up register  Note: D) Dual address mode S) Single address mode	004824A (HW)	DF DE	D2MOD1	Ch.2 transfer mode	D2MOD[1:0]		Mode	0	R/W		
			D2MOD0		1	1	Invalid	0			
					1	0	Block				
					0	1	Successive				
					0	0	Single				
			DD DC	D2IN1	D) Ch.2 destination address control S) Invalid	D2IN[1:0]		Inc/dec	0	R/W	
		D2IN0		1		1	Inc.(no init)	0			
				1	0	Inc.(init)					
				0	1	Dec.(no init)					
				0	0	Fixed					
			DB DA D9 D8 D7 D6 D5 D4 D3 D2 D1 D0	D2ADRH11	D) Ch.2 destination address[27:16] S) Invalid				X	R/W	
		D2ADRH10					X				
		D2ADRH9				X					
D2ADRH8		X									
D2ADRH7		X									
D2ADRH6		X									
D2ADRH5		X									
D2ADRH4		X									
D2ADRH3		X									
D2ADRH2		X									
D2ADRH1		X									
D2ADRH0		X									
High-speed DMA Ch.2 enable register	004824C (HW)	DF-1	–	reserved		–	–	Undefined in read.			
		D0	HS2_EN	Ch.2 enable	1	Enable	0	Disable	0	R/W	
High-speed DMA Ch.2 trigger flag register	004824E (HW)	DF-1	–	reserved		–	–	Undefined in read.			
		D0	HS2_TF	Ch.2 trigger flag clear (writing)	1	Clear	0	No operation	0	R/W	
Ch.2 trigger flag status (reading)	1			Set	0	Cleared					
High-speed DMA Ch.3 transfer counter register	0048250 (HW)	DF DE DD DC DB DA D9 D8	TC3_L7	Ch.3 transfer counter[7:0] (block transfer mode)				X	R/W		
			TC3_L6			X					
			TC3_L5			X					
			TC3_L4		Ch.3 transfer counter[15:8] (single/successive transfer mode)		X				
			TC3_L3				X				
			TC3_L2				X				
			TC3_L1				X				
			TC3_L0			X					
		D7 D6 D5 D4 D3 D2 D1 D0	BLKLEN37	Ch.3 block length (block transfer mode)		X	R/W				
			BLKLEN36			X					
			BLKLEN35			X					
			BLKLEN34		Ch.3 transfer counter[7:0] (single/successive transfer mode)			X			
			BLKLEN33					X			
			BLKLEN32					X			
			BLKLEN31					X			
BLKLEN30		X									
High-speed DMA Ch.3 control register  Note: D) Dual address mode S) Single address mode	0048252 (HW)	DF DE	DUALM3	Ch.3 address mode selection	1	Dual addr	0	Single addr	0	R/W	
			D3DIR	D) Invalid S) Ch.3 transfer direction control		–	–	–	–		
						1	Memory WR	0	Memory RD	0	R/W
		DD-8	–	reserved		–	–	–	–	Undefined in read.	
		D7 D6 D5 D4 D3 D2 D1 D0	TC3_H7 TC3_H6 TC3_H5 TC3_H4 TC3_H3 TC3_H2 TC3_H1 TC3_H0	Ch.3 transfer counter[15:8] (block transfer mode)		X	R/W				
						X					
						X					
					Ch.3 transfer counter[23:16] (single/successive transfer mode)			X			
								X			
								X			
	X										

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
High-speed DMA Ch.3 low-order source address set-up register	0048254 (HW)	DF	S3ADRL15	D) Ch.3 source address[15:0]			X	R/W	
		DE	S3ADRL14	S) Ch.3 memory address[15:0]			X		
		DD	S3ADRL13				X		
		DC	S3ADRL12				X		
		DB	S3ADRL11				X		
		DA	S3ADRL10				X		
		D9	S3ADRL9				X		
		D8	S3ADRL8				X		
		D7	S3ADRL7				X		
		D6	S3ADRL6				X		
		D5	S3ADRL5				X		
		D4	S3ADRL4				X		
		D3	S3ADRL3				X		
		D2	S3ADRL2				X		
		D1	S3ADRL1				X		
		D0	S3ADRL0				X		
High-speed DMA Ch.3 high-order source address set-up register	0048256 (HW)	DF	–	reserved		–	–		
		DE	DATSIZE3	Ch.3 transfer data size	1   Half word   0   Byte	0	R/W		
		DD	S3IN1	D) Ch.3 source address control	S3IN[1:0]	Inc/dec	0	R/W	
		DC	S3IN0	S) Ch.3 memory address control		1   1	Inc.(no init)	0	
						1   0	Inc.(init)		
						0   1	Dec.(no init)		
						0   0	Fixed		
		DB	S3ADRH11	D) Ch.3 source address[27:16]			X	R/W	
		DA	S3ADRH10	S) Ch.3 memory address[27:16]			X		
		D9	S3ADRH9				X		
D8	S3ADRH8				X				
D7	S3ADRH7				X				
D6	S3ADRH6				X				
D5	S3ADRH5				X				
D4	S3ADRH4				X				
D3	S3ADRH3				X				
D2	S3ADRH2				X				
D1	S3ADRH1				X				
D0	S3ADRH0				X				
High-speed DMA Ch.3 low-order destination address set-up register	0048258 (HW)	DF	D3ADRL15	D) Ch.3 destination address[15:0]			X	R/W	
		DE	D3ADRL14	S) Invalid			X		
		DD	D3ADRL13				X		
		DC	D3ADRL12				X		
		DB	D3ADRL11				X		
		DA	D3ADRL10				X		
		D9	D3ADRL9				X		
		D8	D3ADRL8				X		
		D7	D3ADRL7				X		
		D6	D3ADRL6				X		
		D5	D3ADRL5				X		
		D4	D3ADRL4				X		
		D3	D3ADRL3				X		
		D2	D3ADRL2				X		
		D1	D3ADRL1				X		
		D0	D3ADRL0				X		

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
High-speed DMA Ch.3 high-order destination address set-up register  Note: D) Dual address mode S) Single address mode	004825A (HW)	DF	D3MOD1	Ch.3 transfer mode	D3MOD[1:0]	Mode	0	R/W			
			DE			D3MOD0				1 1	Invalid
										1 0	Block
										0 1	Successive
										0 0	Single
			DD	D3IN1	D) Ch.3 destination address control	D3IN[1:0]	Inc/dec	0	R/W		
		DC		D3IN0			1 1				Inc.(no init)
							1 0	Inc.(init)			
							0 1	Dec.(no init)			
							0 0	Fixed			
			DB DA D9 D8 D7 D6 D5 D4 D3 D2 D1 D0	D3ADRH11	D) Ch.3 destination address[27:16] S) Invalid			X	R/W		
		D3ADRH10		X							
		D3ADRH9		X							
D3ADRH8	X										
D3ADRH7	X										
D3ADRH6	X										
D3ADRH5	X										
D3ADRH4	X										
D3ADRH3	X										
D3ADRH2	X										
D3ADRH1	X										
D3ADRH0	X										
High-speed DMA Ch.3 enable register	004825C (HW)	DF-1	–	reserved	–	–	–	Undefined in read.			
		D0	HS3_EN	Ch.3 enable	1 Enable	0 Disable	0	R/W			
High-speed DMA Ch.3 trigger flag register	004825E (HW)	DF-1	–	reserved	–	–	–	Undefined in read.			
		D0	HS3_TF	Ch.3 trigger flag clear (writing)	1 Clear	0 No operation	0	R/W			
				Ch.3 trigger flag status (reading)	1 Set	0 Cleared					

**CFK51–CFK50:** K5[1:0] pin function selection (D[1:0]) / K5 function select register (0x402C0)

**CFK54–CFK53:** K5[4:3] pin function selection (D[4:3]) / K5 function select register (0x402C0)

Set the #DMAREQx pin of HSDMA.

Write "1": #DMAREQx input

Write "0": Input port

Read: Valid

CFK50, CFK51, CFK53 and CFK54 are the function select bits for K50 (#DMAREQ0), K51 (#DMAREQ1), K53 (#DMAREQ2) and K54 (#DMAREQ3), respectively. When using the #DMAREQx signal, write "1" to CFK5x to set the K5x port for inputting the signal.

If this bit is set to "0", the pin is set for an input port.

At cold start, CFK5x is set to "0" (input port). At hot start, CFK5x retains the previous status before an initial reset.

**CFP16–CFP15:** P1[6:5] pin function selection (D[6:5]) / P1 function select register (0x402D4)

Set the #DMAENDx pin of HSDMA.

Write "1": #DMAENDx output

Write "0": I/O port

Read: Valid

When using the #DMAEND0 signal, set the P15 pin for the #DMAEND0 output pin by writing "1" to CFP15.

Similarly, when using the #DMAEND1 signal, set the P16 pin for the #DMAEND1 output pin by writing "1" to CFP16. Furthermore, direct these pins for output by writing "1" to the corresponding I/O control register.

If CFP1x is set to "0", the pin is set for an I/O port.

At cold start, CFP1x is set to "0" (I/O port). At hot start, CFP1x retains the previous status before an initial reset.

**IOC16–IOC15:** P1[6:5] port I/O control (D[6:5]) / P1 I/O control register (0x402D6)

Directs P15 and P16 for input or output and indicates the I/O control signal value of the port.

**When writing data**

Write "1": Output mode

Write "0": Input mode

To use the #DMAEND0 pin (channel 0), direct the pin for output by writing "1" to IOC15; to use the #DMAEND1 pin (channel 1), direct the pin for output by writing "1" to IOC16. If these pins are set for input, the P15 and P16 pins do not function as the #DMAENDx output pins even when CFP15 and CFP16 are set to "1".

**When reading data**

Read "1": I/O control signal (output)

Read "0": I/O control signal (input)

The I/O control signal value for the port pin is read from this register. When I/O port function is selected using the CFP1x register, the value written to the IOC register is read out as is. When peripheral function is selected, the read value depends on the peripheral circuit status and may not indicate the value written to the IOC register.

At cold start, IOC1x is set to "0" (input mode). At hot start, the bit retains its state from prior to the initial reset.

**CFP33–CFP32:** P3[3:2] pin function selection (D[3:2]) / P3 function select register (0x402DC)

Set the #DMAACKx pin of HSDMA.

Write "1": #DMAACKx output

Write "0": I/O port

Read: Valid

When using the #DMAACK0 signal, set the P32 pin for the #DMAACK0 output pin by writing "1" to CFP32. Similarly, when using the #DMAACK1 signal, set the P33 pin for the #DMAACK1 output pin by writing "1" to CFP33.

If CFP3x is set to "0", the pin is set for an I/O port.

At cold start, CFP3x is set to "0" (I/O port). At hot start, CFP3x retains the previous status before an initial reset.

**CFEX7–CFEX4:** P0[7:4] pin function extension (D[7:4]) / Port function extension register (0x402DF)

Set the #DMAACKx and #DMAENDx pins of HSDMA.

Write "1": HSDMA output

Write "0": I/O-port/serial interface I/O

Read: Valid

CFEX4, CFEX5, CFEX6 and CFEX7 are the function extension bits for P04 (#DMAACK2), P05 (#DMAEND2), P06 (#DMAACK3) and P07 (#DMAEND3), respectively. When using the HSDMA signal, write "1" to CFEXx to set the P0x port for outputting the signal.

When CFEXx is set to "0", the corresponding CFP bit becomes effective.

At cold start, these bits are set to "0" (I/O-port/serial interface I/O pin). At hot start, these bits retain the previous status before an initial reset.

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

**HSD0S3–HSD0S0:** Ch. 0 trigger set-up (D[3:0]) / HSDMA Ch. 0/1 trigger set-up register (0x40298)

**HSD1S3–HSD1S0:** Ch. 1 trigger set-up (D[7:4]) / HSDMA Ch. 0/1 trigger set-up register (0x40298)

**HSD2S3–HSD2S0:** Ch. 2 trigger set-up (D[3:0]) / HSDMA Ch. 2/3 trigger set-up register (0x40299)

**HSD3S3–HSD3S0:** Ch. 3 trigger set-up (D[7:4]) / HSDMA Ch. 2/3 trigger set-up register (0x40299)

Select a trigger factor for each HSDMA channel.

Table V.2.6 HSDMA Trigger Factor

Value	Ch.0 trigger factor	Ch.1 trigger factor	Ch.2 trigger factor	Ch.3 trigger factor
0000	Software trigger	Software trigger	Software trigger	Software trigger
0001	K50 port input (falling edge)	K51 port input (falling edge)	K53 port input (falling edge)	K54 port input (falling edge)
0010	K50 port input (rising edge)	K51 port input (rising edge)	K53 port input (rising edge)	K54 port input (rising edge)
0011	Port 0 input	Port 1 input	Port 2 input	Port 3 input
0100	Port 4 input	Port 5 input	Port 6 input	Port 7 input
0101	8-bit timer 0 underflow	8-bit timer 1 underflow	8-bit timer 2 underflow	8-bit timer 3 underflow
0110	16-bit timer 0 compare B	16-bit timer 1 compare B	16-bit timer 2 compare B	16-bit timer 3 compare B
0111	16-bit timer 0 compare A	16-bit timer 1 compare A	16-bit timer 2 compare A	16-bit timer 3 compare A
1000	16-bit timer 4 compare B	16-bit timer 5 compare B	16-bit timer 4 compare B	16-bit timer 5 compare B
1001	16-bit timer 4 compare A	16-bit timer 5 compare A	16-bit timer 4 compare A	16-bit timer 5 compare A
1010	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full
1011	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty
1100	A/D conversion completion	A/D conversion completion	A/D conversion completion	A/D conversion completion
1101	reserved	FIFO serial I/F Ch.0 Rx buffer full	reserved	FIFO serial I/F Ch.0 Rx buffer full
1110	reserved	FIFO serial I/F Ch.0 Tx buffer empty	reserved	FIFO serial I/F Ch.0 Tx buffer empty

At initial reset, HSDxS is set to "0000" (software trigger).

**HST0:** Ch. 0 software trigger (D0) / HSDMA software trigger register (0x4029A)

**HST1:** Ch. 1 software trigger (D1) / HSDMA software trigger register (0x4029A)

**HST2:** Ch. 2 software trigger (D2) / HSDMA software trigger register (0x4029A)

**HST3:** Ch. 3 software trigger (D3) / HSDMA software trigger register (0x4029A)

Start a DMA transfer.

Write "1": Trigger

Write "0": Invalid

Read: Invalid

Writing "1" to HSTx generates a trigger pulse that starts a DMA transfer.

HSTx is effective only when software trigger is selected as the trigger factor of the HSDMA channel by the HSDxS bits.

At initial reset, HSTx is set to "0".

**HS0\_TF:** Ch. 0 trigger flag clear/status (D0) / HSDMA Ch. 0 trigger flag register (0x4822E)

**HS1\_TF:** Ch. 1 trigger flag clear/status (D0) / HSDMA Ch. 1 trigger flag register (0x4823E)

**HS2\_TF:** Ch. 2 trigger flag clear/status (D0) / HSDMA Ch. 2 trigger flag register (0x4824E)

**HS3\_TF:** Ch. 3 trigger flag clear/status (D0) / HSDMA Ch. 3 trigger flag register (0x4825E)

These bits are used to check and clear the trigger flag status.

Write "1": Trigger flag clear

Write "0": Invalid

Read "1": Trigger flag has been set

Read "0": Trigger flag has been cleared

The trigger flag is set when the trigger factor is input to the HSDMA channel and is cleared when the HSDMA channel starts a data transfer. By reading HSx\_TF, the flag status can be checked. Writing "1" to HSx\_TF clears the trigger flag if the DMA transfer has not been started.

At initial reset, HSx\_TF is set to "0".

---

**HS0\_EN:** Ch. 0 enable (D0) / HSDMA Ch. 0 enable register (0x4822C)

**HS1\_EN:** Ch. 1 enable (D0) / HSDMA Ch. 1 enable register (0x4823C)

**HS2\_EN:** Ch. 2 enable (D0) / HSDMA Ch. 2 enable register (0x4824C)

**HS3\_EN:** Ch. 3 enable (D0) / HSDMA Ch. 3 enable register (0x4825C)

---

Enable a DMA transfer.

Write "1": Enabled

Write "0": Disabled

Read: Valid

DMA transfer is enabled by writing "1" to this bit.

HSDMA is placed in a state ready to accept a DMA request from the #DMAREQx pin or by the selected trigger factor.

DMA transfer is disabled by writing "0" to this bit.

When DMA transfers are completed (transfer counter = 0), HSx\_EN is cleared by the hardware.

Be sure to disable DMA transfers (HSx\_EN = "0") before setting the transfer condition.

At initial reset, HSx\_EN is set to "0" (disabled).

---

**DUALM0:** Ch. 0 address mode selection (DF) / HSDMA Ch. 0 control register (0x48222)

**DUALM1:** Ch. 1 address mode selection (DF) / HSDMA Ch. 1 control register (0x48232)

**DUALM2:** Ch. 2 address mode selection (DF) / HSDMA Ch. 2 control register (0x48242)

**DUALM3:** Ch. 3 address mode selection (DF) / HSDMA Ch. 3 control register (0x48252)

---

Select an address mode.

Write "1": Dual-address mode

Write "0": Single-address mode

Read: Valid

When "1" is written to DUALMx, the HSDMA channel enters dual-address mode that allows specification of source and destination addresses. When "0" is written, the HSDMA channel enters single-address mode for high-speed data transfer between the external memory and an I/O device.

At initial reset, DUALMx is set to "0" (single-address mode).

---

**D0DIR:** Ch. 0 transfer direction control (DE) / HSDMA Ch.0 control register (0x48222)

**D1DIR:** Ch. 1 transfer direction control (DE) / HSDMA Ch.1 control register (0x48232)

**D2DIR:** Ch. 2 transfer direction control (DE) / HSDMA Ch.2 control register (0x48242)

**D3DIR:** Ch. 3 transfer direction control (DE) / HSDMA Ch.3 control register (0x48252)

---

Control the direction of data transfer in single-address mode.

Write "1": Memory write (I/O to memory)

Write "0": Memory read (memory to I/O)

Read: Valid

Data transfer from an external I/O device to external memory is performed by writing "1" to DxDIR. Data transfer from external memory to an external I/O is performed by writing "0".

At initial reset, DxDIR is set to "0" (memory to I/O).

This bit is effective only in single-address mode.

V

HSDMA

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

**D0MOD1–D0MOD0:** Ch. 0 transfer mode (D[F:E]) / Ch. 0 high-order destination address set-up register (0x4822A)  
**D1MOD1–D1MOD0:** Ch. 1 transfer mode (D[F:E]) / Ch. 1 high-order destination address set-up register (0x4823A)  
**D2MOD1–D2MOD0:** Ch. 2 transfer mode (D[F:E]) / Ch. 2 high-order destination address set-up register (0x4824A)  
**D3MOD1–D3MOD0:** Ch. 3 transfer mode (D[F:E]) / Ch. 3 high-order destination address set-up register (0x4825A)

Select a transfer mode.

Table V.2.7 Transfer Mode

DxMOD1	DxMOD0	Mode
1	1	Invalid
1	0	Block transfer mode
0	1	Successive transfer mode
0	0	Single transfer mode

In single transfer mode, a transfer operation invoked by one trigger is completed after transferring one unit of data of the size set by DATSIZEx.

In successive transfer mode, data transfer operations are performed by one trigger a number of times as set by the transfer counter.

In block transfer mode, a transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKLENx.

At initial reset, DxMOD is set to "00" (single transfer mode).

**DATSIZE0:** Ch. 0 transfer data size (DE) / Ch. 0 high-order source address register (0x48226)

**DATSIZE1:** Ch. 1 transfer data size (DE) / Ch. 1 high-order source address register (0x48236)

**DATSIZE2:** Ch. 2 transfer data size (DE) / Ch. 2 high-order source address register (0x48246)

**DATSIZE3:** Ch. 3 transfer data size (DE) / Ch. 3 high-order source address register (0x48256)

Select the data size to be transferred.

Write "1": Half-word (16 bits)

Write "0": Byte (8 bits)

Read: Valid

The transfer data size is set to 16 bits by writing "1" to DATSIZEx and set to 8 bits by writing "0".

At initial reset, DATSIZEx is set to "0" (8 bits).

**S0IN1–S0IN0:** Ch. 0 source address control (D[D:C]) / Ch. 0 high-order source address set-up register (0x48226)

**S1IN1–S1IN0:** Ch. 1 source address control (D[D:C]) / Ch. 1 high-order source address set-up register (0x48236)

**S2IN1–S2IN0:** Ch. 2 source address control (D[D:C]) / Ch. 2 high-order source address set-up register (0x48246)

**S3IN1–S3IN0:** Ch. 3 source address control (D[D:C]) / Ch. 3 high-order source address set-up register (0x48256)

Control the incrementing or decrementing of the memory address.

Table V.2.8 Address Control

SxIN1	SxIN0	Address control
1	1	Increment without initialization
1	0	Increment with initialization
0	1	Decrement without initialization
0	0	Fixed

In dual-address mode, this setting applies to the source address. In single-address mode, this setting applies to the external memory address. When "address fixed" (00) is selected, the source address is not changed by a data transfer performed. Even when transferring multiple data, the transfer data is always read from the same address. When "address increment" (11 or 10) is selected in single and successive transfer modes, the source address is incremented by an amount equal to the data size set by DATSIZEx when one data transfer is completed. When "address decrement" (01) is selected, the source address is decremented in the same way. In block transfer mode too, the source address is incremented or decremented when one data unit is transferred. However, if SxIN is set to "10", the source address that has been incremented during a block transfer recycles back to the initial value when the block transfer is completed.

At initial reset, SxIN is set to "00" (Fixed).



**D0IN1–D0IN0:** Ch. 0 destination address control (D[D:C]) / Ch. 0 high-order destination address set-up register (0x4822A)  
**D1IN1–D1IN0:** Ch. 1 destination address control (D[D:C]) / Ch. 1 high-order destination address set-up register (0x4823A)  
**D2IN1–D2IN0:** Ch. 2 destination address control (D[D:C]) / Ch. 2 high-order destination address set-up register (0x4824A)  
**D3IN1–D3IN0:** Ch. 3 destination address control (D[D:C]) / Ch. 3 high-order destination address set-up register (0x4825A)

Control the incrementing or decrementing of the memory address.

Table V.2.9 Address Control

DxIN1	DxIN0	Address control
1	1	Increment without initialization
1	0	Increment with initialization
0	1	Decrement without initialization
0	0	Fixed

In dual-address mode, this setting applies to the destination address. In single-address mode, these bits are not used. When "address fixed" (00) is selected, the destination address is not changed by a data transfer performed. Even when transferring multiple data, the transfer data is always written to the same address. When "address increment" (11 or 10) is selected in single and successive transfer modes, the destination address is incremented by an amount equal to the data size set by DATSIZE<sub>x</sub> when one data transfer is completed. When "address decrement" (01) is selected, the destination address is decremented in the same way. In block transfer mode too, the destination address is incremented or decremented when one data unit is transferred. However, if DxIN is set to "10", the destination address that has been incremented during a block transfer recycles back to the initial value when the block transfer is completed.

At initial reset, DxIN is set to "00" (Fixed).

**BLKLEN07–BLKLEN00:** Ch. 0 block length/transfer counter[7:0] (D[7:0]) / Ch. 0 transfer counter register (0x48220)  
**BLKLEN17–BLKLEN10:** Ch. 1 block length/transfer counter[7:0] (D[7:0]) / Ch. 1 transfer counter register (0x48230)  
**BLKLEN27–BLKLEN20:** Ch. 2 block length/transfer counter[7:0] (D[7:0]) / Ch. 2 transfer counter register (0x48240)  
**BLKLEN37–BLKLEN30:** Ch. 3 block length/transfer counter[7:0] (D[7:0]) / Ch. 3 transfer counter register (0x48250)

In block transfer mode, these bits are used to specify a transfer block size. A transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKLEN<sub>x</sub>.

In single or successive transfer mode, these bits are used to specify the 8 low-order bits of the transfer counter.

At initial reset, these bits are not initialized.

**Note:** When performing data transfer in block transfer mode, the block size must not be set to "0".

**TC0\_L7–TC0\_L0:** Ch. 0 transfer counter[7:0]/[15:8] (D[F:8]) / Ch. 0 transfer counter register (0x48220)  
**TC0\_H7–TC0\_H0:** Ch. 0 transfer counter[15:8]/[23:16] (D[7:0]) / Ch. 0 control register (0x48222)  
**TC1\_L7–TC1\_L0:** Ch. 1 transfer counter[7:0]/[15:8] (D[F:8]) / Ch. 1 transfer counter register (0x48230)  
**TC1\_H7–TC1\_H0:** Ch. 1 transfer counter[15:8]/[23:16] (D[7:0]) / Ch. 1 control register (0x48232)  
**TC2\_L7–TC2\_L0:** Ch. 2 transfer counter[7:0]/[15:8] (D[F:8]) / Ch. 2 transfer counter register (0x48240)  
**TC2\_H7–TC2\_H0:** Ch. 2 transfer counter[15:8]/[23:16] (D[7:0]) / Ch. 2 control register (0x48242)  
**TC3\_L7–TC3\_L0:** Ch. 3 transfer counter[7:0]/[15:8] (D[F:8]) / Ch. 3 transfer counter register (0x48250)  
**TC3\_H7–TC3\_H0:** Ch. 3 transfer counter[15:8]/[23:16] (D[7:0]) / Ch. 3 control register (0x48252)

Set the data transfer count.

In block transfer mode, TC<sub>x</sub>\_L[7:0] is bits[7:0] of the transfer counter, and TC<sub>x</sub>\_H[7:0] is bits[15:8] of the transfer counter.

In single or successive transfer mode, TC<sub>x</sub>\_L[7:0] is bits[15:8] of the transfer counter, and TC<sub>x</sub>\_H[7:0] is bits[23:16] of the transfer counter. The 8 low-order bits are specified by BLKLEN<sub>x</sub>[7:0].

This counter is decremented each time a DMA transfer in the corresponding channel is performed. When the counter reaches 0, an interrupt factor is generated. In single-address mode, the end-of-transfer signal is output from the #DMAEND<sub>x</sub> pin at the same time.

Even when the counter is 0, a DMA request is accepted and the counter is decremented to "0xFFFF" (or "0xFFFFFFFF").

Be sure to disable DMA transfers (HS<sub>x</sub>\_EN = "0") before writing and reading to and from the counter.

At initial reset, these bits are not initialized.

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

---

**S0ADRL15–S0ADRL0:** Ch. 0 source address[15:0] (D[F:0]) / Ch. 0 low-order source address set-up register (0x48224)  
**S0ADRH11–S0ADRH0:** Ch. 0 source address[27:16] (D[B:0]) / Ch. 0 high-order source address set-up register (0x48226)  
**S1ADRL15–S1ADRL0:** Ch. 1 source address[15:0] (D[F:0]) / Ch. 1 low-order source address set-up register (0x48234)  
**S1ADRH11–S1ADRH0:** Ch. 1 source address[27:16] (D[B:0]) / Ch. 1 high-order source address set-up register (0x48236)  
**S2ADRL15–S2ADRL0:** Ch. 2 source address[15:0] (D[F:0]) / Ch. 2 low-order source address set-up register (0x48244)  
**S2ADRH11–S2ADRH0:** Ch. 2 source address[27:16] (D[B:0]) / Ch. 2 high-order source address set-up register (0x48246)  
**S3ADRL15–S3ADRL0:** Ch. 3 source address[15:0] (D[F:0]) / Ch. 3 low-order source address set-up register (0x48254)  
**S3ADRH11–S3ADRH0:** Ch. 3 source address[27:16] (D[B:0]) / Ch. 3 high-order source address set-up register (0x48256)

---

In dual-address mode, these bits are used to specify a source address. In single-address mode, an external memory address at the destination or source of transfer is specified.

Use SxADRL to set the 16 low-order bits of the address and SxADRH to set the 12 high-order bits.

Be sure to disable DMA transfers (HSx\_EN = "0") before writing or reading to and from these registers.

The address is incremented or decremented (as set by SxIN) according to the transfer data size each time a DMA transfer in the corresponding channel is performed.

At initial reset, these bits are not initialized.

---

**D0ADRL15–D0ADRL0:** Ch. 0 destination address[15:0] (D[F:0]) / Ch. 0 low-order destination address set-up register (0x48228)  
**D0ADRH11–D0ADRH0:** Ch. 0 destination address[27:16] (D[B:0]) / Ch. 0 high-order destination address set-up register (0x4822A)  
**D1ADRL15–D1ADRL0:** Ch. 1 destination address[15:0] (D[F:0]) / Ch. 1 low-order destination address set-up register (0x48238)  
**D1ADRH11–D1ADRH0:** Ch. 1 destination address[27:16] (D[B:0]) / Ch. 1 high-order destination address set-up register (0x4823A)  
**D2ADRL15–D2ADRL0:** Ch. 2 destination address[15:0] (D[F:0]) / Ch. 2 low-order destination address set-up register (0x48248)  
**D2ADRH11–D2ADRH0:** Ch. 2 destination address[27:16] (D[B:0]) / Ch. 2 high-order destination address set-up register (0x4824A)  
**D3ADRL15–D3ADRL0:** Ch. 3 destination address[15:0] (D[F:0]) / Ch. 3 low-order destination address set-up register (0x48258)  
**D3ADRH11–D3ADRH0:** Ch. 3 destination address[27:16] (D[B:0]) / Ch. 3 high-order destination address set-up register (0x4825A)

---

In dual-address mode, these bits are used to specify a destination address. In single-address mode, these bits are not used.

Be sure to disable DMA transfers (HSx\_EN = "0") before writing or reading to and from these registers.

The address is incremented or decremented (as set by DxIN) according to the transfer data size each time a DMA transfer in the corresponding channel is performed.

At initial reset, these bits are not initialized.

---

**PHSD0L2–PHSD0L0:** Ch. 0 interrupt level (D[2:0]) / HSDMA Ch. 0/1 interrupt priority register (0x40263)  
**PHSD1L2–PHSD1L0:** Ch. 1 interrupt level (D[6:4]) / HSDMA Ch. 0/1 interrupt priority register (0x40263)  
**PHSD2L2–PHSD2L0:** Ch. 2 interrupt level (D[2:0]) / HSDMA Ch. 2/3 interrupt priority register (0x40264)  
**PHSD3L2–PHSD3L0:** Ch. 3 interrupt level (D[6:4]) / HSDMA Ch. 2/3 interrupt priority register (0x40264)

---

Set the priority level of an end-of-DMA interrupt in the range of 0 to 7.

At initial reset, these registers become indeterminate.

---

**EHDM0:** Ch. 0 interrupt enable (D0) / DMA interrupt enable register (0x40271)  
**EHDM1:** Ch. 1 interrupt enable (D1) / DMA interrupt enable register (0x40271)  
**EHDM2:** Ch. 2 interrupt enable (D2) / DMA interrupt enable register (0x40271)  
**EHDM3:** Ch. 3 interrupt enable (D3) / DMA interrupt enable register (0x40271)

---

Enable or disable interrupt generation to the CPU.

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

EHDMx is the interrupt enable bit for HSDMA channel x. The interrupt is enabled when EHDMx is set to "1" and disabled when EHDMx is set to "0".

At initial reset, EHDMx is set to "0" (interrupt disabled).

---

**FHDM0:** Ch. 0 interrupt factor flag (D0) / DMA interrupt factor flag register (0x40281)

**FHDM1:** Ch. 1 interrupt factor flag (D1) / DMA interrupt factor flag register (0x40281)

**FHDM2:** Ch. 2 interrupt factor flag (D2) / DMA interrupt factor flag register (0x40281)

**FHDM3:** Ch. 3 interrupt factor flag (D3) / DMA interrupt factor flag register (0x40281)

---

Indicate the occurrence status of HSDMA interrupt factor.

**When read**

Read "1": Interrupt factor generated

Read "0": No interrupt factor generated

**When written using the reset-only method (default)**

Write "1": Factor flag is reset

Write "0": Invalid

**When written using the read/write method**

Write "1": Factor flag is set

Write "0": Factor flag is reset

FHDMx is the interrupt factor flag for HSDMA channel x. These flags are set to "1" when the transfer counter reaches 0.

An interrupt to the CPU is generated if the following conditions are met at this time:

1. The corresponding interrupt enable register is set to "1".
2. No other interrupt request of higher priority is generated.
3. The IE bit of the PSR is set to "1" (interrupt enable).
4. The corresponding interrupt priority register is set to a level higher than the CPU's interrupt level (IL).

When using an interrupt factor to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the interrupt factor that has occurred. If interrupts are enabled at the setting of the IDMA side, an interrupt is generated under the above conditions after the data transfer by IDMA is completed. The interrupt factor flag is always set to "1" when an interrupt factor occurs no matter how the interrupt enable and interrupt priority registers are set.

In order for the next interrupt to be accepted after interrupt generation, the interrupt factor flag must be reset and the PSR must be set up again (by setting the IL below the level indicated by the interrupt priority register and setting the IE bit to "1" or executing the reti instruction).

The interrupt factor flag can only be reset by a write instruction in the software application. If the PSR is again set up to accept interrupts (or the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt may occur again. Note also that the value to be written to reset the flag is "1" when using the reset-only method (RSTONLY = "1") and "0" when using the read/write method (RSTONLY = "0"). Be careful not to confuse these two cases.

The FHDMx flag becomes indeterminate when initially reset, so be sure to reset the flag in the software application.

## V-2 DMA BLOCK: HSDMA (High-Speed DMA)

---

**RHDM0:** Ch.0 IDMA request (D4) / Port input 0–3, HSDMA, 16-bit timer 0 IDMA request register (0x40290)

**RHDM1:** Ch.1 IDMA request (D5) / Port input 0–3, HSDMA, 16-bit timer 0 IDMA request register (0x40290)

---

Specify whether IDMA need to be invoked when an interrupt factor occurs.

### When using the set-only method (default)

Write "1": IDMA request

Write "0": Not changed

Read: Valid

### When using the read/write method

Write "1": IDMA request

Write "0": Interrupt request

Read: Valid

RHDM0 and RHDM1 are the IDMA request bits for HSDMA channels 0 and 1, respectively. If the bit is set to "1", IDMA is invoked when an interrupt factor occurs, thus performing a programmed data transfer. If the register is set to "0", regular interrupt processing is performed without ever invoking IDMA.

For details on IDMA, refer to "IDMA (Intelligent DMA)".

At initial reset, RHDMx is set to "0" (interrupt request).

---

**DEHDM0:** Ch.0 IDMA enable (D4) / Port input 0–3, HSDMA, 16-bit timer 0 IDMA enable register (0x40294)

**DEHDM1:** Ch.1 IDMA enable (D5) / Port input 0–3, HSDMA, 16-bit timer 0 IDMA enable register (0x40294)

---

Enables IDMA transfer by means of an interrupt factor.

### When using the set-only method (default)

Write "1": IDMA enabled

Write "0": Not changed

Read: Valid

### When using the read/write method

Write "1": IDMA enabled

Write "0": IDMA disabled

Read: Valid

DEHDM0 and DEHDM1 are the IDMA enable bits for HSDMA channels 0 and 1, respectively. If DEHDMx is set to "1", the IDMA request by the interrupt factor is enabled. If the bit is set to "0", the IDMA request is disabled.

At initial reset, DEHDMx is set to "0" (IDMA disabled).

## Programming Notes

---

- (1) When setting the transfer conditions, always make sure the DMA controller is inactive (HSx\_EN = "0").
- (2) When performing data transfer in block transfer mode, the block size must not be set to "0".
- (3) After an initial reset, the interrupt factor flag (FHDMx) becomes indeterminate. Always be sure to reset the flag to prevent interrupts or IDMA requests from being generated inadvertently.
- (4) To prevent an interrupt from being generated repeatedly for the same factor, be sure to reset the interrupt factor flag before setting up the PSR again or executing the reti instruction.
- (5) HSDMA is given higher priority over IDMA (intelligent DMA) and the CPU. However, since HSDMA and IDMA share the same circuit, HSDMA cannot gain the bus ownership while an IDMA transfer is under way. Requests for HSDMA invocation that have occurred during an IDMA transfer are kept pending until the IDMA transfer is completed.  
A request for IDMA invocation or an interrupt request that has occurred during a HSDMA transfer are accepted after completion of the HSDMA transfer.
- (6) In HALT mode, since the DMA and BCU clocks operate, if the next operation is performed in HALT mode, not HALT2 mode, with a setting of 0 in clock option register HLT2OP (D3/0x40190), that operation will be an unpredictable erroneous operation.  
If a DMA trigger occurs and DMA is invoked while the CPU is stopped after HALT mode execution, erroneous operation will result. Ensure that DMA is not invoked in HALT mode.  
In HALT2 mode, DMA is not invoked since the DMA and BCU clocks are stopped.

THIS PAGE IS BLANK.

## V-3 IDMA (Intelligent DMA)

### *Functional Outline of IDMA*

---

The DMA Block contains an intelligent DMA (IDMA), a function that allows control information to be programmed in RAM. Up to 128 channels can be programmed, including 31 channels that are invoked by an interrupt factor that occurs in some internal peripheral circuit.

Although an additional overhead for loading and storing control information in RAM may be incurred, this intelligent DMA supports such functions as successive transfers, block transfers, and linking to another IDMA.

IDMA is invoked by an interrupt factor that occurs in some internal peripheral circuit or a software trigger, thereby performing a data transfer according to the control information in RAM. When the transfer is completed, IDMA can generate an interrupt or invoke another IDMA according to link settings.

### *Programming Control Information*

---

The intelligent DMA operates according to the control information prepared in RAM. The control information can be stored in either internal RAM or external RAM should the necessary area be allocated.

The control information is 3 words (12 bytes) per channel in size, and must be located at contiguous addresses beginning with the base address that is set in the software application as the starting address of channel 0. Consequently, an area of 384 words (1,536 bytes) in RAM is required in order for all of 128 channels to be used.

The following explains how to set the base address and the contents of control information. Before using IDMA, make each the settings described below.

#### **Setting the base address**

Set the starting address of control information (starting address of channel 0) in the IDMA base address register.

16 low-order bits: DBASEL[15:0] (D[F:0]) / IDMA base address low-order register (0x48200)

12 high-order bits: DBASEH[11:0] (D[B:0]) / IDMA base address high-order register (0x48202)

When initially reset, the base address is set to 0x0C003A0.

- Notes:**
- The address you set in the IDMA base address register must always be a word (32-bit) boundary address.
  - Be sure to disable DMA transfers (IDMAEN = "0") before setting the base address. Writing to the IDMA base address register is ignored when the DMA transfer is enabled (IDMAEN = "1"). When the register is read, the read data is indeterminate.

#### **Control information**

Write the control information for the IDMA channels used to RAM.

The addresses at which the control information of each channel is placed are determined by the base address and a channel number.

Starting address of channel = base address + (channel number × 12 [bytes])

- Note:** The control information must be written only when the channel to be set does not start a DMA transfer. If a DMA transfer starts when the control information is being written to the RAM, proper transfer cannot be performed. Reading the control information can always be done.

### V-3 DMA BLOCK: IDMA (Intelligent DMA)

The contents of control information (3 words) in each channel are shown in the table below.

Table V.3.1 IDMA Control Information

Word	Bit	Name	Function		
1st	D31	LNKEN	IDMA link enable "1" = Enabled, "0" = Disabled		
	D30–24	LNKCHN[6:0]	IDMA link field		
	D23–8	TC[15:0]	Transfer counter (block transfer mode) Transfer counter - high-order 16 bits (single or successive transfer mode)		
	D7–0	BLKLEN[7:0]	Block size (block transfer mode) Transfer counter - low-order 8 bits (single or successive transfer mode)		
2nd	D31	DINTEN	End-of-transfer interrupt enable "1" = Enabled, "0" = Disabled		
	D30	DATSIZ	Data size control "1" = Half-word, "0" = Byte		
	D29–28	SRINC[1:0]	Source address control		
			SRINC1	SRINC0	Setting contents
			1	1	Address incremented (In block transfer mode, the transfer address is updated without reset using the initial value.)
1			0	Address incremented (In block transfer mode, the transfer address is updated with the initial value.)	
0	1	Address decremented (In block transfer mode, the transfer address is updated without reset using the initial value.)			
0	0	Address fixed			
D27–0	SRADR[27:0]	Source address			
3rd	D31–30	DMOD[1:0]	Transfer mode (Do not set to "11".)		
			DMOD1	DMOD0	Setting contents
			1	0	Block transfer mode
			0	1	Successive transfer mode
	0	0	Single transfer mode		
D29–28	DSINC[1:0]	Destination address control			
		DSINC1	DSINC0	Setting contents	
		1	1	Address incremented (In block transfer mode, the transfer address is updated without reset using the initial value.)	
		1	0	Address incremented (In block transfer mode, the transfer address is updated with the initial value.)	
0	1	Address decremented (In block transfer mode, the transfer address is updated without reset using the initial value.)			
0	0	Address fixed			
D27–0	DSADR[27:0]	Destination address			

#### LNKEN: IDMA link enable (D31/1st Word)

If this bit remains set (= "1"), the IDMA channel that is set in the IDMA link field is invoked after the completion of a DMA transfer in this channel. DMA transfers in multiple channels can be performed successively by merely triggering the first channel to be executed. There is no limit to the number of channels linked. Set this link in order of the IDMA channels you want to be executed.

If this bit is "0", IDMA is completed by merely executing a DMA transfer in this channel.

#### LNKCHN[6:0]: IDMA link field (D[30:24]/1st Word)

If you want IDMA to be linked, set the channel numbers (0 to 127) to be executed next.

The data in this field is valid only when LINKEN = "1".

#### TC[15:0]: Transfer counter (D[23:8]/1st Word)

In block transfer mode, a transfer count can be specified using up to 16 bits. Set this value here. In single transfer and successive transfer modes, a transfer count can be specified using up to 24 bits. Set a 16-bit high-order value here.



**BLKLEN[7:0]: Block size/transfer counter (D[7:0]/1st Word)**

In block transfer mode, set the size of a block that is transferred in one operation (in units of DATSIZ). In single transfer and successive transfer modes, set an 8-bit low-order value for the transfer count here.

- Notes:**
- When performing data transfer in block transfer mode, the block size must not be set to "0".
  - The transfer count thus set is decremented according to the transfers performed. If the transfer count is set to 0, it is decremented to all Fs by the first transfer performed. This means that you have set the maximum value that is determined by the number of bits available.

**DINTEN: End-of-transfer interrupt enable (D31/2nd Word)**

If this bit is left set (= "1"), when the transfer counter reaches 0, an interrupt request to the CPU is generated based on the interrupt factor flag by which IDMA has been invoked.

If this bit is "0", no interrupt request to the CPU is generated even when the transfer counter has reached 0.

**DATSIZ: Data size control (D30/2nd Word)**

Set the unit size of data to be transferred.

A half-word size (16 bits) is assumed if this bit is "1" and a byte size (8 bits) is assumed if this bit is "0".

**SRINC[1:0]: Source address control (D[29:28]/2nd Word)**

Set the source address updating format.

If the format is set for "address fixed" (00), the source address is not changed by a data transfer performed. Even when transferring multiple data, the transfer data is always read from the same address.

If the format is set for "address increment" (11 or 10) in single and successive transfer modes, the source address is incremented by an amount equal to the data size set by DATSIZ when one data transfer is completed.

If the format is set for "address decrement" (01), the source address is decremented in the same way.

In block transfer mode too, the source address is incremented or decremented when one data unit is transferred. However, if the set format is "10", the source address that has been incremented during a block transfer recycles back to the initial value when the block transfer is completed.

**SRADR[27:0]: Source address (D[27:0]/2nd Word)**

Use these bits to set the starting address at the source of transfer. The content set here is updated according to the setting of SRINC.

**DMOD[1:0]: Transfer mode (D[31:30]/3rd Word)**

Use these bits to set the desired transfer mode.

The transfer modes are outlined below (to be detailed later):

**• Single transfer mode (00)**

In this mode, a transfer operation invoked by one trigger is completed after transferring one unit of data of the size set by DATSIZ. If data transfer need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

**• Successive transfer mode (01)**

In this mode, data transfer operations are performed by one trigger a number of times as set by the transfer counter. The transfer counter is decremented to 0 each time data is transferred.

**• Block transfer mode (10)**

In this mode, a transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKLEN. If a block transfer need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

**DSINC[1:0]: Destination address control (D[29:28]/3rd Word)**

Set the destination address update format.

If the format is set for "address fixed" (00), the destination address is not changed by the performance of a data transfer operation. Even when transferring multiple data, the transfer data is always written to the same address.

If the format is set for "address increment" (11 or 10) in single and successive transfer modes, the destination address is incremented by an amount equal to the data size set by DATSIZ when one data transfer is completed.

If the format is set for "address decrement" (01), the destination address is decremented in the same way.

In block transfer mode as well, the destination address is incremented or decremented when one data unit is transferred. However, if the set format is "10", the destination address that has been incremented during a block transfer recycles back to the initial value when the block transfer is completed.

**DSADR[27:0]: Destination address (D[27:0]/3rd Word)**

Use these bits to set the starting address at the destination of transfer. The content set here is updated according to the setting of DSINC.

Since the control information is placed in RAM, it can be rewritten. However, before rewriting the content of this information, make sure that no DMA transfer is generated in the channel whose information you are going to rewrite.

## IDMA Invocation

The triggers by which IDMA is invoked have the following three causes:

1. Interrupt factor in an internal peripheral circuit
2. Trigger in the software application
3. Link setting

### Enabling/disabling DMA transfer

The IDMA controller is enabled by writing "1" to the IDMA enable bit IDMAEN (D0) / IDMA enable register (0x48205), and is ready to accept the triggers described above. However, before enabling a DMA transfer, be sure to set the base address and the control information for the channel to be invoked correctly. If IDMAEN is set to "0", no IDMA invocation request is accepted.

### IDMA invocation by an interrupt factor in internal peripheral circuits

Some internal peripheral circuits that have an interrupt generating function can invoke IDMA by an interrupt factor in that circuit. The IDMA channel numbers corresponding to such IDMA invocation are predetermined. The relationship between the interrupt factors that have this function and the IDMA channels is shown in Table V.3.2.

Table V.3.2 Interrupt Factors Used to Invoke IDMA

Peripheral circuit	Interrupt factor	IDMA Ch.	IDMA request bit	IDMA enable bit
Ports	Port input 0	1	RP0 (D0/0x40290)	DEP0 (D0/0x40294)
	Port input 1	2	RP1 (D1/0x40290)	DEP1 (D1/0x40294)
	Port input 2	3	RP2 (D2/0x40290)	DEP2 (D2/0x40294)
	Port input 3	4	RP3 (D3/0x40290)	DEP3 (D3/0x40294)
High-speed DMA	Ch.0, end of transfer	5	RHDM0 (D4/0x40290)	DEHDM0 (D4/0x40294)
	Ch.1, end of transfer	6	RHDM1 (D5/0x40290)	DEHDM1 (D5/0x40294)
16-bit programmable timer	Timer 0 comparison B	7	R16TU0 (D6/0x40290)	DE16TU0 (D6/0x40294)
	Timer 0 comparison A	8	R16TC0 (D7/0x40290)	DE16TC0 (D7/0x40294)
	Timer 1 comparison B	9	R16TU1 (D0/0x40291)	DE16TU1 (D0/0x40295)
	Timer 1 comparison A	10	R16TC1 (D1/0x40291)	DE16TC1 (D1/0x40295)
	Timer 2 comparison B	11	R16TU2 (D2/0x40291)	DE16TU2 (D2/0x40295)
	Timer 2 comparison A	12	R16TC2 (D3/0x40291)	DE16TC2 (D3/0x40295)
	Timer 3 comparison B	13	R16TU3 (D4/0x40291)	DE16TU3 (D4/0x40295)
	Timer 3 comparison A	14	R16TC3 (D5/0x40291)	DE16TC3 (D5/0x40295)
	Timer 4 comparison B	15	R16TU4 (D6/0x40291)	DE16TU4 (D6/0x40295)
	Timer 4 comparison A	16	R16TC4 (D7/0x40291)	DE16TC4 (D7/0x40295)
8-bit programmable timer 0–3	Timer 5 comparison B	17	R16TU5 (D0/0x40292)	DE16TU5 (D0/0x40296)
	Timer 5 comparison A	18	R16TC5 (D1/0x40292)	DE16TC5 (D1/0x40296)
	Timer 0 underflow	19	R8TU0 (D2/0x40292)	DE8TU0 (D2/0x40296)
	Timer 1 underflow	20	R8TU1 (D3/0x40292)	DE8TU1 (D3/0x40296)
Serial interface Ch.0/1	Timer 2 underflow	21	R8TU2 (D4/0x40292)	DE8TU2 (D4/0x40296)
	Timer 3 underflow	22	R8TU3 (D5/0x40292)	DE8TU3 (D5/0x40296)
	Ch.0 receive buffer full	23	RSRX0 (D6/0x40292)	DESRX0 (D6/0x40296)
A/D converter	Ch.0 transmit buffer empty	24	RSTX0 (D7/0x40292)	DESTX0 (D7/0x40296)
	Ch.1 receive buffer full	25	RSRX1 (D0/0x40293)	DESRX1 (D0/0x40297)
	Ch.1 transmit buffer empty	26	RSTX1 (D1/0x40293)	DESTX1 (D1/0x40297)
	End of A/D conversion	27	RADE (D2/0x40293)	DEADE (D2/0x40297)
Ports	Port input 4	28	RP4 (D4/0x40293)	DEP4 (D4/0x40297)
	Port input 5	29	RP5 (D5/0x40293)	DEP5 (D5/0x40297)
	Port input 6	30	RP6 (D6/0x40293)	DEP6 (D6/0x40297)
	Port input 7	31	RP7 (D7/0x40293)	DEP7 (D7/0x40297)
8-bit programmable timer 4/5	Timer 4 underflow	32	R8TU4 (D0/0x4029B)	DE8TU4 (D0/0x4029C)
	Timer 5 underflow	33	R8TU5 (D1/0x4029B)	DE8TU5 (D1/0x4029C)
Serial interface Ch.2/3	Ch.2 receive buffer full	34	RSRX2 (D2/0x4029B)	DESRX2 (D2/0x4029C)
	Ch.2 transmit buffer empty	35	RSTX2 (D3/0x4029B)	DESTX2 (D3/0x4029C)
	Ch.3 receive buffer full	36	RSRX3 (D4/0x4029B)	DESRX3 (D4/0x4029C)
	Ch.3 transmit buffer empty	37	RSTX3 (D5/0x4029B)	DESTX3 (D5/0x4029C)
FIFO serial interface Ch.0	Ch.0 receive buffer full	56	RFSRX0 (D2/0x402B3)	DEFFSRX0 (D2/0x402B4)
	Ch.0 transmit buffer empty	57	RFSTX0 (D3/0x402B3)	DEFSTX0 (D3/0x402B4)

### V-3 DMA BLOCK: IDMA (Intelligent DMA)

These interrupt factors are used in common for interrupt requests and IDMA invocation requests.

To invoke IDMA upon the occurrence of an interrupt factor, set the corresponding bits of the IDMA request and IDMA enable registers shown in the table by writing "1". Then when an interrupt factor occurs, an interrupt request to the CPU is kept pending and the corresponding IDMA channel is invoked.

The interrupt factor flag that has been set to "1" remains set until the DMA transfer invoked by it is completed. If the following two conditions are met when one DMA transfer is completed, an interrupt request is generated without resetting the interrupt factor flag.

- The transfer counter has reached 0.
- DINTEN in control information is set to "1" (interrupt enabled).

In this case, the IDMA request register is cleared to "0". Therefore, if IDMA needs to be invoked when an interrupt factor occurs next time, this register must be set up again. To prevent unwanted IDMA requests from being generated, this setting must be performed before enabling interrupts and after resetting the interrupt factor flag. The IDMA enable bit is not cleared and remains set to "1".

If the transfer counter is not 0, the interrupt factor flag is reset when the DMA transfer is completed, so that no interrupt is generated. In this case, the IDMA request bit and IDMA enable bit are not cleared and remain set to "1".

When DINTEN in control information has been set to "0", the interrupt factor flag is reset even if the transfer counter reaches 0, so that no interrupt is generated. In this case, the IDMA request bit is not cleared but the IDMA enable bit is cleared.

If the IDMA request register bit is left reset to "0", the relevant interrupt factor generates an interrupt request and not a IDMA request.

The control registers (interrupt enable register and interrupt priority register) corresponding to the interrupt factor do not affect IDMA invocation. IDMA can be invoked even if the interrupt enable bit in ITC is set to "0" (interrupt disabled). However, these register must be set to enable the interrupt when generating the interrupt after completing the DMA transfer.

#### **IDMA invocation by a trigger in the software application**

All IDMA channels for which control information is set, including those corresponding to interrupt factors described above, can be invoked by a trigger in the software application.

The following bits are used for this control:

IDMA channel number set-up: DCHN[6:0] (D[6:0]) / IDMA start register (0x48204)

IDMA start control:                   DSTART (D7) / IDMA start register (0x48204)

When the IDMA channel number to be invoked (0 to 127) is written to DCHN and DSTART is set to "1", the specified IDMA channel starts a DMA transfer.

DSTART remains set (= "1") during a DMA transfer and is reset to "0" in hardware when one DMA transfer operation is completed.

Do not modify these bits during a DMA transfer.

If DINTEN is set to "1" (interrupt enabled), an interrupt factor for the completion of IDMA transfer is generated when one DMA transfer is completed.

#### **IDMA invocation by link setting**

If LNKEN in the control information is set to "1" (link enabled), the IDMA channel that is set in the IDMA link field "LNKCHN" is invoked successively after a DMA transfer in the link-enabled channel is completed.

The interrupt request by the first channel is generated after transfers in all linked channels are completed if the interrupt conditions are met.

To generate an interrupt at the end of an IDMA transfer, the DINTEN (end-of-transfer interrupt enable) bits in the IDMA control information for the first IDMA channel to be invoked and all the channels to be linked must be set to "1".

**IDMA invocation request during a DMA transfer**

An IDMA invocation request to another channel that is generated during a DMA transfer is kept pending until the DMA transfer that was being executed at the time is completed. Since an invocation request is not cleared, new requests will be accepted when the DMA transfer under execution is completed.

An IDMA invocation request to the same channel cannot be accepted while the channel is executing a DMA transfer because the same interrupt factor is used. Therefore, an interval longer than the DMA transfer period is required when invoking the same channel.

**IDMA invocation request when DMA transfer is disabled**

An IDMA invocation request generated when IDMAEN is "0" (DMA transfer disabled) is kept pending until IDMAEN is set to "1". Since an invocation request is not cleared, it is accepted when DMA transfer is enabled.

**Simultaneous generation of a software trigger and a hardware trigger**

When a software trigger and the hardware trigger for the same channel are generated simultaneously, the software trigger starts IDMA transfer. The IDMA transfer by the hardware trigger is not executed since the interrupt factor is reset when the DMA transfer is completed. However, an operation like this cannot be recommended.

## Operation of IDMA

IDMA has three transfer modes, in each of which data transfer operates differently. Furthermore, an interrupt factor is processed differently depending on the type of trigger. The following describes the operation of IDMA in each transfer mode and how an interrupt factor is processed for each type of trigger.

### Single transfer mode

The channels for which DMOD in control information is set to "00" operate in single transfer mode. In this mode, a transfer operation invoked by one trigger is completed after transferring one data unit of the size set by DATSIZ. If a data transfer needs to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

The operation of IDMA in single transfer mode is shown by the flow chart in Figure V.3.1.

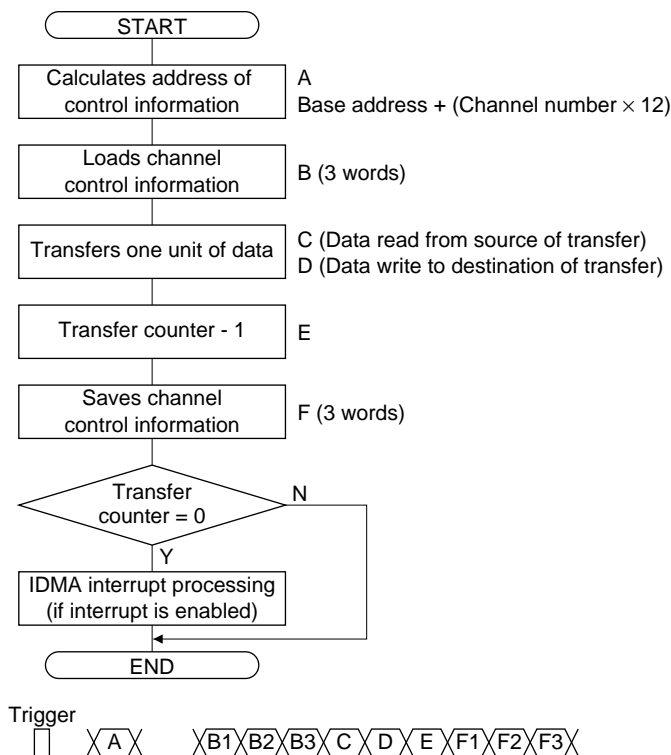


Figure V.3.1 Operation Flow in Single Transfer Mode

- (1) When a trigger is accepted, the address for control information is calculated from the base address and channel number.
- (2) Control information is read from the calculated address into the internal temporary register.
- (3) Data of the size set in the control information is read from the source address.
- (4) The read data is written to the destination address.
- (5) The address is incremented or decremented and the transfer counter is decremented.
- (6) The modified control information is written to RAM.
- (7) In the case of a hardware trigger, the interrupt control bits are processed before completing IDMA.

Condition	Interrupt factor flag	IDMA request bit	IDMA enable bit
Transfer counter ≠ "0":	Reset ("0")	Not changed ("1")	Not changed ("1")
Transfer counter = "0", DINTEN = "1":	Not changed ("1")	Reset ("0")	Not changed ("1")
Transfer counter = "0", DINTEN = "0":	Reset ("0")	Not changed ("1")	Reset ("0")

### Successive transfer mode

The channels for which DMOD in control information is set to "01" operate in successive transfer mode. In this mode, a data transfer is performed by one trigger a number of times as set by the transfer counter. The transfer counter is decremented to "0" by one transfer executed.

The operation of IDMA in successive transfer mode is shown by the flow chart in Figure V.3.2.

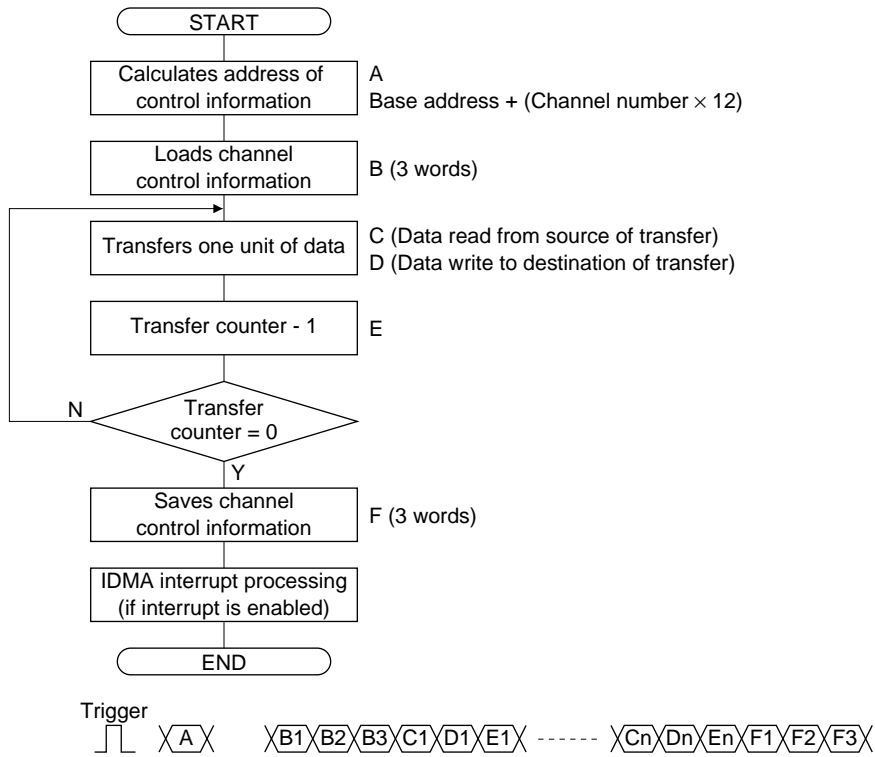


Figure V.3.2 Operation Flow in Successive Transfer Mode

- (1) When a trigger is accepted, the address for control information is calculated from the base address and channel number.
- (2) Control information is read from the calculated address into the internal temporary register.
- (3) Data of the size set in the control information is read from the source address.
- (4) The read data is written to the destination address.
- (5) The address is incremented or decremented and the transfer counter is decremented.
- (6) Steps (3) to (5) are repeated until the transfer counter reaches 0.
- (7) The modified control information is written to RAM.
- (8) In the case of a hardware trigger, the interrupt control bits are processed before completing IDMA.

Condition	Interrupt factor flag	IDMA request bit	IDMA enable bit
Transfer counter ≠ "0":	Reset ("0")	Not changed ("1")	Not changed ("1")
Transfer counter = "0", DINTEN = "1":	Not changed ("1")	Reset ("0")	Not changed ("1")
Transfer counter = "0", DINTEN = "0":	Reset ("0")	Not changed ("1")	Reset ("0")

**Block transfer mode**

The channels for which DMOD in control information is set to "10" operate in block transfer mode. In this mode, a transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKLEN. If a block transfer needs to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

The operation of IDMA in block transfer mode is shown by the flow chart in Figure V.3.3.

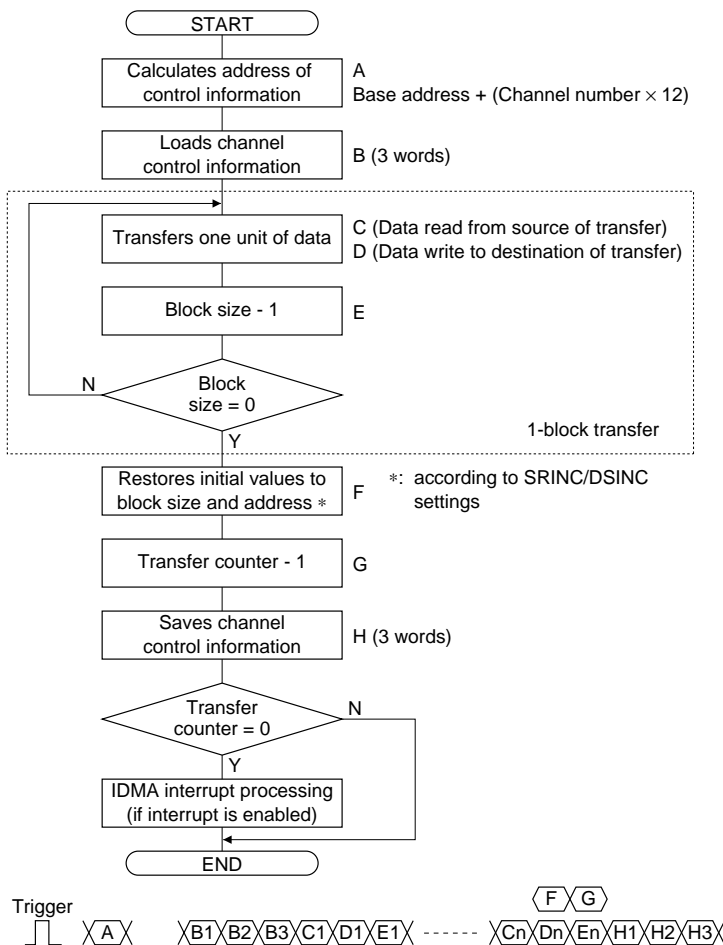


Figure V.3.3 Operation Flow in Block Transfer Mode

- (1) When a trigger is accepted, the address for control information is calculated from the base address and channel number.
- (2) Control information is read from the calculated address into the internal temporary register.
- (3) Data of the size set in the control information is read from the source address.
- (4) The read data is written to the destination address.
- (5) The address is incremented or decremented and BLKLEN is decremented.
- (6) Steps (3) to (5) are repeated until BLKLEN reaches 0.
- (7) If SRINC and DSINC are "10", the address is recycled to the initial value.
- (8) The transfer counter is decremented.
- (9) The modified control information is written to RAM.
- (10) In the case of a hardware trigger, the interrupt control bits are processed before completing IDMA.

Condition	Interrupt factor flag	IDMA request bit	IDMA enable bit
Transfer counter ≠ "0":	Reset ("0")	Not changed ("1")	Not changed ("1")
Transfer counter = "0", DINTEN = "1":	Not changed ("1")	Reset ("0")	Not changed ("1")
Transfer counter = "0", DINTEN = "0":	Reset ("0")	Not changed ("1")	Reset ("0")



## Processing of interrupt factors by type of trigger

### • When invoked by an interrupt factor

The interrupt factor flag by which IDMA has been invoked remains set even during a DMA transfer.

If the transfer counter is decremented to 0 and DINTEN = "1" (interrupt enabled) when one DMA transfer is completed, the interrupt factor that has invoked IDMA is not reset and an interrupt request is generated. At the same time, the IDMA request register is cleared to "0". The IDMA enable bit is not cleared and remains set to "1".

If the transfer counter is not 0, the interrupt factor flag is reset when the DMA transfer is completed, so that no interrupt is generated. In this case, the IDMA request bit and IDMA enable bit are not cleared and remain set to "1".

When DINTEN has been set to "0" (interrupt disabled), the interrupt factor flag is reset even if the transfer counter reaches 0, so that no interrupt is generated. In this case, the IDMA request bit is not cleared but the IDMA enable bit is cleared.

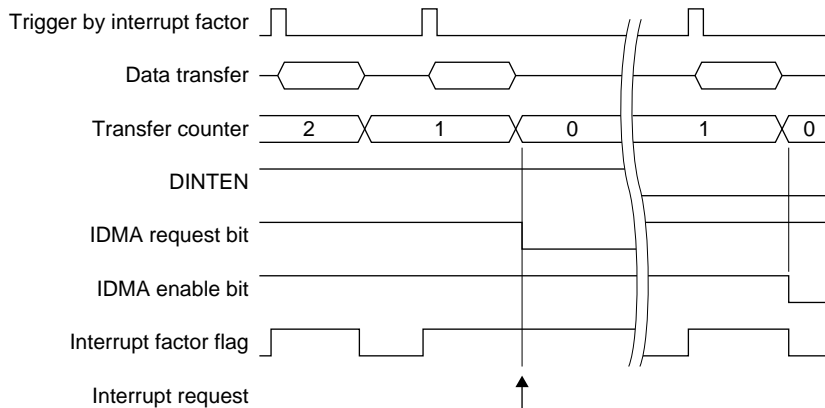


Figure V.3.4 Operation when Invoked by Interrupt Factor

When IDMA is invoked by the software trigger, the IDMA interrupt factor flag FIDMA (D4)/DMA interrupt factor flag register (0x40281) will not be set.

### • When invoked by a software trigger

If the transfer counter is decremented to 0 and DINTEN = "1" (interrupt enabled) when one DMA transfer is completed, the IDMA interrupt factor flag FIDMA (D4)/DMA interrupt factor flag register (0x40281) is set, thereby generating an interrupt request.

If the transfer counter is not 0 or DINTEN = "0" (interrupt disabled), the FIDMA flag is not set.

If the interrupt factor flag for the same channel is set during a software-triggered transfer, the IDMA invocation request by that interrupt factor flag is kept pending. However, the interrupt factor flag will be reset when the current execution is completed, so there will be no DMA transfer by the interrupt factor flag.

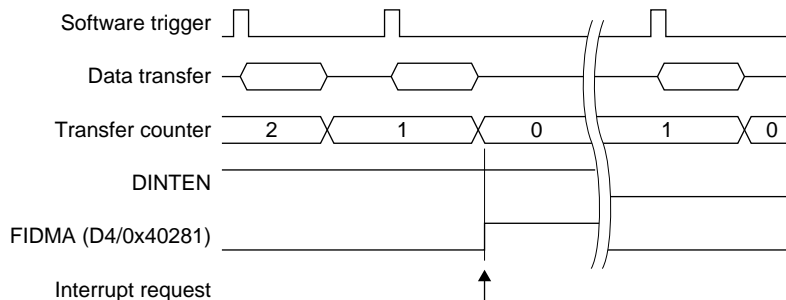


Figure V.3.5 Operation when Invoked by Software Trigger

## Linking

If the IDMA channel number to be executed next is set in the IDMA link field "LNKCHN" of control information and LNKEN is set to "1" (link enabled), DMA successive transfer in that IDMA channel can be performed. An example of link setting is shown in Figure V.3.6.

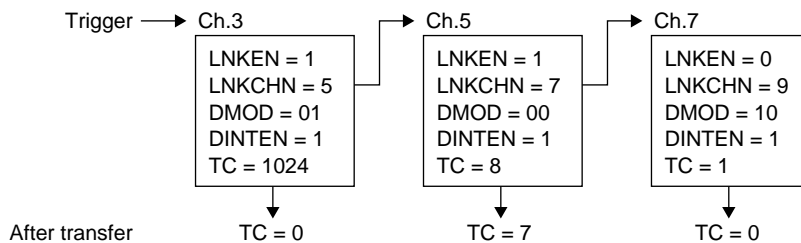


Figure V.3.6 Example of Link Setting

For the above example, IDMA operates as described below.

### • For trigger in hardware

- (1) The IDMA channel 3 is invoked by an interrupt factor and the DMA transfer that is set is performed. Since the IDMA is operating in successive transfer mode and the transfer counter is decremented to 0 and DINTEN is set to "1", the interrupt factor flag by which the channel 3 has been invoked remains set.
- (2) Next, a DMA transfer is performed via the linked IDMA channel 5. Channel 5 is set for single transfer mode and the transfer counter in this transfer is decremented by 1.
- (3) Finally, a DMA transfer in IDMA channel 7 is performed. Although the channel 7 is set for block transfer mode, the transfer counter is decremented to 0 when the transfer is completed because the number of transfers to be performed is 1.
- (4) Since the interrupt factor flag that has invoked IDMA channel 3 in (1) remains set, an interrupt is generated when the IDMA transfer (channel 7) in (3) is completed. The transfer result does not affect the interrupt factor flag of channel 3.  
To generate an interrupt at the end of an IDMA transfer, the DINTEN (end-of-transfer interrupt enable) bits in the IDMA control information for the first IDMA channel to be invoked and all the channels to be linked must be set to "1".

### • For trigger in the software application

- (1) The IDMA channel 3 is invoked by a trigger in the software application and the DMA transfer that is set is performed. Since the IDMA is operating in successive transfer mode and the transfer counter is decremented to 0 and DINTEN is set to "1", the IDMA interrupt factor flag FIDMA (D4)/DMA interrupt factor flag register (0x40281) is set when the transfer is completed.
- (2) Next, a DMA transfer is performed in the linked IDMA channel 5. The channel 5 is set for the single transfer mode and the transfer counter in this transfer is decremented by 1.
- (3) Finally, a DMA transfer in IDMA channel 7 is performed. Although channel 7 is set for the block transfer mode, the transfer counter is decremented to 0 when the transfer is completed because the number of transfers to be performed is 1. The completion of this transfer also causes the FIDMA flag to be set to "1". However, the FIDMA flag has already been set when the transfer is completed in (1) above.
- (4) Since the FIDMA flag is set, an interrupt request is generated here. In cases when IDMA has been invoked by a trigger in the software application, if the transfer counter in any one of the linked channels is decremented to 0 and DINTEN for that channel is set to "1", an interrupt request for the completion of IDMA transfer is generated when a transfer operation in each of the linked channels is completed. The channel in which an interrupt request has been generated can be verified by reading out the transfer counter.  
Transfer operations in each channel are performed as described earlier.

## Interrupt Function of Intelligent DMA

IDMA can generate an interrupt that causes invocation of IDMA and an interrupt for the completion of IDMA transfer itself.

### Interrupt when invoked by an interrupt factor

If the corresponding bits of the IDMA request and interrupt enable registers are left set (= "1"), assertion of an interrupt request is kept pending even when the enabled interrupt factor has occurred and the IDMA channel assigned to that interrupt factor is invoked.

If the transfer counter is decremented to 0 and DINTEN = "1" (interrupt enabled) when one DMA transfer is completed, the interrupt factor that has invoked IDMA is not reset and an interrupt request is generated. At the same time, the IDMA request register is cleared to "0". The IDMA enable bit is not cleared and remains set to "1".

If the transfer counter is not 0, the interrupt factor flag is reset when the DMA transfer is completed, so that no interrupt is generated. In this case, the IDMA request bit and IDMA enable bit are not cleared and remain set to "1". When DINTEN has been set to "0" (interrupt disabled), the interrupt factor flag is reset even if the transfer counter reaches 0, so that no interrupt is generated. In this case, the IDMA request bit is not cleared but the IDMA enable bit is cleared.

When IDMA is invoked by the software trigger, the IDMA interrupt factor flag FIDMA (D4)/DMA interrupt factor flag register (0x40281) will not be set.

For details about the interrupt factors that can be used to invoke IDMA and the interrupt control registers, refer to the descriptions of the peripheral circuits in this manual.

Note that the priority levels of interrupt factors are set by the interrupt priority register. Refer to "ITC (Interrupt Controller)". However, when compared between IDMA and interrupt requests, IDMA is given higher priority over the other. Consequently, even when an interrupt factor occurring during an IDMA transfer has higher priority than the interrupt factor that invoked the IDMA transfer, an interrupt request for it or a new IDMA invocation request is not accepted until after the current IDMA transfer is completed.

### Software-triggered interrupts

If the transfer counter is decremented to 0 and DINTEN = "1" (interrupt enabled) when one DMA transfer operation is completed, the IDMA interrupt factor flag FIDMA (D4)/DMA interrupt factor flag register (0x40281) is set, thereby generating an interrupt request. If the transfer counter is not 0 or DINTEN = "0" (interrupt disabled), the FIDMA flag is not set.

### IDMA interrupt control register in the interrupt controller

The following registers are used to control an interrupt for the completion of IDMA transfer:

Interrupt factor flag: FIDMA (D4) / DMA interrupt factor flag register (0x40281)

Interrupt enable: EIDMA (D4) / DMA interrupt enable register (0x40271)

Interrupt level: PDM[2:0] (D[2:0]) / IDMA interrupt priority register (0x40265)

When a DMA transfer in the IDMA channel invoked by a trigger in the software application or subsequent link is completed and the transfer counter is decremented to 0, the interrupt factor flag for the completion of IDMA transfer is set to "1". However, this requires as a precondition that interrupt be enabled (DINTEN = "1") in the control information for that channel. If the interrupt enable register bit remains set (= "1") when the flag is set, an interrupt request is generated. Interrupts can be disabled by leaving the interrupt enable register bit cleared (= "0"). Use the interrupt priority register to set interrupt priority levels (0 to 7). An interrupt request to the CPU is accepted on condition that no other interrupt request of higher priority is generated.

Furthermore, it is only when the PSR's IE bit = "1" (interrupt enabled) and the set value of IL is smaller than the IDMA interrupt level which is set by the interrupt priority register that the CPU actually accepts an IDMA interrupt request.

For details about these interrupt control registers, and for information on device operation when an interrupt occurs, refer to "ITC (Interrupt Controller)".

### Trap vector

The trap vector address for an interrupt upon completion of IDMA transfer by default is set to 0x0C00068.

The trap table base address can be changed using the TTBR registers (0x48134 to 0x48137).

## I/O Memory of Intelligent DMA

Table V.3.3 shows the control bits of IDMA.

Table V.3.3 Control Bits of IDMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
IDMA interrupt priority register	0040265 (B)	D7-3	–	reserved	–	–	–	0 when being read.				
		D2	PDM2	IDMA interrupt level	0 to 7	X	R/W					
		D1	PDM1			X						
		D0	PDM0			X						
DMA interrupt enable register	0040271 (B)	D7-5	–	reserved	–	–	–	0 when being read.				
		D4	EIDMA	IDMA	1 Enabled 0 Disabled	0	R/W					
		D3	EHDM3			0	R/W					
		D2	EHDM2			0	R/W					
		D1	EHDM1			0	R/W					
		D0	EHDM0			0	R/W					
DMA interrupt factor flag register	0040281 (B)	D7-5	–	reserved	–	–	–	0 when being read.				
		D4	FIDMA	IDMA	1 Factor is generated 0 No factor is generated	X	R/W					
		D3	FHDM3			X	R/W					
		D2	FHDM2			X	R/W					
		D1	FHDM1			X	R/W					
		D0	FHDM0			X	R/W					
IDMA base address low-order register	0048200 (HW)	DF	DBASEL15	IDMA base address		0	R/W					
		DE	DBASEL14	low-order 16 bits								
		DD	DBASEL13	(Initial value: 0x0C003A0)								
		DC	DBASEL12									
		DB	DBASEL11									
		DA	DBASEL10									
		D9	DBASEL9									
		D8	DBASEL8									
		D7	DBASEL7									
		D6	DBASEL6									
		D5	DBASEL5									
		D4	DBASEL4									
		D3	DBASEL3									
		D2	DBASEL2									
		D1	DBASEL1									
D0	DBASEL0											
IDMA base address high-order register	0048202 (HW)	DF-C	–	reserved	–	–	–	Undefined in read.				
		DB	DBASEH11	IDMA base address		0	R/W					
		DA	DBASEH10	high-order 12 bits								
		D9	DBASEH9	(Initial value: 0x0C003A0)								
		D8	DBASEH8									
		D7	DBASEH7									
		D6	DBASEH6									
		D5	DBASEH5									
		D4	DBASEH4									
		D3	DBASEH3									
		D2	DBASEH2									
		D1	DBASEH1									
		D0	DBASEH0									
		IDMA start register	0048204 (B)	D7		DSTART	IDMA start		1 IDMA start 0 Stop	0	R/W	
				D6-0		DCHN	IDMA channel number		0 to 127	0	R/W	
IDMA enable register	0048205 (B)	D7-1	–	reserved		–	–		–			
		D0	IDMAEN	IDMA enable	1 Enabled 0 Disabled	0	R/W					

---

**DBASEL[15:0]:** IDMA base address [15:0] (D[F:0]) / IDMA base address low-order register (0x48200)

**DBASEH[11:0]:** IDMA base address [27:16] (D[B:0]) / IDMA base address high-order register (0x48202)

---

Specify the starting address of the control information to be placed in RAM.

Use DBASEL to set the 16 low-order bits of the address and DBASEH to set the 12 high-order bits.

The address to be set in these registers must always be a word (32-bit) boundary address.

These registers cannot be read or written in bytes. The registers must be accessed in words for read/write operations to address 0x48200, and in half-words for read/write operations to addresses 0x48200 and 0x48202. Write operations in half-words must be performed in order of 0x48200 and 0x48202. Read operations in half-words may be performed in any order.

Write operations to the IDMA base address registers during a DMA transfer are ignored. When the register is read during a DMA transfer, the read data is indeterminate.

At initial reset, the base address is set to 0xC003A0.

---

**IDMAEN:** DMA enable (D0) / DMA enable register (0x48205)

---

Enable a IDMA transfer.

Write "1": Enabled

Write "0": Disabled

Read: Valid

A data transfer operation by intelligent DMA is enabled by writing "1" to IDMAEN.

IDMA transfer is disabled by writing "0" to IDMAEN.

At initial reset, IDMAEN is set to "0" (disabled).

---

**DCHN[6:0]:** IDMA channel number (D[6:0]) / IDMA start register (0x48204)

---

Set the channel numbers (0 to 127) to be invoked by a trigger in the software application.

At initial reset, DCHN is set to "0".

---

**DSTART:** IDMA start (D7) / IDMA start register (0x48204)

---

Use this register for a trigger in the software application and for monitoring the operation of IDMA.

#### When written

Write "1": IDMA started

Write "0": Invalid

#### When read

Read "1": IDMA operating (only when invoked by software trigger)

Read "0": IDMA inactive

When DSTART is set to "1", it functions as a trigger in the software application, invoking the IDMA channel that is set in the DCHN register.

At initial reset, DSTART is set to "0".

---

**PDM2-PDM0:** IDMA interrupt level (D[2:0]) / IDMA interrupt priority register (0x40265)

---

Set the priority level of the interrupt upon completion of IDMA transfer in the range of 0 to 7.

At initial reset, the contents of this register are indeterminate.

---

**EIDMA:** IDMA interrupt enable (D4) / DMA interrupt enable register (0x40271)

---

Enable or disable occurrence of an interrupt to the CPU.

Write "1": Interrupt enabled

Write "0": Interrupt disabled

Read: Valid

This bit controls the interrupt generated upon completion of IDMA transfer. The interrupt is enabled by setting this bit to "1" and disabled by setting this bit to "0".

At initial reset, EIDMA is set to "0" (interrupt disable).

---

**FIDMA:** IDMA interrupt factor flag (D4) / DMA interrupt factor flag register (0x40281)

---

Indicate the occurrence status of an IDMA interrupt request.

**When read**

- Read "1": Interrupt factor occurred
- Read "0": No interrupt factor occurred

**When written using reset-only method (default)**

- Write "1": Interrupt factor flag is reset
- Write "0": Invalid

**When written using the read/write method**

- Write "1": Interrupt factor flag is set
- Write "0": Interrupt factor flag is reset

This flag is set to "1" when one DMA transfer initiated by a software trigger or subsequent link is completed and the transfer counter is decremented to 0. However, this requires as a precondition that interrupts be enabled in control information (DINTEN = "1").

At this time, an interrupt to the CPU is generated if the following conditions are met:

1. The corresponding interrupt enable register bit is set to "1".
2. No interrupt request of higher priority is generated.
3. The IE bit of the PSR is set to "1" (interrupt enable).
4. The corresponding interrupt priority register is set to a level higher than the CPU's interrupt level (IL).

In order for the next interrupt to be accepted after interrupt generation, the interrupt factor flag must be reset and the PSR must be set up again (by setting the IL below the level indicated by the interrupt priority register and setting the IE bit to "1" or executing a reti instruction).

The interrupt factor flag can only be reset by a write instruction in the software application. If the PSR is set up again to accept interrupts (or the reti instruction is executed) without resetting the interrupt factor flag, the same interrupt may occur again. Note also that the value to be written to reset the flag is "1" when using the reset-only method (RSTONLY = "1") and "0" when using the read/write method (RSTONLY = "0"). Be careful not to confuse these two cases.

This flag becomes indeterminate when initially reset, so be sure to reset it in the software application.

## Programming Notes

- (1) Before setting the IDMA base address, be sure to disable DMA transfers (IDMAEN = "0"). Writing to the IDMA base address register is ignored when the DMA transfer is enabled (IDMAEN = "1"). Also, when the register is read during a DMA transfer, the data is indeterminate. When setting or rewriting control information for each channel, make sure that DMA transfers will not occur in any channel.
- (2) The address that is set in the IDMA base address register must always be a word (32-bit) boundary address.
- (3) When performing data transfer in block transfer mode, the block size must not be set to "0".
- (4) After an initial reset, the interrupt factor flag (FIDMA) becomes indeterminate. To prevent unwanted interrupts from occurring, be sure to reset the flag in a program.
- (5) Once an interrupt occurs, be sure to reset the interrupt factor flag (FIDMA) before setting up the PSR again or executing the reti instruction. This ensures that an interrupt will not be generated for the same factor.
- (6) In HALT mode, since the DMA and BCU clocks operate, if the next operation is performed in HALT mode, not HALT2 mode, with a setting of 0 in clock option register HLT2OP (D3/0x40190), that operation will be an unpredictable erroneous operation.  
If a DMA trigger occurs and DMA is invoked while the CPU is stopped after HALT mode execution, erroneous operation will result. Ensure that DMA is not invoked in HALT mode.  
In HALT2 mode, DMA is not invoked since the DMA and BCU clocks are stopped.
- (7) If a start-up request to an IDMA channel occurs while the channel is transferring data, the start-up request cannot be accepted (it is ignored) since the same interrupt factor is used. When starting IDMA transfer using the same channel successively, the trigger interval must be longer than  $\langle \text{IDMA control information read cycles} \rangle + \langle \text{IDMA transfer read/write cycles} \rangle + \langle \text{IDMA control information write cycles} \rangle$ .

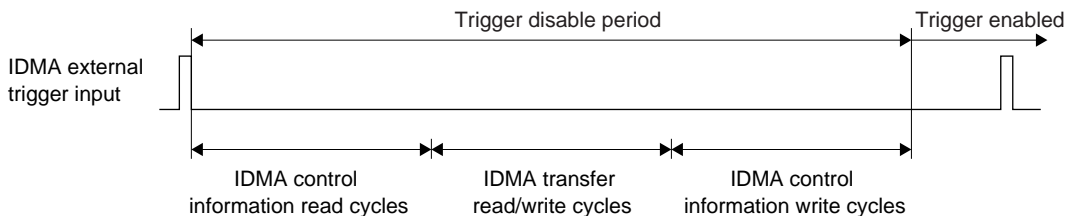


Figure V.3.7 External Trigger Input Interval

Note that the trigger disable period depends on the bus access conditions including wait setting for the memory area in which the control information is located as well as the data transfer conditions.

THIS PAGE IS BLANK.



**S1C33301 Technical Manual**  
**Appendix**



## I/O Map

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit timer 4/5 clock select register	0040140 (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	P8TPCK5	8-bit timer 5 clock selection	1 0/1	0 Divided clk.	0 R/W	0: selected by
		D0	P8TPCK4	8-bit timer 4 clock selection	1 0/1	0 Divided clk.	0 R/W	Prescaler clock select register (0x40181)
8-bit timer 4/5 clock control register	0040145 (B)	D7	P8TON5	8-bit timer 5 clock control	1 On	0 Off	0 R/W	
		D6	P8TS52	8-bit timer 5	1 1 1	0/256	0 R/W	0: selected by
		D5	P8TS51	clock division ratio selection	1 1 0	0/128	0 R/W	Prescaler clock select register (0x40181)
		D4	P8TS50	1 0 1	0/64	0 R/W		
				1 0 0	0/32			
				0 1 1	0/16		8-bit timer 5 can generate the clock for the serial I/F Ch.3.	
				0 1 0	0/8			
				0 0 1	0/4			
				0 0 0	0/2			
		D3	P8TON4	8-bit timer 4 clock control	1 On	0 Off	0 R/W	
		D2	P8TS42	8-bit timer 4	1 1 1	0/4096	0 R/W	0: selected by
		D1	P8TS41	clock division ratio selection	1 1 0	0/2048	0 R/W	Prescaler clock select register (0x40181)
		D0	P8TS40	1 0 1	0/64	0 R/W		
1 0 0	0/32							
0 1 1	0/16				8-bit timer 4 can generate the clock for the serial I/F Ch.2.			
0 1 0	0/8							
		0 0 1	0/4					
		0 0 0	0/2					
8-bit timer 0-3 clock select register	0040146 (B)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	P8TPCK3	8-bit timer 3 clock selection	1 0/1	0 Divided clk.	0 R/W	0: selected by
		D2	P8TPCK2	8-bit timer 2 clock selection	1 0/1	0 Divided clk.	0 R/W	Prescaler clock select register (0x40181)
		D1	P8TPCK1	8-bit timer 1 clock selection	1 0/1	0 Divided clk.	0 R/W	
		D0	P8TPCK0	8-bit timer 0 clock selection	1 0/1	0 Divided clk.	0 R/W	
16-bit timer 0 clock control register	0040147 (B)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	P16TON0	16-bit timer 0 clock control	1 On	0 Off	0 R/W	
		D2	P16TS02	16-bit timer 0	P16TS0[2:0]	Division ratio	0 R/W	0: selected by
		D1	P16TS01	1 1 1	0/4096	0	Prescaler clock select register (0x40181)	
				1 1 0	0/1024	0		
				1 0 1	0/256			
				1 0 0	0/64		16-bit timer 0 can be used as a watchdog timer.	
				0 1 1	0/16			
				0 1 0	0/4			
0 0 1	0/2							
D0	P16TS00	clock division ratio selection	1 1 1	0/4096	0			
		1 1 0	0/1024	0				
		1 0 1	0/256					
		1 0 0	0/64					
		0 1 1	0/16					
		0 1 0	0/4					
		0 0 1	0/2					
		0 0 0	0/1					
16-bit timer 1 clock control register	0040148 (B)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	P16TON1	16-bit timer 1 clock control	1 On	0 Off	0 R/W	
		D2	P16TS12	16-bit timer 1	P16TS1[2:0]	Division ratio	0 R/W	0: selected by
		D1	P16TS11	1 1 1	0/4096	0	Prescaler clock select register (0x40181)	
				1 1 0	0/1024	0		
				1 0 1	0/256			
				1 0 0	0/64			
				0 1 1	0/16			
				0 1 0	0/4			
0 0 1	0/2							
D0	P16TS10	clock division ratio selection	1 1 1	0/4096	0			
		1 1 0	0/1024	0				
		1 0 1	0/256					
		1 0 0	0/64					
		0 1 1	0/16					
		0 1 0	0/4					
		0 0 1	0/2					
		0 0 0	0/1					
16-bit timer 2 clock control register	0040149 (B)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	P16TON2	16-bit timer 2 clock control	1 On	0 Off	0 R/W	
		D2	P16TS22	16-bit timer 2	P16TS2[2:0]	Division ratio	0 R/W	0: selected by
		D1	P16TS21	1 1 1	0/4096	0	Prescaler clock select register (0x40181)	
				1 1 0	0/1024	0		
				1 0 1	0/256			
				1 0 0	0/64			
				0 1 1	0/16			
0 1 0	0/4							
D0	P16TS20	clock division ratio selection	1 1 1	0/4096	0			
		1 1 0	0/1024	0				
		1 0 1	0/256					
		1 0 0	0/64					
		0 1 1	0/16					
		0 1 0	0/4					
		0 0 1	0/2					
		0 0 0	0/1					

(B) in [Address] indicates an 8-bit register and (HW) indicates a 16-bit register.

The meaning of the symbols described in [Init.] are listed below:

0, 1: Initial values that are set at initial reset. (However, the registers for the bus and input/output ports are not initialized at hot start.)

X: Not initialized at initial reset.

–: Not set in the circuit.

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 3 clock control register	004014A (B)	D7-4	–	reserved		–	–	0 when being read.	
		D3	P16TON3	16-bit timer 3 clock control	1 On	0 Off	0	R/W	
		D2	P16TS32	16-bit timer 3	P16TS3[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P16TS31	clock division ratio selection	1 1 1	θ/4096	0		
		D0	P16TS30	1 1 0	θ/1024	0			
				1 0 1	θ/256	0			
				1 0 0	θ/64	0			
				0 1 1	θ/16	0			
				0 1 0	θ/4	0			
				0 0 1	θ/2	0			
0 0 0	θ/1	0							
16-bit timer 4 clock control register	004014B (B)	D7-4	–	reserved		–	–	0 when being read.	
		D3	P16TON4	16-bit timer 4 clock control	1 On	0 Off	0	R/W	
		D2	P16TS42	16-bit timer 4	P16TS4[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P16TS41	clock division ratio selection	1 1 1	θ/4096	0		
		D0	P16TS40	1 1 0	θ/1024	0			
				1 0 1	θ/256	0			
				1 0 0	θ/64	0			
				0 1 1	θ/16	0			
				0 1 0	θ/4	0			
				0 0 1	θ/2	0			
0 0 0	θ/1	0							
16-bit timer 5 clock control register	004014C (B)	D7-4	–	reserved		–	–	0 when being read.	
		D3	P16TON5	16-bit timer 5 clock control	1 On	0 Off	0	R/W	
		D2	P16TS52	16-bit timer 5	P16TS5[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P16TS51	clock division ratio selection	1 1 1	θ/4096	0		
		D0	P16TS50	1 1 0	θ/1024	0			
				1 0 1	θ/256	0			
				1 0 0	θ/64	0			
				0 1 1	θ/16	0			
				0 1 0	θ/4	0			
				0 0 1	θ/2	0			
0 0 0	θ/1	0							
8-bit timer 0/1 clock control register	004014D (B)	D7	P8TON1	8-bit timer 1 clock control	1 On	0 Off	0	R/W	
		D6	P8TS12	8-bit timer 1	P8TS1[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D5	P8TS11	clock division ratio selection	1 1 1	θ/4096	0		
		D4	P8TS10	1 1 0	θ/2048	0			
				1 0 1	θ/1024	0			
				1 0 0	θ/512	0			
				0 1 1	θ/256	0			
				0 1 0	θ/128	0			
				0 0 1	θ/64	0			
		0 0 0	θ/32	0					
		D3	P8TON0	8-bit timer 0 clock control	1 On	0 Off	0	R/W	
		D2	P8TS02	8-bit timer 0	P8TS0[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P8TS01	clock division ratio selection	1 1 1	θ/256	0		
		D0	P8TS00	1 1 0	θ/128	0			
1 0 1	θ/64			0					
1 0 0	θ/32			0					
0 1 1	θ/16			0					
0 1 0	θ/8	0							
0 0 1	θ/4	0							
0 0 0	θ/2	0							

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
8-bit timer 2/3 clock control register	004014E (B)	D7	P8TON3	8-bit timer 3 clock control	1 On	0 Off	0	R/W			
		D6	P8TS32	8-bit timer 3 clock division ratio selection	P8TS3[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)  8-bit timer 3 can generate the clock for the serial I/F Ch.1.		
		D5	P8TS31		1 1 1	θ/256	0				
		D4	P8TS30		1 1 0	θ/128	0				
					1 0 1	θ/64					
					1 0 0	θ/32					
					0 1 1	θ/16					
					0 1 0	θ/8					
					0 0 1	θ/4					
				0 0 0	θ/2						
				D3	P8TON2	8-bit timer 2 clock control	1 On	0 Off	0	R/W	
				D2	P8TS22	8-bit timer 2 clock division ratio selection	P8TS2[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)  8-bit timer 2 can generate the clock for the serial I/F Ch.0.
		D1	P8TS21	1 1 1	θ/4096		0				
		D0	P8TS20	1 1 0	θ/2048		0				
				1 0 1	θ/64						
				1 0 0	θ/32						
				0 1 1	θ/16						
				0 1 0	θ/8						
				0 0 1	θ/4						
				0 0 0	θ/2						
A/D clock control register	004014F (B)	D7-4	–	reserved	–		–	–	0 when being read.		
		D3	PSONAD	A/D converter clock control	1 On	0 Off	0	R/W			
		D2	PSAD2	A/D converter clock division ratio selection	PSAD[2:0] Division ratio		0	R/W	θ: selected by Prescaler clock select register (0x40181)		
		D1	PSAD1		1 1 1	θ/256	0				
		D0	PSAD0		1 1 0	θ/128	0				
					1 0 1	θ/64					
					1 0 0	θ/32					
					0 1 1	θ/16					
		0 1 0	θ/8								
		0 0 1	θ/4								
		0 0 0	θ/2								
Clock timer Run/Stop register	0040151 (B)	D7-2	–	reserved	–		–	–	0 when being read.		
		D1	TCRST	Clock timer reset	1 Reset	0 Invalid	X	W			
		D0	TCRUN	Clock timer Run/Stop control	1 Run	0 Stop	X	R/W			
Clock timer interrupt control register	0040152 (B)	D7	TCISE2	Clock timer interrupt factor selection	TCISE[2:0] Interrupt factor		X	R/W			
		D6	TCISE1		1 1 1	None	X				
		D5	TCISE0		1 1 0	Day	X				
					1 0 1	Hour					
					1 0 0	Minute					
					0 1 1	1 Hz					
					0 1 0	2 Hz					
				0 0 1	8 Hz						
				0 0 0	32 Hz						
				D4	TCASE2	Clock timer alarm factor selection	TCASE[2:0] Alarm factor		X	R/W	
D3	TCASE1	1 X X	Day	X							
D2	TCASE0	X 1 X	Hour	X							
		X X 1	Minute								
		0 0 0	None								
		D1	TCIF	Interrupt factor generation flag	1 Generated	0 Not generated	X	R/W	Reset by writing 1.		
		D0	TCAF	Alarm factor generation flag	1 Generated	0 Not generated	X	R/W	Reset by writing 1.		
Clock timer divider register	0040153 (B)	D7	TCD7	Clock timer data 1 Hz	1 High	0 Low	X	R			
		D6	TCD6	Clock timer data 2 Hz	1 High	0 Low	X	R			
		D5	TCD5	Clock timer data 4 Hz	1 High	0 Low	X	R			
		D4	TCD4	Clock timer data 8 Hz	1 High	0 Low	X	R			
		D3	TCD3	Clock timer data 16 Hz	1 High	0 Low	X	R			
		D2	TCD2	Clock timer data 32 Hz	1 High	0 Low	X	R			
		D1	TCD1	Clock timer data 64 Hz	1 High	0 Low	X	R			
		D0	TCD0	Clock timer data 128 Hz	1 High	0 Low	X	R			
Clock timer second register	0040154 (B)	D7-6	–	reserved	–		–	–	0 when being read.		
		D5	TCMD5	Clock timer second counter data	0 to 59 seconds		X	R			
		D4	TCMD4	TCMD5 = MSB			X				
		D3	TCMD3	TCMD0 = LSB			X				
		D2	TCMD2				X				
		D1	TCMD1				X				
		D0	TCMD0				X				

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock timer minute register	0040155 (B)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	TCHD5	Clock timer minute counter data	0 to 59 minutes	X	R/W	
		D4	TCHD4	TCHD5 = MSB		X		
		D3	TCHD3	TCHD0 = LSB		X		
		D2	TCHD2			X		
		D1	TCHD1			X		
		D0	TCHD0			X		
Clock timer hour register	0040156 (B)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	TCDD4	Clock timer hour counter data	0 to 23 hours	X	R/W	
		D3	TCDD3	TCDD4 = MSB		X		
		D2	TCDD2	TCDD0 = LSB		X		
		D1	TCDD1			X		
		D0	TCDD0			X		
Clock timer day (low-order) register	0040157 (B)	D7	TCND7	Clock timer day counter data	0 to 65535 days	X	R/W	
		D6	TCND6	(low-order 8 bits)	(low-order 8 bits)	X		
		D5	TCND5	TCND0 = LSB		X		
		D4	TCND4			X		
		D3	TCND3			X		
		D2	TCND2			X		
		D1	TCND1			X		
		D0	TCND0			X		
Clock timer day (high-order) register	0040158 (B)	D7	TCND15	Clock timer day counter data	0 to 65535 days	X	R/W	
		D6	TCND14	(high-order 8 bits)	(high-order 8 bits)	X		
		D5	TCND13	TCND15 = MSB		X		
		D4	TCND12			X		
		D3	TCND11			X		
		D2	TCND10			X		
		D1	TCND9			X		
		D0	TCND8			X		
Clock timer minute comparison register	0040159 (B)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	TCCH5	Clock timer minute comparison data	0 to 59 minutes	X	R/W	
		D4	TCCH4	(Note) Can be set within 0–63.		X		
		D3	TCCH3	TCCH5 = MSB		X		
		D2	TCCH2	TCCH0 = LSB		X		
		D1	TCCH1			X		
		D0	TCCH0			X		
Clock timer hour comparison register	004015A (B)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	TCCD4	Clock timer hour comparison data	0 to 23 hours	X	R/W	
		D3	TCCD3	TCCD4 = MSB	(Note) Can be set within 0–31.	X		
		D2	TCCD2	TCCD0 = LSB		X		
		D1	TCCD1			X		
		D0	TCCD0			X		
Clock timer day comparison register	004015B (B)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	TCCN4	Clock timer day comparison data	0 to 31 days	X	R/W	Compared with TCND[4:0].
		D3	TCCN3	TCCN4 = MSB		X		
		D2	TCCN2	TCCN0 = LSB		X		
		D1	TCCN1			X		
		D0	TCCN0			X		
8-bit timer 0 control register	0040160 (B)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	PTOUT0	8-bit timer 0 clock output control	1 On 0 Off	0	R/W	
		D1	PSET0	8-bit timer 0 preset	1 Preset 0 Invalid	–	W	0 when being read.
		D0	PTRUN0	8-bit timer 0 Run/Stop control	1 Run 0 Stop	0	R/W	
8-bit timer 0 reload data register	0040161 (B)	D7	RLD07	8-bit timer 0 reload data	0 to 255	X	R/W	
		D6	RLD06	RLD07 = MSB		X		
		D5	RLD05	RLD00 = LSB		X		
		D4	RLD04			X		
		D3	RLD03			X		
		D2	RLD02			X		
		D1	RLD01			X		
		D0	RLD00			X		
8-bit timer 0 counter data register	0040162 (B)	D7	PTD07	8-bit timer 0 counter data	0 to 255	X	R	
		D6	PTD06	PTD07 = MSB		X		
		D5	PTD05	PTD00 = LSB		X		
		D4	PTD04			X		
		D3	PTD03			X		
		D2	PTD02			X		
		D1	PTD01			X		
		D0	PTD00			X		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
8-bit timer 1 control register	0040164 (B)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	PTOUT1	8-bit timer 1 clock output control	1 On 0 Off	0	R/W		
		D1	PSET1	8-bit timer 1 preset	1 Preset 0 Invalid	–	W	0 when being read.	
		D0	PTRUN1	8-bit timer 1 Run/Stop control	1 Run 0 Stop	0	R/W		
8-bit timer 1 reload data register	0040165 (B)	D7	RLD17	8-bit timer 1 reload data	0 to 255		X	R/W	
		D6	RLD16	RLD17 = MSB			X		
		D5	RLD15	RLD10 = LSB			X		
		D4	RLD14				X		
		D3	RLD13				X		
		D2	RLD12				X		
		D1	RLD11				X		
		D0	RLD10				X		
8-bit timer 1 counter data register	0040166 (B)	D7	PTD17	8-bit timer 1 counter data	0 to 255		X	R	
		D6	PTD16	PTD17 = MSB			X		
		D5	PTD15	PTD10 = LSB			X		
		D4	PTD14				X		
		D3	PTD13				X		
		D2	PTD12				X		
		D1	PTD11				X		
		D0	PTD10				X		
8-bit timer 2 control register	0040168 (B)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	PTOUT2	8-bit timer 2 clock output control	1 On 0 Off	0	R/W		
		D1	PSET2	8-bit timer 2 preset	1 Preset 0 Invalid	–	W	0 when being read.	
		D0	PTRUN2	8-bit timer 2 Run/Stop control	1 Run 0 Stop	0	R/W		
8-bit timer 2 reload data register	0040169 (B)	D7	RLD27	8-bit timer 2 reload data	0 to 255		X	R/W	
		D6	RLD26	RLD27 = MSB			X		
		D5	RLD25	RLD20 = LSB			X		
		D4	RLD24				X		
		D3	RLD23				X		
		D2	RLD22				X		
		D1	RLD21				X		
		D0	RLD20				X		
8-bit timer 2 counter data register	004016A (B)	D7	PTD27	8-bit timer 2 counter data	0 to 255		X	R	
		D6	PTD26	PTD27 = MSB			X		
		D5	PTD25	PTD20 = LSB			X		
		D4	PTD24				X		
		D3	PTD23				X		
		D2	PTD22				X		
		D1	PTD21				X		
		D0	PTD20				X		
8-bit timer 3 control register	004016C (B)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	PTOUT3	8-bit timer 3 clock output control	1 On 0 Off	0	R/W		
		D1	PSET3	8-bit timer 3 preset	1 Preset 0 Invalid	–	W	0 when being read.	
		D0	PTRUN3	8-bit timer 3 Run/Stop control	1 Run 0 Stop	0	R/W		
8-bit timer 3 reload data register	004016D (B)	D7	RLD37	8-bit timer 3 reload data	0 to 255		X	R/W	
		D6	RLD36	RLD37 = MSB			X		
		D5	RLD35	RLD30 = LSB			X		
		D4	RLD34				X		
		D3	RLD33				X		
		D2	RLD32				X		
		D1	RLD31				X		
		D0	RLD30				X		
8-bit timer 3 counter data register	004016E (B)	D7	PTD37	8-bit timer 3 counter data	0 to 255		X	R	
		D6	PTD36	PTD37 = MSB			X		
		D5	PTD35	PTD30 = LSB			X		
		D4	PTD34				X		
		D3	PTD33				X		
		D2	PTD32				X		
		D1	PTD31				X		
		D0	PTD30				X		
Watchdog timer write-protect register	0040170 (B)	D7	WRWD	EWD write protection	1 Write enabled 0 Write-protect	0	R/W		
		D6-0	–	–	–	–	–	0 when being read.	
Watchdog timer enable register	0040171 (B)	D7-2	–	–	–	–	–	0 when being read.	
		D1	EWD	Watchdog timer enable	1 NMI enabled 0 NMI disabled	0	R/W		
		D0	–	–	–	–	–	0 when being read.	

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
8-bit timer 4 control register	0040174 (B)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	PTOUT4	8-bit timer 4 clock output control	1 On   0 Off	0	R/W		
		D1	PSET4	8-bit timer 4 preset	1 Preset   0 Invalid	–	W	0 when being read.	
		D0	PTRUN4	8-bit timer 4 Run/Stop control	1 Run   0 Stop	0	R/W		
8-bit timer 4 reload data register	0040175 (B)	D7	RLD47	8-bit timer 4 reload data RLD47 = MSB RLD40 = LSB	0 to 255		X	R/W	
		D6	RLD46		X				
		D5	RLD45		X				
		D4	RLD44		X				
		D3	RLD43		X				
		D2	RLD42		X				
		D1	RLD41		X				
		D0	RLD40		X				
8-bit timer 4 counter data register	0040176 (B)	D7	PTD47	8-bit timer 4 counter data PTD47 = MSB PTD40 = LSB	0 to 255		X	R	
		D6	PTD46		X				
		D5	PTD45		X				
		D4	PTD44		X				
		D3	PTD43		X				
		D2	PTD42		X				
		D1	PTD41		X				
		D0	PTD40		X				
8-bit timer 5 control register	0040178 (B)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	PTOUT5	8-bit timer 5 clock output control	1 On   0 Off	0	R/W		
		D1	PSET5	8-bit timer 5 preset	1 Preset   0 Invalid	–	W	0 when being read.	
		D0	PTRUN5	8-bit timer 5 Run/Stop control	1 Run   0 Stop	0	R/W		
8-bit timer 5 reload data register	0040179 (B)	D7	RLD57	8-bit timer 5 reload data RLD57 = MSB RLD50 = LSB	0 to 255		X	R/W	
		D6	RLD56		X				
		D5	RLD55		X				
		D4	RLD54		X				
		D3	RLD53		X				
		D2	RLD52		X				
		D1	RLD51		X				
		D0	RLD50		X				
8-bit timer 5 counter data register	004017A (B)	D7	PTD57	8-bit timer 5 counter data PTD57 = MSB PTD50 = LSB	0 to 255		X	R	
		D6	PTD56		X				
		D5	PTD55		X				
		D4	PTD54		X				
		D3	PTD53		X				
		D2	PTD52		X				
		D1	PTD51		X				
		D0	PTD50		X				
Power control register	0040180 (B)	D7	CLKDT1	System clock division ratio selection	CLKDIT[1:0]		0	R/W	
		D6	CLKDT0		1   1	1/8			
					1   0	1/4			
					0   1	1/2			
				0   0	1/1				
		D5	PSCON	Prescaler On/Off control	1 On   0 Off	1	R/W		
		D4-3	–	reserved	–	0	–	Writing 1 not allowed.	
		D2	CLKCHG	CPU operating clock switch	1 OSC3   0 OSC1	1	R/W		
D1	SOSC3	High-speed (OSC3) oscillation On/Off	1 On   0 Off	1	R/W				
D0	SOSC1	Low-speed (OSC1) oscillation On/Off	1 On   0 Off	1	R/W				
Prescaler clock select register	0040181 (B)	D7-1	–	reserved	–	0	–		
		D0	PSCDT0	Prescaler clock selection	1 OSC1   0 OSC3/PLL	0	R/W		
Clock option register	0040190 (B)	D7-4	–	–	–	–	–	0 when being read.	
		D3	HLT2OP	HALT clock option	1 On   0 Off	0	R/W		
		D2	8T1ON	OSC3-stabilize waiting function	1 Off   0 On	1	R/W		
		D1	–	reserved	–	0	–	Do not write 1.	
	D0	PF1ON	OSC1 external output control	1 On   0 Off	0	R/W			
Power control protect register	004019E (B)	D7	CLGP7	Power control register protect flag Writing 10010110 (0x96) removes the write protection of the power control register (0x40180) and the clock option register (0x40190). Writing another value set the write protection.	Writing 10010110 (0x96)		0	R/W	
		D6	CLGP6		0				
		D5	CLGP5		0				
		D4	CLGP4		0				
		D3	CLGP3		0				
		D2	CLGP2		0				
		D1	CLGP1		0				
		D0	CLGP0		0				



Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Serial I/F Ch.0 transmit data register	00401E0 (B)	D7	TXD07	Serial I/F Ch.0 transmit data TXD07(06) = MSB TXD00 = LSB	0x0 to 0xFF(0x7F)		X	R/W	7-bit asynchronous mode does not use TXD07.		
		D6	TXD06								
		D5	TXD05								
		D4	TXD04								
		D3	TXD03								
		D2	TXD02								
		D1	TXD01								
		D0	TXD00								
Serial I/F Ch.0 receive data register	00401E1 (B)	D7	RXD07	Serial I/F Ch.0 receive data RXD07(06) = MSB RXD00 = LSB	0x0 to 0xFF(0x7F)		X	R	7-bit asynchronous mode does not use RXD07 (fixed at 0).		
		D6	RXD06								
		D5	RXD05								
		D4	RXD04								
		D3	RXD03								
		D2	RXD02								
		D1	RXD01								
		D0	RXD00								
Serial I/F Ch.0 status register	00401E2 (B)	D7-6	–	–	–		–	–	0 when being read.		
		D5	TEND0	Ch.0 transmit-completion flag	1	Transmitting	0	End	0	R	
		D4	FER0	Ch.0 flaming error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	PER0	Ch.0 parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D2	OER0	Ch.0 overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D1	TDBE0	Ch.0 transmit data buffer empty	1	Empty	0	Buffer full	1	R	
		D0	RDBF0	Ch.0 receive data buffer full	1	Buffer full	0	Empty	0	R	
Serial I/F Ch.0 control register	00401E3 (B)	D7	TXEN0	Ch.0 transmit enable	1	Enabled	0	Disabled	0	R/W	
		D6	RXEN0	Ch.0 receive enable	1	Enabled	0	Disabled	0	R/W	
		D5	EPR0	Ch.0 parity enable	1	With parity	0	No parity	X	R/W	Valid only in asynchronous mode.
		D4	PMD0	Ch.0 parity mode selection	1	Odd	0	Even	X	R/W	
		D3	STPB0	Ch.0 stop bit selection	1	2 bits	0	1 bit	X	R/W	
		D2	SCLK0	Ch.0 input clock selection	1	#SCLK0	0	Internal clock	X	R/W	
		D1	SMD01	Ch.0 transfer mode selection	SMD0[1:0]		Transfer mode		X	R/W	
		D0	SMD00		1	1	8-bit asynchronous	X			
					1	0	7-bit asynchronous				
		0	1		Clock sync. Slave						
				0	0	Clock sync. Master					
Serial I/F Ch.0 IrDA register	00401E4 (B)	D7-5	–	–	–		–	–	0 when being read.		
		D4	DIVMD0	Ch.0 async. clock division ratio	1	1/8	0	1/16	X	R/W	
		D3	IRTL0	Ch.0 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.
		D2	IRRL0	Ch.0 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W	
		D1	IRMD01	Ch.0 interface mode selection	IRMD0[1:0]		I/F mode		X	R/W	
		D0	IRMD00		1	1	reserved	X			
					1	0	IrDA 1.0				
				0	1	reserved					
				0	0	General I/F					
Serial I/F Ch.1 transmit data register	00401E5 (B)	D7	TXD17	Serial I/F Ch.1 transmit data TXD17(16) = MSB TXD10 = LSB	0x0 to 0xFF(0x7F)		X	R/W	7-bit asynchronous mode does not use TXD17.		
		D6	TXD16								
		D5	TXD15								
		D4	TXD14								
		D3	TXD13								
		D2	TXD12								
		D1	TXD11								
		D0	TXD10								
Serial I/F Ch.1 receive data register	00401E6 (B)	D7	RXD17	Serial I/F Ch.1 receive data RXD17(16) = MSB RXD10 = LSB	0x0 to 0xFF(0x7F)		X	R	7-bit asynchronous mode does not use RXD17 (fixed at 0).		
		D6	RXD16								
		D5	RXD15								
		D4	RXD14								
		D3	RXD13								
		D2	RXD12								
		D1	RXD11								
		D0	RXD10								
Serial I/F Ch.1 status register	00401E7 (B)	D7-6	–	–	–		–	–	0 when being read.		
		D5	TEND1	Ch.1 transmit-completion flag	1	Transmitting	0	End	0	R	
		D4	FER1	Ch.1 flaming error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	PER1	Ch.1 parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D2	OER1	Ch.1 overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D1	TDBE1	Ch.1 transmit data buffer empty	1	Empty	0	Buffer full	1	R	
		D0	RDBF1	Ch.1 receive data buffer full	1	Buffer full	0	Empty	0	R	

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Serial I/F Ch.1 control register	00401E8 (B)	D7	<b>TXEN1</b>	Ch.1 transmit enable	1 Enabled	0 Disabled	0	R/W		
		D6	<b>RXEN1</b>	Ch.1 receive enable	1 Enabled	0 Disabled	0	R/W		
		D5	<b>EPR1</b>	Ch.1 parity enable	1 With parity	0 No parity	X	R/W	Valid only in asynchronous mode.	
		D4	<b>PMD1</b>	Ch.1 parity mode selection	1 Odd	0 Even	X	R/W		
		D3	<b>STPB1</b>	Ch.1 stop bit selection	1 2 bits	0 1 bit	X	R/W		
		D2	<b>SSCK1</b>	Ch.1 input clock selection	1 #SCLK1	0 Internal clock	X	R/W		
		D1	<b>SMD11</b>	Ch.1 transfer mode selection		SMD1[1:0]		Transfer mode	X	R/W
		D0	<b>SMD10</b>			1	1	8-bit asynchronous	X	
		1	0			7-bit asynchronous				
		0	1			Clock sync. Slave				
				0	0	Clock sync. Master				
Serial I/F Ch.1 IrDA register	00401E9 (B)	D7-5	–	–	–		–	–	0 when being read.	
		D4	<b>DIVMD1</b>	Ch.1 async. clock division ratio	1 1/8	0 1/16	X	R/W		
		D3	<b>IRTL1</b>	Ch.1 IrDA I/F output logic inversion	1 Inverted	0 Direct	X	R/W	Valid only in asynchronous mode.	
		D2	<b>IRRL1</b>	Ch.1 IrDA I/F input logic inversion	1 Inverted	0 Direct	X	R/W		
		D1	<b>IRMD11</b>	Ch.1 interface mode selection		IRMD1[1:0]		I/F mode	X	R/W
D0	<b>IRMD10</b>	1	1			reserved	X			
				1	0	IrDA 1.0				
				0	1	reserved				
				0	0	General I/F				
Serial I/F Ch.2 transmit data register	00401F0 (B)	D7	<b>TXD27</b>	Serial I/F Ch.2 transmit data TXD27(26) = MSB TXD20 = LSB	0x0 to 0xFF(0x7F)			X	R/W	
		D6	<b>TXD26</b>		X					
		D5	<b>TXD25</b>		X					
		D4	<b>TXD24</b>		X					
		D3	<b>TXD23</b>		X					
		D2	<b>TXD22</b>		X					
		D1	<b>TXD21</b>		X					
		D0	<b>TXD20</b>		X					
Serial I/F Ch.2 receive data register	00401F1 (B)	D7	<b>RXD27</b>	Serial I/F Ch.2 receive data RXD27(26) = MSB RXD20 = LSB	0x0 to 0xFF(0x7F)			X	R	
		D6	<b>RXD26</b>		X					
		D5	<b>RXD25</b>		X					
		D4	<b>RXD24</b>		X					
		D3	<b>RXD23</b>		X					
		D2	<b>RXD22</b>		X					
		D1	<b>RXD21</b>		X					
		D0	<b>RXD20</b>		X					
Serial I/F Ch.2 status register	00401F2 (B)	D7-6	–	reserved	–		–	–	0 when being read.	
		D5	<b>TEND2</b>	Ch.2 transmit-completion flag	1 Transmitting	0 End	0	R		
		D4	<b>FER2</b>	Ch.2 flaming error flag	1 Error	0 Normal	0	R/W	Reset by writing 0.	
		D3	<b>PER2</b>	Ch.2 parity error flag	1 Error	0 Normal	0	R/W	Reset by writing 0.	
		D2	<b>OER2</b>	Ch.2 overrun error flag	1 Error	0 Normal	0	R/W	Reset by writing 0.	
		D1	<b>TDBE2</b>	Ch.2 transmit data buffer empty	1 Empty	0 Buffer full	1	R		
		D0	<b>RDBF2</b>	Ch.2 receive data buffer full	1 Buffer full	0 Empty	0	R		
Serial I/F Ch.2 control register	00401F3 (B)	D7	<b>TXEN2</b>	Ch.2 transmit enable	1 Enabled	0 Disabled	0	R/W		
		D6	<b>RXEN2</b>	Ch.2 receive enable	1 Enabled	0 Disabled	0	R/W		
		D5	<b>EPR2</b>	Ch.2 parity enable	1 With parity	0 No parity	X	R/W	Valid only in asynchronous mode.	
		D4	<b>PMD2</b>	Ch.2 parity mode selection	1 Odd	0 Even	X	R/W		
		D3	<b>STPB2</b>	Ch.2 stop bit selection	1 2 bits	0 1 bit	X	R/W		
		D2	<b>SSCK2</b>	Ch.2 input clock selection	1 #SCLK2	0 Internal clock	X	R/W		
		D1	<b>SMD21</b>	Ch.2 transfer mode selection		SMD2[1:0]		Transfer mode	X	R/W
		D0	<b>SMD20</b>			1	1	8-bit asynchronous	X	
		1	0			7-bit asynchronous				
		0	1			Clock sync. Slave				
				0	0	Clock sync. Master				
Serial I/F Ch.2 IrDA register	00401F4 (B)	D7-5	–	reserved	–		–	–	0 when being read.	
		D4	<b>DIVMD2</b>	Ch.2 async. clock division ratio	1 1/8	0 1/16	X	R/W		
		D3	<b>IRTL2</b>	Ch.2 IrDA I/F output logic inversion	1 Inverted	0 Direct	X	R/W	Valid only in asynchronous mode.	
		D2	<b>IRRL2</b>	Ch.2 IrDA I/F input logic inversion	1 Inverted	0 Direct	X	R/W		
		D1	<b>IRMD21</b>	Ch.2 interface mode selection		IRMD2[1:0]		I/F mode	X	R/W
D0	<b>IRMD20</b>	1	1			reserved	X			
				1	0	IrDA 1.0				
				0	1	reserved				
				0	0	General I/F				

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Serial I/F Ch.3 transmit data register	00401F5 (B)	D7	TXD37	Serial I/F Ch.3 transmit data	0x0 to 0xFF(0x7F)		X	R/W	
		D6	TXD36	TXD37(36) = MSB			X		
		D5	TXD35	TXD30 = LSB			X		
		D4	TXD34				X		
		D3	TXD33				X		
		D2	TXD32				X		
		D1	TXD31				X		
		D0	TXD30				X		
		Serial I/F Ch.3 receive data register	00401F6 (B)	D7	RXD37	Serial I/F Ch.3 receive data	0x0 to 0xFF(0x7F)		X
D6	RXD36			RXD37(36) = MSB			X		
D5	RXD35			RXD30 = LSB			X		
D4	RXD34						X		
D3	RXD33						X		
D2	RXD32						X		
D1	RXD31						X		
D0	RXD30						X		
Serial I/F Ch.3 status register	00401F7 (B)			D7-6	–	reserved	–		–
		D5	TEND3	Ch.3 transmit-completion flag	1 Transmitting	0 End	0	R	
		D4	FER3	Ch.3 flaming error flag	1 Error	0 Normal	0	R/W	Reset by writing 0.
		D3	PER3	Ch.3 parity error flag	1 Error	0 Normal	0	R/W	Reset by writing 0.
		D2	OER3	Ch.3 overrun error flag	1 Error	0 Normal	0	R/W	Reset by writing 0.
		D1	TDBE3	Ch.3 transmit data buffer empty	1 Empty	0 Buffer full	1	R	
		D0	RDBF3	Ch.3 receive data buffer full	1 Buffer full	0 Empty	0	R	
		Serial I/F Ch.3 control register	00401F8 (B)	D7	TXEN3	Ch.3 transmit enable	1 Enabled	0 Disabled	0
D6	RXEN3			Ch.3 receive enable	1 Enabled	0 Disabled	0	R/W	
D5	EPR3			Ch.3 parity enable	1 With parity	0 No parity	X	R/W	Valid only in asynchronous mode.
D4	PMD3			Ch.3 parity mode selection	1 Odd	0 Even	X	R/W	
D3	STPB3			Ch.3 stop bit selection	1 2 bits	0 1 bit	X	R/W	
D2	SSCK3			Ch.3 input clock selection	1 #SCLK3	0 Internal clock	X	R/W	
D1	SMD31 SMD30			Ch.3 transfer mode selection	SMD3[1:0] Transfer mode		X	R/W	
D0					1 1	8-bit asynchronous	X		
1 0					7-bit asynchronous				
0 1		Clock sync. Slave							
0 0	Clock sync. Master								
Serial I/F Ch.3 IrDA register	00401F9 (B)	D7-5	–	reserved	–		–	–	0 when being read.
		D4	DIVMD3	Ch.3 async. clock division ratio	1 1/8	0 1/16	X	R/W	
		D3	IRTL3	Ch.3 IrDA I/F output logic inversion	1 Inverted	0 Direct	X	R/W	Valid only in asynchronous mode.
		D2	IRRL3	Ch.3 IrDA I/F input logic inversion	1 Inverted	0 Direct	X	R/W	
		D1	IRMD31 IRMD30	Ch.3 interface mode selection	IRMD3[1:0] I/F mode		X	R/W	
		D0			1 1	reserved	X		
		1 0			IrDA 1.0				
		0 1			reserved				
0 0	General I/F								
A/D conversion result (low-order) register	0040240 (B)	D7	ADD7	A/D converted data (low-order 8 bits) ADD0 = LSB	0x0 to 0x3FF (low-order 8 bits)		0	R	
		D6	ADD6		0				
		D5	ADD5		0				
		D4	ADD4		0				
		D3	ADD3		0				
		D2	ADD2		0				
		D1	ADD1		0				
		D0	ADD0		0				
A/D conversion result (high-order) register	0040241 (B)	D7-2	–	–	–		–	–	0 when being read.
		D1	ADD9	A/D converted data (high-order 2 bits) ADD9 = MSB	0x0 to 0x3FF (high-order 2 bits)		0	R	
		D0	ADD8		0				
A/D trigger register	0040242 (B)	D7-6	–	–	–		–	–	0 when being read.
		D5	MS	A/D conversion mode selection	1 Continuous	0 Normal	0	R/W	
		D4	TS1	A/D conversion trigger selection	TS[1:0] Trigger		0	R/W	
		D3	TS0		1 1	#ADTRG pin	0		
		1 0	8-bit timer 0						
		0 1	16-bit timer 0						
		0 0	Software						
		D2	CH2	A/D conversion channel status	CH[2:0] Channel		0	R	
		D1	CH1		1 1 1	AD7	0		
		D0	CH0		1 1 0	AD6	0		
		1 0 1	AD5						
1 0 0	AD4								
0 1 1	AD3								
0 1 0	AD2								
0 0 1	AD1								
0 0 0	AD0								

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
A/D channel register	0040243 (B)	D7-6	–	–	–	–	–	–	0 when being read.			
		D5	<b>CE2</b>	A/D converter end channel selection	CE[2:0]		End channel	0	R/W			
		D4	<b>CE1</b>		1	1	1	AD7	0			
		D3	<b>CE0</b>		1	1	0	AD6	0			
					1	0	1	AD5				
					1	0	0	AD4				
					0	1	1	AD3				
					0	1	0	AD2				
					0	0	1	AD1				
					0	0	0	AD0				
		D2	<b>CS2</b>	A/D converter start channel selection	CS[2:0]		Start channel	0	R/W			
		D1	<b>CS1</b>		1	1	1	AD7	0			
		D0	<b>CS0</b>		1	1	0	AD6	0			
					1	0	1	AD5				
		1	0		0	AD4						
		0	1		1	AD3						
		0	1		0	AD2						
		0	0		1	AD1						
		0	0		0	AD0						
A/D enable register	0040244 (B)	D7	–	reserved	–		–	–	0 when being read.			
		D6	<b>INTMODE</b>	Interrupt signal mode	1	Complete only	0	OR	0	R/W	Can be used when	
		D5	<b>CMPINTEN</b>	Out-of-range int. enable	1	Enabled	0	Disabled	0	R/W	ADCADV = "1".	
		D4	<b>CONVINTEN</b>	Conversion-complete int. enable	1	Enabled	0	Disabled	1	R/W		
		D3	<b>ADF</b>	Conversion-complete flag	1	Completed	0	Run/Standby	0	R	Reset when ADD is read.	
		D2	<b>ADE</b>	A/D enable	1	Enabled	0	Disabled	0	R/W		
		D1	<b>ADST</b>	A/D conversion control/status	1	Start/Run	0	Stop	0	R/W		
		D0	<b>OWE</b>	Overwrite error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.	
A/D sampling register	0040245 (B)	D7	<b>ADCMPE</b>	Upper/lower-limit comparison enable	1	Enabled	0	Disabled	0	R/W	Can be used when	
		D6	<b>ADCMP2</b>	Upper/lower-limit comparison channel selection	0 to 7			0	R/W	0	R/W	ADCADV = "1".
		D5	<b>ADCMP1</b>		0							
		D4	<b>ADCMP0</b>		0							
		D3	<b>ADUPRST</b>	Upper-limit comparison status	1	Out of range	0	Within range	0	R		
		D2	<b>ADLWRST</b>	Lower-limit comparison status	1	Out of range	0	Within range	0	R		
		D1	<b>ST1</b>	Input signal sampling time setup	ST[1:0]		Sampling time	1	R/W	1	R/W	Use with 9 clocks.
D0	<b>ST0</b>	1	1		9 clocks							
		1	0		7 clocks							
		0	1		5 clocks							
			0	0	3 clocks							
A/D conversion complete flag register	0040246 (B)	D7	<b>ADF7</b>	Ch.7 conversion-complete flag	1	Completed	0	Run/Standby	0	R	Can be used when ADCADV = "1".	
		D6	<b>ADF6</b>	Ch.6 conversion-complete flag					0	R		
		D5	<b>ADF5</b>	Ch.5 conversion-complete flag					0	R		
		D4	<b>ADF4</b>	Ch.4 conversion-complete flag					0	R		
		D3	<b>ADF3</b>	Ch.3 conversion-complete flag					0	R		
		D2	<b>ADF2</b>	Ch.2 conversion-complete flag					0	R		
		D1	<b>ADF1</b>	Ch.1 conversion-complete flag					0	R		
		D0	<b>ADF0</b>	Ch.0 conversion-complete flag					0	R		
A/D overwrite error flag	0040247 (B)	D7	<b>OWE7</b>	Ch.7 overwrite error flag	1	Error	0	Normal	0	R/W	Can be used when ADCADV = "1".	
		D6	<b>OWE6</b>	Ch.6 overwrite error flag					0	R/W		
		D5	<b>OWE5</b>	Ch.5 overwrite error flag					0	R/W		
		D4	<b>OWE4</b>	Ch.4 overwrite error flag					0	R/W		
		D3	<b>OWE3</b>	Ch.3 overwrite error flag					0	R/W		
		D2	<b>OWE2</b>	Ch.2 overwrite error flag					0	R/W		
		D1	<b>OWE1</b>	Ch.1 overwrite error flag					0	R/W		
		D0	<b>OWE0</b>	Ch.0 overwrite error flag					0	R/W		
A/D Ch.0 conversion result (low-order) buffer	0040248 (B)	D7	<b>AD0BUF7</b>	A/D Ch.0 converted data (low-order 8 bits) AD0BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)		0	R	Can be used when ADCADV = "1".			
		D6	<b>AD0BUF6</b>		0							
		D5	<b>AD0BUF5</b>		0							
		D4	<b>AD0BUF4</b>		0							
		D3	<b>AD0BUF3</b>		0							
		D2	<b>AD0BUF2</b>		0							
		D1	<b>AD0BUF1</b>		0							
		D0	<b>AD0BUF0</b>		0							
		A/D Ch.0 conversion result (high-order) buffer	0040249 (B)		D7-2	–	reserved	–		–	–	0 when being read.
D1	<b>AD0BUF9</b>			A/D Ch.0 converted data (high-order 2 bits) AD0BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)		0	R	Can be used when ADCADV = "1".			
D0	<b>AD0BUF8</b>	0										

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Ch.1 conversion result (low-order) buffer	004024A (B)	D7	AD1BUF7	A/D Ch.1 converted data (low-order 8 bits) AD1BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD1BUF6					
		D5	AD1BUF5					
		D4	AD1BUF4					
		D3	AD1BUF3					
		D2	AD1BUF2					
		D1	AD1BUF1					
		D0	AD1BUF0					
		A/D Ch.1 conversion result (high-order) buffer	004024B (B)			D7-2	–	
D1	AD1BUF9			A/D Ch.1 converted data (high-order 2 bits) AD1BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
D0	AD1BUF8							
A/D Ch.2 conversion result (low-order) buffer	004024C (B)	D7	AD2BUF7	A/D Ch.2 converted data (low-order 8 bits) AD2BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD2BUF6					
		D5	AD2BUF5					
		D4	AD2BUF4					
		D3	AD2BUF3					
		D2	AD2BUF2					
		D1	AD2BUF1					
		D0	AD2BUF0					
		A/D Ch.2 conversion result (high-order) buffer	004024D (B)			D7-2	–	
D1	AD2BUF9			A/D Ch.2 converted data (high-order 2 bits) AD2BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
D0	AD2BUF8							
A/D Ch.3 conversion result (low-order) buffer	004024E (B)	D7	AD3BUF7	A/D Ch.3 converted data (low-order 8 bits) AD3BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD3BUF6					
		D5	AD3BUF5					
		D4	AD3BUF4					
		D3	AD3BUF3					
		D2	AD3BUF2					
		D1	AD3BUF1					
		D0	AD3BUF0					
		A/D Ch.3 conversion result (high-order) buffer	004024F (B)			D7-2	–	
D1	AD3BUF9			A/D Ch.3 converted data (high-order 2 bits) AD3BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
D0	AD3BUF8							
A/D Ch.4 conversion result (low-order) buffer	0040250 (B)	D7	AD4BUF7	A/D Ch.4 converted data (low-order 8 bits) AD4BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD4BUF6					
		D5	AD4BUF5					
		D4	AD4BUF4					
		D3	AD4BUF3					
		D2	AD4BUF2					
		D1	AD4BUF1					
		D0	AD4BUF0					
		A/D Ch.4 conversion result (high-order) buffer	0040251 (B)			D7-2	–	
D1	AD4BUF9			A/D Ch.4 converted data (high-order 2 bits) AD4BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
D0	AD4BUF8							
A/D Ch.5 conversion result (low-order) buffer	0040252 (B)	D7	AD5BUF7	A/D Ch.5 converted data (low-order 8 bits) AD5BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD5BUF6					
		D5	AD5BUF5					
		D4	AD5BUF4					
		D3	AD5BUF3					
		D2	AD5BUF2					
		D1	AD5BUF1					
		D0	AD5BUF0					
		A/D Ch.5 conversion result (high-order) buffer	0040253 (B)			D7-2	–	
D1	AD5BUF9			A/D Ch.5 converted data (high-order 2 bits) AD5BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
D0	AD5BUF8							
A/D Ch.6 conversion result (low-order) buffer	0040254 (B)	D7	AD6BUF7	A/D Ch.6 converted data (low-order 8 bits) AD6BUF0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD6BUF6					
		D5	AD6BUF5					
		D4	AD6BUF4					
		D3	AD6BUF3					
		D2	AD6BUF2					
		D1	AD6BUF1					
		D0	AD6BUF0					

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Ch.6 conversion result (high-order) buffer	0040255 (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	AD6BUF9	A/D Ch.6 converted data (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	AD6BUF8	AD6BUF9 = MSB		0		
A/D Ch.7 conversion result (low-order) buffer	0040256 (B)	D7	AD7BUF7	A/D Ch.7 converted data (low-order 8 bits)	0x0 to 0x3FF (low-order 8 bits)	0	R	Can be used when ADCADV = "1".
		D6	AD7BUF6			0		
		D5	AD7BUF5	AD7BUF0 = LSB		0		
		D4	AD7BUF4			0		
		D3	AD7BUF3			0		
		D2	AD7BUF2			0		
		D1	AD7BUF1			0		
		D0	AD7BUF0			0		
		A/D Ch.7 conversion result (high-order) buffer	0040257 (B)	D7-2	–	reserved	–	
D1	AD7BUF9			A/D Ch.7 converted data (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
D0	AD7BUF8			AD7BUF9 = MSB		0		
A/D upper limit value (low-order) register	0040258 (B)	D7	ADUPR7	A/D conversion upper limit value (low-order 8 bits)	0x0 to 0x3FF (low-order 8 bits)	0	R/W	Can be used when ADCADV = "1".
		D6	ADUPR6			0		
		D5	ADUPR5	ADUPR0 = LSB		0		
		D4	ADUPR4			0		
		D3	ADUPR3			0		
		D2	ADUPR2			0		
		D1	ADUPR1			0		
		D0	ADUPR0			0		
		A/D upper limit value (high-order) register	0040259 (B)	D7-2	–	reserved	–	
D1	ADUPR9			A/D conversion upper limit value (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)	0	R/W	Can be used when ADCADV = "1".
D0	ADUPR8			ADUPR9 = MSB		0		
A/D lower limit value (low-order) register	004025A (B)	D7	ADLWR7	A/D conversion lower limit value (low-order 8 bits)	0x0 to 0x3FF (low-order 8 bits)	0	R/W	Can be used when ADCADV = "1".
		D6	ADLWR6			0		
		D5	ADLWR5	ADLWR0 = LSB		0		
		D4	ADLWR4			0		
		D3	ADLWR3			0		
		D2	ADLWR2			0		
		D1	ADLWR1			0		
		D0	ADLWR0			0		
		A/D lower limit value (high-order) register	004025B (B)	D7-2	–	reserved	–	
D1	ADLWR9			A/D conversion lower limit value (high-order 2 bits)	0x0 to 0x3FF (high-order 2 bits)	0	R/W	Can be used when ADCADV = "1".
D0	ADLWR8			ADLWR9 = MSB		0		
A/D conversion complete interrupt mask register	004025C (B)	D7	INTMASK7	Ch.7 conversion-complete int. mask	1 Interrupt enabled 0 Interrupt mask	1 1 1 1 1 1 1 1	R/W	Can be used when ADCADV = "1".
		D6	INTMASK6	Ch.6 conversion-complete int. mask			R/W	
		D5	INTMASK5	Ch.5 conversion-complete int. mask			R/W	
		D4	INTMASK4	Ch.4 conversion-complete int. mask			R/W	
		D3	INTMASK3	Ch.3 conversion-complete int. mask			R/W	
		D2	INTMASK2	Ch.2 conversion-complete int. mask			R/W	
		D1	INTMASK1	Ch.1 conversion-complete int. mask			R/W	
		D0	INTMASK0	Ch.0 conversion-complete int. mask			R/W	
A/D converter mode select register	004025F (B)	D7-1	–	reserved	–	–	–	0 when being read.
		D0	ADCADV	33209 compatible mode/advanced mode selection	1 Advanced mode 0 33209 compatible mode	0	R/W	
Port input 0/1 interrupt priority register	0040260 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP1L2	Port input 1 interrupt level	0 to 7	X	R/W	
		D5	PP1L1			X		
		D4	PP1L0			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP0L2	Port input 0 interrupt level	0 to 7	X	R/W	
D1	PP0L1	X						
D0	PP0L0	X						
Port input 2/3 interrupt priority register	0040261 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP3L2	Port input 3 interrupt level	0 to 7	X	R/W	
		D5	PP3L1			X		
		D4	PP3L0			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP2L2	Port input 2 interrupt level	0 to 7	X	R/W	
D1	PP2L1	X						
D0	PP2L0	X						

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Key input interrupt priority register	0040262 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PK1L2	Key input 1 interrupt level	0 to 7	X	R/W		
		D5	PK1L1						
		D4	PK1L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PK0L2	Key input 0 interrupt level	0 to 7	X	R/W		
		D1	PK0L1						
		D0	PK0L0						
High-speed DMA Ch.0/1 interrupt priority register	0040263 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PHSD1L2	High-speed DMA Ch.1 interrupt level	0 to 7	X	R/W		
		D5	PHSD1L1						
		D4	PHSD1L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PHSD0L2	High-speed DMA Ch.0 interrupt level	0 to 7	X	R/W		
		D1	PHSD0L1						
		D0	PHSD0L0						
High-speed DMA Ch.2/3 interrupt priority register	0040264 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PHSD3L2	High-speed DMA Ch.3 interrupt level	0 to 7	X	R/W		
		D5	PHSD3L1						
		D4	PHSD3L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PHSD2L2	High-speed DMA Ch.2 interrupt level	0 to 7	X	R/W		
		D1	PHSD2L1						
		D0	PHSD2L0						
IDMA interrupt priority register	0040265 (B)	D7–3	–	reserved	–	–	–	0 when being read.	
		D2	PDM2	IDMA interrupt level	0 to 7	X	R/W		
		D1	PDM1						
		D0	PDM0						
16-bit timer 0/1 interrupt priority register	0040266 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	P16T12	16-bit timer 1 interrupt level	0 to 7	X	R/W		
		D5	P16T11						
		D4	P16T10						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	P16T02	16-bit timer 0 interrupt level	0 to 7	X	R/W		
		D1	P16T01						
		D0	P16T00						
16-bit timer 2/3 interrupt priority register	0040267 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	P16T32	16-bit timer 3 interrupt level	0 to 7	X	R/W		
		D5	P16T31						
		D4	P16T30						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	P16T22	16-bit timer 2 interrupt level	0 to 7	X	R/W		
		D1	P16T21						
		D0	P16T20						
16-bit timer 4/5 interrupt priority register	0040268 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	P16T52	16-bit timer 5 interrupt level	0 to 7	X	R/W		
		D5	P16T51						
		D4	P16T50						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	P16T42	16-bit timer 4 interrupt level	0 to 7	X	R/W		
		D1	P16T41						
		D0	P16T40						
8-bit timer, serial I/F Ch.0 interrupt priority register	0040269 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PSIO02	Serial interface Ch.0 interrupt level	0 to 7	X	R/W		
		D5	PSIO01						
		D4	PSIO00						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	P8TM2	8-bit timer 0–5 interrupt level	0 to 7	X	R/W		
		D1	P8TM1						
		D0	P8TM0						
Serial I/F Ch.1, A/D interrupt priority register	004026A (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PAD2	A/D converter interrupt level	0 to 7	X	R/W		
		D5	PAD1						
		D4	PAD0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PSIO12	Serial interface Ch.1 interrupt level	0 to 7	X	R/W		
		D1	PSIO11						
		D0	PSIO10						

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Clock timer interrupt priority register	004026B (B)	D7-3	–	reserved	–	–	–	Writing 1 not allowed.		
		D2	PCTM2	Clock timer interrupt level	0 to 7	X	R/W			
		D1	PCTM1							
		D0	PCTM0							
Port input 4/5 interrupt priority register	004026C (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PP5L2	Port input 5 interrupt level	0 to 7	X	R/W			
		D5	PP5L1							
		D4	PP5L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PP4L2	Port input 4 interrupt level	0 to 7	X	R/W			
D1	PP4L1									
D0	PP4L0									
Port input 6/7 interrupt priority register	004026D (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PP7L2	Port input 7 interrupt level	0 to 7	X	R/W			
		D5	PP7L1							
		D4	PP7L0							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PP6L2	Port input 6 interrupt level	0 to 7	X	R/W			
D1	PP6L1									
D0	PP6L0									
Serial I/F Ch.2/3 interrupt priority register	004026E (B)	D7	–	reserved	–	–	–	0 when being read.		
		D6	PSIO32	Serial interface Ch.3 interrupt level	0 to 7	X	R/W			
		D5	PSIO31							
		D4	PSIO30							
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	PSIO22	Serial interface Ch.2 interrupt level	0 to 7	X	R/W			
D1	PSIO21									
D0	PSIO20									
Key input, port input 0-3 interrupt enable register	0040270 (B)	D7-6	–	reserved	–	–	–	0 when being read.		
		D5	EK1	Key input 1	1	Enabled	0	R/W		
		D4	EK0							Key input 0
		D3	EP3	Port input 3	0	R/W				
		D2	EP2					Port input 2	0	R/W
		D1	EP1	Port input 1	0	R/W				
		D0	EP0				Port input 0	0	R/W	
DMA interrupt enable register	0040271 (B)	D7-5	–	reserved	–	–				–
		D4	EIDMA	IDMA	1	Enabled	0	Disabled	0	R/W
		D3	EHDM3							
		D2	EHDM2	High-speed DMA Ch.2	0	R/W				
		D1	EHDM1				High-speed DMA Ch.1	0	R/W	
		D0	EHDM0	High-speed DMA Ch.0	0	R/W				
16-bit timer 0/1 interrupt enable register	0040272 (B)	D7	E16TC1				16-bit timer 1 comparison A	1	Enabled	0
		D6	E16TU1	16-bit timer 1 comparison B	0	R/W				
		D5-4	–				reserved	–	–	–
		D3	E16TC0	16-bit timer 0 comparison A	1	Enabled	0	Disabled	0	R/W
		D2	E16TU0							
		D1-0	–	reserved	–	–	–	–	–	0 when being read.
16-bit timer 2/3 interrupt enable register	0040273 (B)	D7	E16TC3	16-bit timer 3 comparison A	1	Enabled	0	Disabled	0	R/W
		D6	E16TU3							
		D5-4	–	reserved	–	–	–	–	–	0 when being read.
		D3	E16TC2	16-bit timer 2 comparison A	1	Enabled	0	Disabled	0	R/W
		D2	E16TU2							
		D1-0	–	reserved	–	–	–	–	–	0 when being read.
16-bit timer 4/5 interrupt enable register	0040274 (B)	D7	E16TC5	16-bit timer 5 comparison A	1	Enabled	0	Disabled	0	R/W
		D6	E16TU5							
		D5-4	–	reserved	–	–	–	–	–	0 when being read.
		D3	E16TC4	16-bit timer 4 comparison A	1	Enabled	0	Disabled	0	R/W
		D2	E16TU4							
		D1-0	–	reserved	–	–	–	–	–	0 when being read.
8-bit timer 0-3 interrupt enable register	0040275 (B)	D7-4	–	reserved	–	–	–	–	–	0 when being read.
		D3	E8TU3	8-bit timer 3 underflow	1	Enabled	0	Disabled	0	R/W
		D2	E8TU2							
		D1	E8TU1	8-bit timer 1 underflow	0	R/W				
		D0	E8TU0				8-bit timer 0 underflow	0	R/W	
Serial I/F Ch.0/1 interrupt enable register	0040276 (B)	D7-6	–	reserved	–	–				–
		D5	ESTX1	SIF Ch.1 transmit buffer empty	1	Enabled	0	Disabled	0	R/W
		D4	ESRX1							
		D3	ESERR1	SIF Ch.1 receive error	0	R/W				
		D2	ESTX0				SIF Ch.0 transmit buffer empty	0	R/W	
		D1	ESRX0	SIF Ch.0 receive buffer full	0	R/W				
		D0	ESERR0				SIF Ch.0 receive error	0	R/W	



Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks					
Port input 4–7, clock timer, A/D interrupt enable register	0040277 (B)	D7–6	–	reserved	–		–	–	0 when being read.					
		D5	EP7	Port input 7	1	Enabled	0	Disabled	0	R/W				
		D4	EP6	Port input 6					0	R/W				
		D3	EP5	Port input 5					0	R/W				
		D2	EP4	Port input 4					0	R/W				
		D1	ECTM	Clock timer					0	R/W				
		D0	EADE	A/D converter					0	R/W				
8-bit timer 4/5 interrupt enable register	0040278 (B)	D7–2	–	reserved					–		–	–	0 when being read.	
		D1	E8TU5	8-bit timer 5 underflow	1	Enabled	0	Disabled	0	R/W				
D0	E8TU4	8-bit timer 4 underflow	0	R/W										
Serial I/F Ch.2/3 interrupt enable register	0040279 (B)	D7–6	–	reserved	–		–	–	0 when being read.					
		D5	ESTX3	SIF Ch.3 transmit buffer empty	1	Enabled	0	Disabled	0	R/W				
		D4	ESRX3	SIF Ch.3 receive buffer full					0	R/W				
		D3	ESERR3	SIF Ch.3 receive error					0	R/W				
		D2	ESTX2	SIF Ch.2 transmit buffer empty					0	R/W				
		D1	ESRX2	SIF Ch.2 receive buffer full					0	R/W				
		D0	ESERR2	SIF Ch.2 receive error					0	R/W				
Key input, port input 0–3 interrupt factor flag register	0040280 (B)	D7–6	–	reserved					–		–	–	0 when being read.	
		D5	FK1	Key input 1	1	Factor is generated	0	No factor is generated	X	R/W				
		D4	FK0	Key input 0					X	R/W				
		D3	FP3	Port input 3					X	R/W				
		D2	FP2	Port input 2					X	R/W				
		D1	FP1	Port input 1					X	R/W				
		D0	FP0	Port input 0					X	R/W				
DMA interrupt factor flag register	0040281 (B)	D7–5	–	reserved					–		–	–	0 when being read.	
		D4	FIDMA	IDMA	1	Factor is generated	0	No factor is generated	X	R/W				
		D3	FHDM3	High-speed DMA Ch.3					X	R/W				
		D2	FHDM2	High-speed DMA Ch.2					X	R/W				
		D1	FHDM1	High-speed DMA Ch.1					X	R/W				
		D0	FHDM0	High-speed DMA Ch.0					X	R/W				
16-bit timer 0/1 interrupt factor flag register	0040282 (B)	D7	F16TC1	16-bit timer 1 comparison A					1	Factor is generated	0	No factor is generated	X	R/W
		D6	F16TU1	16-bit timer 1 comparison B	X	R/W								
		D5–4	–	reserved	–		–	–					0 when being read.	
		D3	F16TC0	16-bit timer 0 comparison A	1	Factor is generated	0	No factor is generated					X	R/W
		D2	F16TU0	16-bit timer 0 comparison B									X	R/W
		D1–0	–	reserved									–	
16-bit timer 2/3 interrupt factor flag register	0040283 (B)	D7	F16TC3	16-bit timer 3 comparison A	1	Factor is generated	0	No factor is generated	X	R/W				
		D6	F16TU3	16-bit timer 3 comparison B					X	R/W				
		D5–4	–	reserved					–		–	–	0 when being read.	
		D3	F16TC2	16-bit timer 2 comparison A					1	Factor is generated	0	No factor is generated	X	R/W
		D2	F16TU2	16-bit timer 2 comparison B									X	R/W
		D1–0	–	reserved									–	
16-bit timer 4/5 interrupt factor flag register	0040284 (B)	D7	F16TC5	16-bit timer 5 comparison A	1	Factor is generated	0	No factor is generated	X	R/W				
		D6	F16TU5	16-bit timer 5 comparison B					X	R/W				
		D5–4	–	reserved					–		–	–	0 when being read.	
		D3	F16TC4	16-bit timer 4 comparison A					1	Factor is generated	0	No factor is generated	X	R/W
		D2	F16TU4	16-bit timer 4 comparison B									X	R/W
		D1–0	–	reserved									–	
8-bit timer 0–3 interrupt factor flag register	0040285 (B)	D7–4	–	reserved	–		–	–	0 when being read.					
		D3	F8TU3	8-bit timer 3 underflow	1	Factor is generated	0	No factor is generated	X	R/W				
		D2	F8TU2	8-bit timer 2 underflow					X	R/W				
		D1	F8TU1	8-bit timer 1 underflow					X	R/W				
		D0	F8TU0	8-bit timer 0 underflow					X	R/W				
Serial I/F Ch.0/1 interrupt factor flag register	0040286 (B)	D7–6	–	reserved					–		–	–	0 when being read.	
		D5	FSTX1	SIF Ch.1 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W				
		D4	FSRX1	SIF Ch.1 receive buffer full					X	R/W				
		D3	FSERR1	SIF Ch.1 receive error					X	R/W				
		D2	FSTX0	SIF Ch.0 transmit buffer empty					X	R/W				
		D1	FSRX0	SIF Ch.0 receive buffer full					X	R/W				
		D0	FSERR0	SIF Ch.0 receive error					X	R/W				
Port input 4–7, clock timer, A/D interrupt factor flag register	0040287 (B)	D7–6	–	reserved					–		–	–	0 when being read.	
		D5	FP7	Port input 7	1	Factor is generated	0	No factor is generated	X	R/W				
		D4	FP6	Port input 6					X	R/W				
		D3	FP5	Port input 5					X	R/W				
		D2	FP4	Port input 4					X	R/W				
		D1	FACTM	Clock timer					X	R/W				
		D0	FADE	A/D converter					X	R/W				
8-bit timer 4/5 interrupt factor flag register	0040288 (B)	D7–2	–	reserved					–		–	–	0 when being read.	
		D1	F8TU5	8-bit timer 5 underflow	1	Factor is generated	0	No factor is generated	X	R/W				
		D0	F8TU4	8-bit timer 4 underflow					X	R/W				

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Serial I/F Ch.2/3 interrupt factor flag register	0040289 (B)	D7-6	–	reserved	–	–	–	0 when being read.		
		D5	FSTX3	SIF Ch.3 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W
		D4	FSRX3	SIF Ch.3 receive buffer full					X	R/W
		D3	FSERR3	SIF Ch.3 receive error					X	R/W
		D2	FSTX2	SIF Ch.2 transmit buffer empty					X	R/W
		D1	FSRX2	SIF Ch.2 receive buffer full					X	R/W
		D0	FSERR2	SIF Ch.2 receive error					X	R/W
Port input 0–3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA request register	0040290 (B)	D7	R16TC0	16-bit timer 0 comparison A					1	IDMA request
		D6	R16TU0	16-bit timer 0 comparison B	0	R/W				
		D5	RHDM1	High-speed DMA Ch.1	0	R/W				
		D4	RHDM0	High-speed DMA Ch.0	0	R/W				
		D3	RP3	Port input 3	0	R/W				
		D2	RP2	Port input 2	0	R/W				
		D1	RP1	Port input 1	0	R/W				
		D0	RP0	Port input 0	0	R/W				
16-bit timer 1–4 IDMA request register	0040291 (B)	D7	R16TC4	16-bit timer 4 comparison A	1	IDMA request	0	Interrupt request	0	R/W
		D6	R16TU4	16-bit timer 4 comparison B					0	R/W
		D5	R16TC3	16-bit timer 3 comparison A					0	R/W
		D4	R16TU3	16-bit timer 3 comparison B					0	R/W
		D3	R16TC2	16-bit timer 2 comparison A					0	R/W
		D2	R16TU2	16-bit timer 2 comparison B					0	R/W
		D1	R16TC1	16-bit timer 1 comparison A					0	R/W
		D0	R16TU1	16-bit timer 1 comparison B					0	R/W
		16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA request register	0040292 (B)	D7					RSTX0	SIF Ch.0 transmit buffer empty
D6	RSRX0			SIF Ch.0 receive buffer full	0	R/W				
D5	R8TU3			8-bit timer 3 underflow	0	R/W				
D4	R8TU2			8-bit timer 2 underflow	0	R/W				
D3	R8TU1			8-bit timer 1 underflow	0	R/W				
D2	R8TU0			8-bit timer 0 underflow	0	R/W				
D1	R16TC5			16-bit timer 5 comparison A	0	R/W				
D0	R16TU5			16-bit timer 5 comparison B	0	R/W				
Serial I/F Ch.1, A/D, port input 4–7 IDMA request register	0040293 (B)			D7	RP7	Port input 7	1	IDMA request	0	Interrupt request
		D6	RP6	Port input 6	0	R/W				
		D5	RP5	Port input 5	0	R/W				
		D4	RP4	Port input 4	0	R/W				
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	RADE	A/D converter	1	IDMA request	0	Interrupt request	0	R/W
		D1	RSTX1	SIF Ch.1 transmit buffer empty					0	R/W
		D0	RSRX1	SIF Ch.1 receive buffer full					0	R/W
Port input 0–3, high-speed DMA Ch. 0/1, 16-bit timer 0 IDMA enable register	0040294 (B)	D7	DE16TC0	16-bit timer 0 comparison A	1	IDMA enabled	0	IDMA disabled	0	R/W
		D6	DE16TU0	16-bit timer 0 comparison B					0	R/W
		D5	DEHDM1	High-speed DMA Ch.1					0	R/W
		D4	DEHDM0	High-speed DMA Ch.0					0	R/W
		D3	DEP3	Port input 3					0	R/W
		D2	DEP2	Port input 2					0	R/W
		D1	DEP1	Port input 1					0	R/W
		D0	DEP0	Port input 0					0	R/W
16-bit timer 1–4 IDMA enable register	0040295 (B)	D7	DE16TC4	16-bit timer 4 comparison A	1	IDMA enabled	0	IDMA disabled	0	R/W
		D6	DE16TU4	16-bit timer 4 comparison B					0	R/W
		D5	DE16TC3	16-bit timer 3 comparison A					0	R/W
		D4	DE16TU3	16-bit timer 3 comparison B					0	R/W
		D3	DE16TC2	16-bit timer 2 comparison A					0	R/W
		D2	DE16TU2	16-bit timer 2 comparison B					0	R/W
		D1	DE16TC1	16-bit timer 1 comparison A					0	R/W
		D0	DE16TU1	16-bit timer 1 comparison B					0	R/W
16-bit timer 5, 8-bit timer 0–3, serial I/F Ch.0 IDMA enable register	0040296 (B)	D7	DESTX0	SIF Ch.0 transmit buffer empty	1	IDMA enabled	0	IDMA disabled	0	R/W
		D6	DESRX0	SIF Ch.0 receive buffer full					0	R/W
		D5	DE8TU3	8-bit timer 3 underflow					0	R/W
		D4	DE8TU2	8-bit timer 2 underflow					0	R/W
		D3	DE8TU1	8-bit timer 1 underflow					0	R/W
		D2	DE8TU0	8-bit timer 0 underflow					0	R/W
		D1	DE16TC5	16-bit timer 5 comparison A					0	R/W
		D0	DE16TU5	16-bit timer 5 comparison B					0	R/W
Serial I/F Ch.1, A/D, port input 4–7 IDMA enable register	0040297 (B)	D7	DEP7	Port input 7	1	IDMA enabled	0	IDMA disabled	0	R/W
		D6	DEP6	Port input 6					0	R/W
		D5	DEP5	Port input 5					0	R/W
		D4	DEP4	Port input 4					0	R/W
		D3	–	reserved	–	–	–	–	0 when being read.	
		D2	DEADE	A/D converter	1	IDMA enabled	0	IDMA disabled	0	R/W
		D1	DESTX1	SIF Ch.1 transmit buffer empty					0	R/W
		D0	DESRX1	SIF Ch.1 receive buffer full					0	R/W

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
High-speed DMA Ch.0/1 trigger set-up register	0040298 (B)	D7	HSD1S3	High-speed DMA Ch.1 trigger set-up	0	Software trigger	0	R/W			
		D6	HSD1S2		1	K51 input (falling edge)	0				
D5	HSD1S1	2	K51 input (rising edge)		0						
D4	HSD1S0	3	Port 1 input		0						
			4		Port 5 input						
			5		8-bit timer Ch.1 underflow						
			6		16-bit timer Ch.1 compare B						
			7		16-bit timer Ch.1 compare A						
			8		16-bit timer Ch.5 compare B						
			9		16-bit timer Ch.5 compare A						
			A	SI/F Ch.1 Rx buffer full							
			B	SI/F Ch.1 Tx buffer empty							
			C	A/D conversion completion							
			D	FSI/F Ch.0 Rx buffer full							
			E	FSI/F Ch.0 Tx buffer empty							
High-speed DMA Ch.2/3 trigger set-up register	0040299 (B)	D7	HSD3S3	High-speed DMA Ch.3 trigger set-up	0	Software trigger	0	R/W			
		D6	HSD3S2		1	K54 input (falling edge)	0				
D5	HSD3S1	2	K54 input (rising edge)		0						
D4	HSD3S0	3	Port 3 input		0						
			4		Port 7 input						
			5		8-bit timer Ch.3 underflow						
			6		16-bit timer Ch.3 compare B						
			7		16-bit timer Ch.3 compare A						
			8		16-bit timer Ch.5 compare B						
			9		16-bit timer Ch.5 compare A						
			A	SI/F Ch.1 Rx buffer full							
			B	SI/F Ch.1 Tx buffer empty							
			C	A/D conversion completion							
			D	FSI/F Ch.0 Rx buffer full							
			E	FSI/F Ch.0 Tx buffer empty							
High-speed DMA Ch.2 trigger set-up register	004029A (B)	D3	HSD2S3	High-speed DMA Ch.2 trigger set-up	0	Software trigger	0	R/W			
		D2	HSD2S2		1	K53 input (falling edge)	0				
D1	HSD2S1	2	K53 input (rising edge)		0						
D0	HSD2S0	3	Port 2 input		0						
			4		Port 6 input						
			5		8-bit timer Ch.2 underflow						
			6		16-bit timer Ch.2 compare B						
			7		16-bit timer Ch.2 compare A						
			8		16-bit timer Ch.4 compare B						
			9		16-bit timer Ch.4 compare A						
			A	SI/F Ch.0 Rx buffer full							
			B	SI/F Ch.0 Tx buffer empty							
			C	A/D conversion completion							
			D	reserved							
			E	reserved							
High-speed DMA software trigger register	004029A (B)	D7-4	–	reserved	–	–	–	0 when being read.			
		D3	HST3	HSDMA Ch.3 software trigger	1	Trigger	0		Invalid	0	W
		D2	HST2	HSDMA Ch.2 software trigger						0	W
		D1	HST1	HSDMA Ch.1 software trigger						0	W
		D0	HST0	HSDMA Ch.0 software trigger						0	W
8-bit timer 4/5 serial I/F Ch.2/3 IDMA request register	004029B (B)	D7-6	–	reserved	–	–	–	0 when being read.			
		D5	RSTX3	SIF Ch.3 transmit buffer empty	1	IDMA request	0		Interrupt request	0	R/W
		D4	RSRX3	SIF Ch.3 receive buffer full						0	R/W
		D3	RSTX2	SIF Ch.2 transmit buffer empty						0	R/W
		D2	RSRX2	SIF Ch.2 receive buffer full						0	R/W
		D1	R8TU5	8-bit timer 5 underflow						0	R/W
		D0	R8TU4	8-bit timer 4 underflow						0	R/W

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
8-bit timer 4/5 serial I/F Ch.2/3 IDMA enable register	004029C (B)	D7-6	–	reserved	–	–	–	0 when being read.	
		D5	DESTX3	SIF Ch.3 transmit buffer empty	1 IDMA enabled	0 IDMA disabled	0	R/W	
		D4	DESRX3	SIF Ch.3 receive buffer full			0	R/W	
		D3	DESTX2	SIF Ch.2 transmit buffer empty			0	R/W	
		D2	DESRX2	SIF Ch.2 receive buffer full			0	R/W	
		D1	DE8TU5	8-bit timer 5 underflow			0	R/W	
		D0	DE8TU4	8-bit timer 4 underflow			0	R/W	
Flag set/reset method select register	004029F (B)	D7-3	–	reserved	–	–	–		
		D2	DENONLY	IDMA enable register set method selection	1 Set only	0 RD/WR	1	R/W	
		D1	IDMAONLY	IDMA request register set method selection	1 Set only	0 RD/WR	1	R/W	
		D0	RSTONLY	Interrupt factor flag reset method selection	1 Reset only	0 RD/WR	1	R/W	
Interrupt priority register for functions extended	00402A0 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D5	–				X		
		D4	–				X		
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D1	–				X		
D0	–				X				
Interrupt priority register for functions extended	00402A1 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D5	–				X		
		D4	–				X		
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D1	–				X		
D0	–				X				
Interrupt priority register for functions extended	00402A2 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D5	–				X		
		D4	–				X		
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D1	–				X		
D0	–				X				
Interrupt priority register for functions extended	00402A3 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D5	–				X		
		D4	–				X		
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D1	–				X		
D0	–				X				
Interrupt priority register for functions extended	00402A4 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D5	–				X		
		D4	–				X		
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D1	–				X		
D0	–				X				
Interrupt priority register for functions extended	00402A5 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D5	–				X		
		D4	–				X		
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	–	Interrupt level of extended function (reserved)	0 to 7		X	R/W	
		D1	–				X		
D0	–				X				

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Interrupt enable register for functions extended	00402A6 (B)	D7	–	Extended interrupt (reserved)	1	Enabled	0	Disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
Interrupt enable register for functions extended	00402A7 (B)	D7	–	Extended interrupt (reserved)	1	Enabled	0	Disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
Interrupt enable register for functions extended	00402A8 (B)	D7	–	Extended interrupt (reserved)	1	Enabled	0	Disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
Interrupt factor flag register for functions extended	00402A9 (B)	D7	–	Extended interrupt (reserved)	1	Factor is generated	0	No factor is generated	X	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					X	R/W	
		D5	–	Extended interrupt (reserved)					X	R/W	
		D4	–	Extended interrupt (reserved)					X	R/W	
		D3	–	Extended interrupt (reserved)					X	R/W	
		D2	–	Extended interrupt (reserved)					X	R/W	
		D1	–	Extended interrupt (reserved)					X	R/W	
		D0	–	Extended interrupt (reserved)					X	R/W	
Interrupt factor flag register for functions extended	00402AA (B)	D7	–	Extended interrupt (reserved)	1	Factor is generated	0	No factor is generated	X	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					X	R/W	
		D5	–	Extended interrupt (reserved)					X	R/W	
		D4	–	Extended interrupt (reserved)					X	R/W	
		D3	–	Extended interrupt (reserved)					X	R/W	
		D2	–	Extended interrupt (reserved)					X	R/W	
		D1	–	Extended interrupt (reserved)					X	R/W	
		D0	–	Extended interrupt (reserved)					X	R/W	
Interrupt factor flag register for functions extended	00402AB (B)	D7	–	Extended interrupt (reserved)	1	Factor is generated	0	No factor is generated	X	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					X	R/W	
		D5	–	Extended interrupt (reserved)					X	R/W	
		D4	–	Extended interrupt (reserved)					X	R/W	
		D3	–	Extended interrupt (reserved)					X	R/W	
		D2	–	Extended interrupt (reserved)					X	R/W	
		D1	–	Extended interrupt (reserved)					X	R/W	
		D0	–	Extended interrupt (reserved)					X	R/W	
IDMA request register for functions extended	00402AC (B)	D7	–	Extended interrupt (reserved)	1	IDMA request	0	Interrupt request	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
IDMA request register for functions extended	00402AD (B)	D7	–	Extended interrupt (reserved)	1	IDMA request	0	Interrupt request	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
IDMA enable register for functions extended	00402AE (B)	D7	–	Extended interrupt (reserved)	1	IDMA enabled	0	IDMA disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
IDMA enable register for functions extended	00402AF (B)	D7	–	Extended interrupt (reserved)	1	IDMA enabled	0	IDMA disabled	0	R/W	Do not write 1.
		D6	–	Extended interrupt (reserved)					0	R/W	
		D5	–	Extended interrupt (reserved)					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	
		D3	–	Extended interrupt (reserved)					0	R/W	
		D2	–	Extended interrupt (reserved)					0	R/W	
		D1	–	Extended interrupt (reserved)					0	R/W	
		D0	–	Extended interrupt (reserved)					0	R/W	
FIFO serial I/F Ch.0 interrupt priority register	00402B0 (B)	D7	–	reserved		–	–	–	–	0 when being read.	
		D6	PFSIO02	FIFO serial interface Ch.0 interrupt level		0 to 7	X	R/W			
		D5	PFSIO01				X				
		D4	PFSIO00				X				
D3–0	–	reserved		–	–	–	–	0 when being read.			
FIFO serial I/F Ch.0 interrupt enable register	00402B1 (B)	D7–6	–	reserved		–	–	–	–	0 when being read.	
		D5	EFSTX0	FSI/F Ch.0 transmit buffer empty	1	Enabled	0	Disabled	0	R/W	
		D4	EFSRX0	FSI/F Ch.0 receive buffer full					0	R/W	
		D3	EFSERR0	FSI/F Ch.0 receive error					0	R/W	
D2–0	–	reserved		–	–	–	–	0 when being read.			
FIFO serial I/F Ch.0 interrupt factor flag register	00402B2 (B)	D7–6	–	reserved		–	–	–	–	0 when being read.	
		D5	FFSTX0	FSI/F Ch.0 transmit buffer empty	1	Factor is generated	0	No factor is generated	X	R/W	
		D4	FFSRX0	FSI/F Ch.0 receive buffer full					X	R/W	
		D3	FFSERR0	FSI/F Ch.0 receive error					X	R/W	
D2–0	–	reserved		–	–	–	–	0 when being read.			
FIFO serial I/F Ch.0 IDMA request register	00402B3 (B)	D7–4	–	reserved		–	–	–	–	0 when being read.	
		D3	RFSTX0	FSI/F Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request	0	R/W	
		D2	RFSRX0	FSI/F Ch.0 receive buffer full					0	R/W	
D1–0	–	reserved		–	–	–	–	0 when being read.			
FIFO serial I/F Ch.0 IDMA enable register	00402B4 (B)	D7–4	–	reserved		–	–	–	–	0 when being read.	
		D3	DEFSTX0	FSI/F Ch.0 transmit buffer empty	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D2	DEFSRX0	FSI/F Ch.0 receive buffer full					0	R/W	
D1–0	–	reserved		–	–	–	–	0 when being read.			
K5 function select register	00402C0 (B)	D7–5	–	reserved		–	–	–	–	0 when being read.	
		D4	CFK54	K54 function selection	1	#DMAREQ3	0	K54	0	R/W	
		D3	CFK53	K53 function selection	1	#DMAREQ2	0	K53	0	R/W	
		D2	CFK52	K52 function selection	1	#ADTRG	0	K52	0	R/W	
		D1	CFK51	K51 function selection	1	#DMAREQ1	0	K51	0	R/W	
D0	CFK50	K50 function selection	1	#DMAREQ0	0	K50	0	R/W			
K5 input port data register	00402C1 (B)	D7–5	–	reserved		–	–	–	–	0 when being read.	
		D4	K54D	K54 input port data	1	High	0	Low	–	R	
		D3	K53D	K53 input port data					–	R	
		D2	K52D	K52 input port data					–	R	
		D1	K51D	K51 input port data					–	R	
		D0	K50D	K50 input port data					–	R	
K6 function select register	00402C3 (B)	D7	CFK67	K67 function selection					1	AD7	
		D6	CFK66	K66 function selection	1	AD6	0	K66	0	R/W	
		D5	CFK65	K65 function selection	1	AD5	0	K65	0	R/W	
		D4	CFK64	K64 function selection	1	AD4	0	K64	0	R/W	
		D3	CFK63	K63 function selection	1	AD3	0	K63	0	R/W	
		D2	CFK62	K62 function selection	1	AD2	0	K62	0	R/W	
		D1	CFK61	K61 function selection	1	AD1	0	K61	0	R/W	
		D0	CFK60	K60 function selection	1	AD0	0	K60	0	R/W	
K6 input port data register	00402C4 (B)	D7	K67D	K67 input port data	1	High	0	Low	–	R	
		D6	K66D	K66 input port data					–	R	
		D5	K65D	K65 input port data					–	R	
		D4	K64D	K64 input port data					–	R	
		D3	K63D	K63 input port data					–	R	
		D2	K62D	K62 input port data					–	R	
		D1	K61D	K61 input port data					–	R	
		D0	K60D	K60 input port data					–	R	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Port input interrupt select register 1	00402C6 (B)	D7	SPT31	FPT3 interrupt input port selection	11	10	01	00	0	R/W	
		D6	SPT30		P23	P03	K53	K63	0		
		D5	SPT21	FPT2 interrupt input port selection	11	10	01	00	0	R/W	
		D4	SPT20		P22	P02	K52	K62	0		
		D3	SPT11	FPT1 interrupt input port selection	11	10	01	00	0	R/W	
		D2	SPT10		P21	P01	K51	K61	0		
		D1	SPT01	FPT0 interrupt input port selection	11	10	01	00	0	R/W	
		D0	SPT00		P20	P00	K50	K60	0		
Port input interrupt select register 2	00402C7 (B)	D7	SPT71	FPT7 interrupt input port selection	11	10	01	00	0	R/W	
		D6	SPT70		P27	P07	P33	K67	0		
		D5	SPT61	FPT6 interrupt input port selection	11	10	01	00	0	R/W	
		D4	SPT60		P26	P06	P32	K66	0		
		D3	SPT51	FPT5 interrupt input port selection	11	10	01	00	0	R/W	
		D2	SPT50		P25	P05	P31	K65	0		
		D1	SPT41	FPT4 interrupt input port selection	11	10	01	00	0	R/W	
		D0	SPT40		P24	P04	K54	K64	0		
Port input interrupt input polarity select register	00402C8 (B)	D7	SPPT7	FPT7 input polarity selection	1	High level or Rising edge	0	Low level or Falling edge	1	R/W	
		D6	SPPT6	FPT6 input polarity selection					1	R/W	
		D5	SPPT5	FPT5 input polarity selection					1	R/W	
		D4	SPPT4	FPT4 input polarity selection					1	R/W	
		D3	SPPT3	FPT3 input polarity selection					1	R/W	
		D2	SPPT2	FPT2 input polarity selection					1	R/W	
		D1	SPPT1	FPT1 input polarity selection					1	R/W	
		D0	SPPT0	FPT0 input polarity selection					1	R/W	
Port input interrupt edge/level select register	00402C9 (B)	D7	SEPT7	FPT7 edge/level selection	1	Edge	0	Level	1	R/W	
		D6	SEPT6	FPT6 edge/level selection					1	R/W	
		D5	SEPT5	FPT5 edge/level selection					1	R/W	
		D4	SEPT4	FPT4 edge/level selection					1	R/W	
		D3	SEPT3	FPT3 edge/level selection					1	R/W	
		D2	SEPT2	FPT2 edge/level selection					1	R/W	
		D1	SEPT1	FPT1 edge/level selection					1	R/W	
		D0	SEPT0	FPT0 edge/level selection					1	R/W	
Key input interrupt select register	00402CA (B)	D7-4	–	reserved	–				–	–	0 when being read.
		D3	SPPK11	FPK11 interrupt input port selection	11	10	01	00	0	R/W	
		D2	SPPK10		P2[7:4]	P0[7:4]	K6[7:4]	K6[3:0]	0		
		D1	SPPK01	FPK0 interrupt input port selection	11	10	01	00	0	R/W	
		D0	SPPK00		P2[4:0]	P0[4:0]	K6[4:0]	K5[4:0]	0		
Key input interrupt (FPK0) input comparison register	00402CC (B)	D7-5	–	reserved	–				–	–	0 when being read.
		D4	SCPK04	FPK04 input comparison	1	High	0	Low	0	R/W	
		D3	SCPK03	FPK03 input comparison					0	R/W	
		D2	SCPK02	FPK02 input comparison					0	R/W	
		D1	SCPK01	FPK01 input comparison					0	R/W	
D0	SCPK00	FPK00 input comparison			0		R/W				
Key input interrupt (FPK1) input comparison register	00402CD (B)	D7-4	–	reserved	–				–	–	0 when being read.
		D3	SCPK13	FPK13 input comparison	1	High	0	Low	0	R/W	
		D2	SCPK12	FPK12 input comparison					0	R/W	
		D1	SCPK11	FPK11 input comparison					0	R/W	
		D0	SCPK10	FPK10 input comparison					0	R/W	
Key input interrupt (FPK0) input mask register	00402CE (B)	D7-5	–	reserved	–				–	–	
		D4	SMPK04	FPK04 input mask	1	Interrupt enabled	0	Interrupt disabled	0	R/W	
		D3	SMPK03	FPK03 input mask					0	R/W	
		D2	SMPK02	FPK02 input mask					0	R/W	
		D1	SMPK01	FPK01 input mask					0	R/W	
D0	SMPK00	FPK00 input mask			0		R/W				
Key input interrupt (FPK1) input mask register	00402CF (B)	D7-4	–	reserved	–				–	–	0 when being read.
		D3	SMPK13	FPK13 input mask	1	Interrupt enabled	0	Interrupt disabled	0	R/W	
		D2	SMPK12	FPK12 input mask					0	R/W	
		D1	SMPK11	FPK11 input mask					0	R/W	
		D0	SMPK10	FPK10 input mask					0	R/W	
P0 function select register	00402D0 (B)	D7	CFP07	P07 function selection	1		#SRDY1		0	P07	
		D6	CFP06	P06 function selection	1	#SCLK1	0	P06	0	R/W	
		D5	CFP05	P05 function selection	1	SOUT1	0	P05	0	R/W	
		D4	CFP04	P04 function selection	1	SIN1	0	P04	0	R/W	
		D3	CFP03	P03 function selection	1	#SRDY0	0	P03	0	R/W	Extended functions (0x300040)
		D2	CFP02	P02 function selection	1	#SCLK0	0	P02	0	R/W	
		D1	CFP01	P01 function selection	1	SOUT0	0	P01	0	R/W	
		D0	CFP00	P00 function selection	1	SIN0	0	P00	0	R/W	

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
P0 I/O port data register	00402D1 (B)	D7	P07D	P07 I/O port data	1	High	0	Low	0	R/W	
		D6	P06D	P06 I/O port data					0	R/W	
		D5	P05D	P05 I/O port data					0	R/W	
		D4	P04D	P04 I/O port data					0	R/W	
		D3	P03D	P03 I/O port data					0	R/W	
		D2	P02D	P02 I/O port data					0	R/W	
		D1	P01D	P01 I/O port data					0	R/W	
		D0	P00D	P00 I/O port data					0	R/W	
P0 I/O control register	00402D2 (B)	D7	IOC07	P07 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D6	IOC06	P06 I/O control					0	R/W	
		D5	IOC05	P05 I/O control					0	R/W	
		D4	IOC04	P04 I/O control					0	R/W	
		D3	IOC03	P03 I/O control					0	R/W	
		D2	IOC02	P02 I/O control					0	R/W	
		D1	IOC01	P01 I/O control					0	R/W	
		D0	IOC00	P00 I/O control					0	R/W	
P1 function select register	00402D4 (B)	D7	–	reserved	–		–	–	0	–	0 when being read.
		D6	CFP16	P16 function selection	1	EXCL5 #DMAEND1	0	P16	0	R/W	Extended functions (0x402D7)
		D5	CFP15	P15 function selection	1	EXCL4 #DMAEND0	0	P15	0	R/W	
		D4	CFP14	P14 function selection	1	FOSC1	0	P14	0	R/W	Extended functions (0x402DF)
		D3	CFP13	P13 function selection	1	EXCL3 T8UF3	0	P13	0	R/W	
		D2	CFP12	P12 function selection	1	EXCL2 T8UF2	0	P12	0	R/W	
		D1	CFP11	P11 function selection	1	EXCL1 T8UF1	0	P11	0	R/W	
D0	CFP10	P10 function selection	1	EXCL0 T8UF0	0	P10	0	R/W			
P1 I/O port data register	00402D5 (B)	D7	–	reserved	–		–	–	0	–	0 when being read.
		D6	P16D	P16 I/O port data	1	High	0	Low	0	R/W	
		D5	P15D	P15 I/O port data					0	R/W	
		D4	P14D	P14 I/O port data					0	R/W	
		D3	P13D	P13 I/O port data					0	R/W	
		D2	P12D	P12 I/O port data					0	R/W	
		D1	P11D	P11 I/O port data					0	R/W	
D0	P10D	P10 I/O port data	0	R/W							
P1 I/O control register	00402D6 (B)	D7	–	reserved	–		–	–	0	–	0 when being read.
		D6	IOC16	P16 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D5	IOC15	P15 I/O control					0	R/W	
		D4	IOC14	P14 I/O control					0	R/W	
		D3	IOC13	P13 I/O control					0	R/W	
		D2	IOC12	P12 I/O control					0	R/W	
		D1	IOC11	P11 I/O control					0	R/W	
D0	IOC10	P10 I/O control	0	R/W							
Port SIO function extension register	00402D7 (B)	D7–4	–	reserved	–		–	–	0	–	0 when being read.
		D3	SSRDY3	Serial I/F Ch.3 SRDY selection	1	#SRDY3	0	P32/#DMAACK0	0	R/W	
		D2	SSCLK3	Serial I/F Ch.3 SCLK selection	1	#SCLK3	0	P15/EXCL4/#DMAEND0	0	R/W	
		D1	SSOUT3	Serial I/F Ch.3 SOUT selection	1	SOUT3	0	P16/EXCL5/#DMAEND1	0	R/W	
D0	SSIN3	Serial I/F Ch.3 SIN selection	1	SIN3	0	P33/#DMAACK1	0	R/W			
P2 function select register	00402D8 (B)	D7	CFP27	P27 function selection	1	TM5	0	P27	0	R/W	Ext. func.(0x402DB)
		D6	CFP26	P26 function selection	1	TM4	0	P26	0	R/W	
		D5	CFP25	P25 function selection	1	TM3	0	P25	0	R/W	
		D4	CFP24	P24 function selection	1	TM2	0	P24	0	R/W	
		D3	CFP23	P23 function selection	1	TM1	0	P23	0	R/W	
		D2	CFP22	P22 function selection	1	TM0	0	P22	0	R/W	
		D1	CFP21	P21 function selection	1	#DWE	0	P21	0	R/W	Ext. func.(0x402DF)
D0	CFP20	P20 function selection	1	#DRD	0	P20	0	R/W			



Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P2 I/O port data register	00402D9 (B)	D7	P27D	P27 I/O port data	1	High	0	Low	0	R/W	
		D6	P26D	P26 I/O port data					0	R/W	
		D5	P25D	P25 I/O port data					0	R/W	
		D4	P24D	P24 I/O port data					0	R/W	
		D3	P23D	P23 I/O port data					0	R/W	
		D2	P22D	P22 I/O port data					0	R/W	
		D1	P21D	P21 I/O port data					0	R/W	
		D0	P20D	P20 I/O port data					0	R/W	
		P2 I/O control register	00402DA (B)	D7					IOC27	P27 I/O control	
D6	IOC26			P26 I/O control	0	R/W					
D5	IOC25			P25 I/O control	0	R/W					
D4	IOC24			P24 I/O control	0	R/W					
D3	IOC23			P23 I/O control	0	R/W					
D2	IOC22			P22 I/O control	0	R/W					
D1	IOC21			P21 I/O control	0	R/W					
D0	IOC20			P20 I/O control	0	R/W					
Port SIO function extension register	00402DB (B)			D7-4	–	reserved	–		–	–	–
		D3	SSRDY2	Serial I/F Ch.2 SRDY selection	1	#SRDY2	0	P24/TM2	0	R/W	
		D2	SSCLK2	Serial I/F Ch.2 SCLK selection	1	#SCLK2	0	P25/TM3	0	R/W	
		D1	SSOUT2	Serial I/F Ch.2 SOUT selection	1	SOUT2	0	P26/TM4	0	R/W	
		D0	SSIN2	Serial I/F Ch.2 SIN selection	1	SIN2	0	P27/TM5	0	R/W	
P3 function select register	00402DC (B)	D7-6	–	reserved	–		–	–	–	–	
		D5	CFP35	P35 function selection	1	#BUSACK	0	P35	0	R/W	Ext. func.(0x300047)
		D4	CFP34	P34 function selection	1	#BUSREQ #CE6	0	P34	0	R/W	
		D3	CFP33	P33 function selection	1	#DMAACK1	0	P33	0	R/W	Ext. func.(0x402D7)
		D2	CFP32	P32 function selection	1	#DMAACK0	0	P32	0	R/W	Ext. func.(0x402DF)
		D1	CFP31	P31 function selection	1	#BUSGET	0	P31	0	R/W	
		D0	CFP30	P30 function selection	1	#WAIT #CE4/#CE5	0	P30	0	R/W	
P3 I/O port data register	00402DD (B)	D7-6	–	reserved	–		–	–	–	–	0 when being read.
		D5	P35D	P35 I/O port data	1	High	0	Low	0	R/W	
		D4	P34D	P34 I/O port data					0	R/W	
		D3	P33D	P33 I/O port data					0	R/W	
		D2	P32D	P32 I/O port data					0	R/W	
		D1	P31D	P31 I/O port data					0	R/W	
		D0	P30D	P30 I/O port data					0	R/W	
P3 I/O control register	00402DE (B)	D7-6	–	reserved					–		
		D5	IOC35	P35 I/O control	1	Output	0	Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D4	IOC34	P34 I/O control					0	R/W	
		D3	IOC33	P33 I/O control					0	R/W	
		D2	IOC32	P32 I/O control					0	R/W	
		D1	IOC31	P31 I/O control					0	R/W	
		D0	IOC30	P30 I/O control					0	R/W	
Port function extension register	00402DF (B)	D7	CFEX7	P07 port extended function					1	#DMAEND3	
		D6	CFEX6	P06 port extended function	1	#DMAACK3	0	P06, etc.	0	R/W	
		D5	CFEX5	P05 port extended function	1	#DMAEND2	0	P05, etc.	0	R/W	
		D4	CFEX4	P04 port extended function	1	#DMAACK2	0	P04, etc.	0	R/W	
		D3	CFEX3	P31 port extended function	1	#GARD	0	P31, etc.	0	R/W	
		D2	CFEX2	P21 port extended function	1	#GAAS	0	P21, etc.	0	R/W	
		D1	CFEX1	P10, P11, P13 port extended function	1	DST0 DST1 DPCO	0	P10, etc. P11, etc. P13, etc.	1	R/W	
		D0	CFEX0	P12, P14 port extended function	1	DST2 DCLK	0	P12, etc. P14, etc.	1	R/W	

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Areas 18–15 set-up register	0048120 (HW)	DF	–	reserved		–	–	0 when being read.	
		DE	A18SZ	Areas 18–17 device size selection	1   8 bits   0   16 bits	0	R/W		
		DD	A18DF1	Areas 18–17	output disable delay time	A18DF[1:0]   Number of cycles		1	R/W
		DC	A18DF0			1   1   3.5	1		
						1   0   2.5			
						0   1   1.5			
						0   0   0.5			
		DB	–	reserved		–	–	–	0 when being read.
		DA	A18WT2	Areas 18–17 wait control	A18WT[2:0]   Wait cycles		1	R/W	
		D9	A18WT1			1   1   1   7	1		
		D8	A18WT0			1   1   0   6	1		
						1   0   1   5			
						1   0   0   4			
						0   1   1   3			
						0   1   0   2			
						0   0   1   1			
					0   0   0   0				
		D7	–	reserved		–	–	–	0 when being read.
		D6	A16SZ	Areas 16–15 device size selection	1   8 bits   0   16 bits	0	R/W		
		D5	A16DF1	Areas 16–15	output disable delay time	A16DF[1:0]   Number of cycles		1	R/W
D4	A16DF0		1   1   3.5	1					
			1   0   2.5						
			0   1   1.5						
				0   0   0.5					
D3	–	reserved		–	–	–	0 when being read.		
D2	A16WT2	Areas 16–15 wait control	A16WT[2:0]   Wait cycles		1	R/W			
D1	A16WT1			1   1   1   7	1				
D0	A16WT0			1   1   0   6	1				
				1   0   1   5					
				1   0   0   4					
				0   1   1   3					
				0   1   0   2					
				0   0   1   1					
			0   0   0   0						
Areas 14–13 set-up register	0048122 (HW)	DF–9	–	reserved		–	–	0 when being read.	
		D8	A14DRA	Area 14 DRAM selection	1   Used   0   Not used	0	R/W		
		D7	A13DRA	Area 13 DRAM selection	1   Used   0   Not used	0	R/W		
		D6	A14SZ	Areas 14–13 device size selection	1   8 bits   0   16 bits	0	R/W		
		D5	A14DF1	Areas 14–13	output disable delay time	A14DF[1:0]   Number of cycles		1	R/W
		D4	A14DF0			1   1   3.5	1		
						1   0   2.5			
						0   1   1.5			
				0   0   0.5					
D3	–	reserved		–	–	–	0 when being read.		
D2	A14WT2	Areas 14–13 wait control	A14WT[2:0]   Wait cycles		1	R/W			
D1	A14WT1			1   1   1   7	1				
D0	A14WT0			1   1   0   6	1				
				1   0   1   5					
				1   0   0   4					
				0   1   1   3					
				0   1   0   2					
				0   0   1   1					
			0   0   0   0						
Areas 12–11 set-up register	0048124 (HW)	DF–7	–	reserved		–	–	0 when being read.	
		D6	A12SZ	Areas 12–11 device size selection	1   8 bits   0   16 bits	0	R/W		
		D5	A12DF1	Areas 12–11	output disable delay time	A12DF[1:0]   Number of cycles		1	R/W
		D4	A12DF0			1   1   3.5	1		
						1   0   2.5			
						0   1   1.5			
						0   0   0.5			
		D3	–	reserved		–	–	–	0 when being read.
D2	A12WT2	Areas 12–11 wait control	A12WT[2:0]   Wait cycles		1	R/W			
D1	A12WT1			1   1   1   7	1				
D0	A12WT0			1   1   0   6	1				
				1   0   1   5					
				1   0   0   4					
				0   1   1   3					
				0   1   0   2					
				0   0   1   1					
			0   0   0   0						

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks							
Areas 10–9 set-up register	0048126 (HW)	DF	–	reserved	–	–	–	0 when being read.							
		DE	A10IR2	Area 10 internal ROM size selection	A10IR[2:0]	ROM size	1	R/W							
		DD	A10IR1		1 1 1	2MB	1								
		DC	A10IR0		1 1 0	1MB	1								
					1 0 1	512KB									
					1 0 0	256KB									
					0 1 1	128KB									
					0 1 0	64KB									
					0 0 1	32KB									
				0 0 0	16KB										
		DB	–	reserved	–	–			–	–					
		DA	A10BW1	Areas 10–9 burst ROM burst read cycle wait control	A10BW[1:0]	Wait cycles			0	R/W					
		D9	A10BW0		1 1	3			0						
					1 0	2									
					0 1	1									
					0 0	0									
		D8	A10DRA	Area 10 burst ROM selection	1 Used	0 Not used	0		R/W						
		D7	A9DRA	Area 9 burst ROM selection	1 Used	0 Not used	0	R/W							
		D6	A10SZ	Areas 10–9 device size selection	1 8 bits	0 16 bits	0	R/W							
D5	A10DF1	Areas 10–9 output disable delay time	A10DF[1:0]	Number of cycles	1	R/W									
D4	A10DF0		1 1	3.5	1										
			1 0	2.5											
			0 1	1.5											
			0 0	0.5											
D3	–	reserved	–	–		–	–				0 when being read.				
D2	A10WT2	Areas 10–9 wait control	A10WT[2:0]	Wait cycles		1	R/W								
D1	A10WT1		1 1 1	7		1									
D0	A10WT0		1 1 0	6		1									
			1 0 1	5											
			1 0 0	4											
			0 1 1	3											
			0 1 0	2											
			0 0 1	1											
		0 0 0	0												
Areas 8–7 set-up register	0048128 (HW)	DF–9	–	reserved				–	–					–	0 when being read.
		D8	A8DRA	Area 8 DRAM selection		1 Used		0 Not used	0					R/W	
		D7	A7DRA	Area 7 DRAM selection		1 Used		0 Not used	0	R/W					
		D6	A8SZ	Areas 8–7 device size selection		1 8 bits		0 16 bits	0	R/W					
		D5	A8DF1	Areas 8–7 output disable delay time		A8DF[1:0]		Number of cycles	1	R/W					
		D4	A8DF0			1 1		3.5	1						
					1 0	2.5									
					0 1	1.5									
					0 0	0.5									
		D3	–	reserved	–	–				–	–		0 when being read.		
		D2	A8WT2	Areas 8–7 wait control	A8WT[2:0]	Wait cycles			1	R/W					
D1	A8WT1	1 1 1	7		1										
D0	A8WT0	1 1 0	6		1										
		1 0 1	5												
		1 0 0	4												
		0 1 1	3												
		0 1 0	2												
		0 0 1	1												
		0 0 0	0												

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
Areas 6–4 set-up register	004812A (HW)	DF–E	–	reserved		–	–	0 when being read.				
		DD	A6DF1	Area 6	A6DF[1:0] Number of cycles 1 1 3.5 1 0 2.5 0 1 1.5 0 0 0.5	1	R/W					
		DC	A6DF0	output disable delay time								
		DB	–	reserved					–	–	0 when being read.	
		DA	A6WT2	Area 6 wait control					A6WT[2:0] Wait cycles	1	R/W	
		D9	A6WT1						1 1 1 7	1		
		D8	A6WT0		1 1 0 6	1						
					1 0 1 5							
					1 0 0 4							
					0 1 1 3							
					0 1 0 2							
		D7	–	reserved	–	–	–	0 when being read.				
		D6	A5SZ	Areas 5–4 device size selection	1 8 bits	0 16 bits	0	R/W				
		D5	A5DF1	Areas 5–4	A5DF[1:0] Number of cycles 1 1 3.5 1 0 2.5 0 1 1.5 0 0 0.5	1	R/W					
D4	A5DF0	output disable delay time										
D3	–	reserved	–	–					0 when being read.			
D2	A5WT2	Areas 5–4 wait control	A5WT[2:0] Wait cycles	1					R/W			
D1	A5WT1		1 1 1 7	1								
D0	A5WT0		1 1 0 6	1								
			1 0 1 5									
			1 0 0 4									
			0 1 1 3									
			0 1 0 2									
			0 0 1 1									
			0 0 0 0									
TTBR write protect register	004812D (B)	D7	TBRP7	TTBR register write protect	Writing 01011001 (0x59) removes the TTBR (0x48134) write protection. Writing other data sets the write protection.	0	W	Undefined in read.				
		D6	TBRP6			0						
		D5	TBRP5			0						
		D4	TBRP4			0						
		D3	TBRP3			0						
		D2	TBRP2			0						
		D1	TBRP1			0						
		D0	TBRP0			0						
Bus control register	004812E (HW)	DF	RBCLK	BCLK output control	1 Fixed at H	0 Enabled	0	R/W				
		DE	–	reserved	–	–	0	–	Writing 1 not allowed.			
		DD	RBST8	Burst ROM burst mode selection	1 8-successive	0 4-successive	0	R/W				
		DC	REDO	DRAM page mode selection	1 EDO	0 Fast page	0	R/W				
		DB	RCA1	Column address size selection	RCA[1:0] Size		0	R/W				
		DA	RCA0		1 1 11							
					1 0 10							
					0 1 9							
					0 0 8							
		D9	RPC2	Refresh enable	1 Enabled	0 Disabled	0	R/W				
		D8	RPC1	Refresh method selection	1 Self-refresh	0 CBR-refresh	0	R/W				
		D7	RPC0	Refresh RPC delay setup	1 2.0	0 1.0	0	R/W				
		D6	RAA1	Refresh RAS pulse width selection	RAA[1:0] Number of cycles		0	R/W				
		D5	RAA0		1 1 5							
		1 0 4										
		0 1 3										
		0 0 2										
D4	–	reserved	–	–	0	–	Writing 1 not allowed.					
D3	SBUSST	External interface method selection	1 #BSL	0 A0	0	R/W						
D2	SEMAS	External bus master setup	1 Existing	0 Nonexistent	0	R/W						
D1	SEPD	External power-down control	1 Enabled	0 Disabled	0	R/W						
D0	SWAITE	#WAIT enable	1 Enabled	0 Disabled	0	R/W						

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks					
DRAM timing set-up register	0048130 (HW)	DF-C	-	reserved		-	-	-	0 when being read.				
		DB	A3EEN	Area 3 emulation	1   Internal ROM   0   Emulation	1	R/W						
		DA	CEFUNC1	#CE pin function selection	CEFUNC[1:0]   #CE output	1   x   #CE7/8..#CE17/18 0   1   #CE6..#CE17 0   0   #CE4..#CE10	0	R/W					
		D9	CEFUNC0										
		D8	CRAS							Successive RAS mode setup	1   Successive   0   Normal	0	R/W
		D7	RPRC1	DRAM	RPRC[1:0]   Number of cycles	1   1   4 1   0   3 0   1   2 0   0   1	0	R/W					
		D6	RPRC0	RAS precharge cycles selection									
		D5	-	reserved						-	-	-	0 when being read.
		D4	CASC1	DRAM						CASC[1:0]   Number of cycles	1   1   4 1   0   3 0   1   2 0   0   1	0	R/W
		D3	CASC0	CAS cycles selection									
		D2	-	reserved	-	-	-	0 when being read.					
		D1	RASC1	DRAM	RASC[1:0]   Number of cycles	1   1   4 1   0   3 0   1   2 0   0   1	0	R/W					
		D0	RASC0	RAS cycles selection									
		DF	A18IO	Area 18, 17 internal/external access						1   Internal access	0   External access	0	R/W
DE	A16IO	Area 16, 15 internal/external access	1   Internal access	0   External access						0	R/W		
DD	A14IO	Area 14, 13 internal/external access											
DC	A12IO	Area 12, 11 internal/external access	1   Internal access	0   External access	0	R/W							
DB	-	reserved						-	-	-	0 when being read.		
DA	A8IO	Area 8, 7 internal/external access											
D9	A6IO	Area 6 internal/external access											
D8	A5IO	Area 5, 4 internal/external access											
D7	A18EC	Area 18, 17 endian control						1   Big endian	0   Little endian	0	R/W		
D6	A16EC	Area 16, 15 endian control						1   Big endian	0   Little endian	0	R/W		
D5	A14EC	Area 14, 13 endian control											
D4	A12EC	Area 12, 11 endian control											
D3	A10EC	Area 10, 9 endian control											
D2	A8EC	Area 8, 7 endian control											
D1	A6EC	Area 6 endian control											
D0	A5EC	Area 5, 4 endian control											
TTBR low-order register	0048134 (HW)	DF	TTBR15	Trap table base address [15:10]		0	R/W						
		DE	TTBR14										
		DD	TTBR13										
		DC	TTBR12										
		DB	TTBR11										
		DA	TTBR10										
		D9	TTBR09	Trap table base address [9:0]	Fixed at 0	0	R	0 when being read. Writing 1 not allowed.					
		D8	TTBR08										
		D7	TTBR07										
		D6	TTBR06										
		D5	TTBR05										
		D4	TTBR04										
		D3	TTBR03										
		D2	TTBR02										
		D1	TTBR01										
D0	TTBR00												

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
TTBR high-order register	0048136 (HW)	DF	TTBR33	Trap table base address [31:28]	Fixed at 0	0	R	0 when being read. Writing 1 not allowed.			
		DE	TTBR32								
		DD	TTBR31								
		DC	TTBR30								
	DB	TTBR2B	Trap table base address [27:16]	0x0C0	0	R/W					
	DA	TTBR2A									
	D9	TTBR29									
	D8	TTBR28									
	D7	TTBR27									
	D6	TTBR26									
	D5	TTBR25									
	D4	TTBR24									
	D3	TTBR23									
	D2	TTBR22									
	D1	TTBR21									
	D0	TTBR20									
	G/A read signal control register	0048138 (HW)	DF	A18AS	Area 18, 17 address strobe signal	1	Enabled	0	Disabled	0	R/W
DE			A16AS	Area 16, 15 address strobe signal							
DD			A14AS	Area 14, 13 address strobe signal							
DC			A12AS	Area 12, 11 address strobe signal							
DB			–	reserved	–	–	0	–	0 when being read.		
DA			A8AS	Area 8, 7 address strobe signal	1	Enabled	0	Disabled	0	R/W	
D9			A6AS	Area 6 address strobe signal							
D8			A5AS	Area 5, 4 address strobe signal							
D7			A18RD	Area 18, 17 read signal	1	Enabled	0	Disabled	0	R/W	
D6			A16RD	Area 16, 15 read signal							
D5			A14RD	Area 14, 13 read signal							
D4			A12RD	Area 12, 11 read signal							
D3			–	reserved							
D2			A8RD	Area 8, 7 read signal	1	Enabled	0	Disabled	0	R/W	
D1			A6RD	Area 6 read signal							
D0			A5RD	Area 5, 4 read signal							
BCLK select register			004813A (B)	D7–4	–	reserved	–	–	0	–	0 when being read.
	D3	A1X1MD		Area 1 access-speed	1	2 cycles	0	4 cycles	0	R/W	x2 speed mode only (Setting "0" is not allowed.)
	D2	–		reserved	–	–	0	–	0	–	0 when being read.
	D1	BCLKSEL1		BCLK output clock selection	BCLKSEL[1:0]		BCLK		0	R/W	
	D0	BCLKSELO			1	1	PLL_CLK	0			
			1	0	OSC3_CLK						
			0	1	BUS_CLK						
			0	0	CPU_CLK						
Read cycle hold time control register	004813C (B)	D7	A18RH	Areas 18–17 read hold cycle	1	Inserted	0	Not inserted	0	R/W	
		D6	A16RH	Areas 16–15 read hold cycle							
		D5	A14RH	Areas 14–13 read hold cycle							
		D4	A12RH	Areas 12–11 read hold cycle							
		D3	A10RH	Areas 10–9 read hold cycle							
		D2	A8RH	Areas 8–7 read hold cycle							
		D1	A6RH	Area 6 read hold cycle							
		D0	A5RH	Areas 5–4 read hold cycle							
Bus speed setting register	004813E (B)	D7	A18BS	Areas 18–17 bus speed selection	1	x1 speed	0	#X2SPD	0	R/W	This register's setting is meaningful when #X2SPD = 0 (2 × speed mode)
		D6	A16BS	Areas 16–15 bus speed selection							
		D5	A14BS	Areas 14–13 bus speed selection							
		D4	A12BS	Areas 12–11 bus speed selection							
		D3	A10BS	Areas 10– 9 bus speed selection							
		D2	A8BS	Areas 8–7 bus speed selection							
		D1	A6BS	Area 6 bus speed selection							
		D0	A5BS	Areas 5–4 bus speed selection							

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
16-bit timer 0 comparison data A set-up register	0048180 (HW)	DF	CR0A15	16-bit timer 0 comparison data A CR0A15 = MSB CR0A0 = LSB	0 to 65535	X	R/W				
		DE	CR0A14								
		DD	CR0A13								
		DC	CR0A12								
		DB	CR0A11								
		DA	CR0A10								
		D9	CR0A9								
		D8	CR0A8								
		D7	CR0A7								
		D6	CR0A6								
		D5	CR0A5								
		D4	CR0A4								
		D3	CR0A3								
		D2	CR0A2								
		D1	CR0A1								
		D0	CR0A0								
16-bit timer 0 comparison data B set-up register	0048182 (HW)	DF	CR0B15	16-bit timer 0 comparison data B CR0B15 = MSB CR0B0 = LSB	0 to 65535	X	R/W				
		DE	CR0B14								
		DD	CR0B13								
		DC	CR0B12								
		DB	CR0B11								
		DA	CR0B10								
		D9	CR0B9								
		D8	CR0B8								
		D7	CR0B7								
		D6	CR0B6								
		D5	CR0B5								
		D4	CR0B4								
		D3	CR0B3								
		D2	CR0B2								
		D1	CR0B1								
		D0	CR0B0								
16-bit timer 0 counter data register	0048184 (HW)	DF	TC015	16-bit timer 0 counter data TC015 = MSB TC00 = LSB	0 to 65535	X	R				
		DE	TC014								
		DD	TC013								
		DC	TC012								
		DB	TC011								
		DA	TC010								
		D9	TC09								
		D8	TC08								
		D7	TC07								
		D6	TC06								
		D5	TC05								
		D4	TC04								
		D3	TC03								
		D2	TC02								
		D1	TC01								
		D0	TC00								
16-bit timer 0 control register	0048186 (B)	D7	–	reserved	–	0	–	0 when being read.			
		D6	SELFM0	16-bit timer 0 fine mode selection	1	Fine mode	0	Normal	0	R/W	
		D5	SELCRB0	16-bit timer 0 comparison buffer	1	Enabled	0	Disabled	0	R/W	
		D4	OUTINV0	16-bit timer 0 output inversion	1	Invert	0	Normal	0	R/W	
		D3	CKSL0	16-bit timer 0 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	PTM0	16-bit timer 0 clock output control	1	On	0	Off	0	R/W	
		D1	PRESET0	16-bit timer 0 reset	1	Reset	0	Invalid	0	W	0 when being read.
		D0	PRUN0	16-bit timer 0 Run/Stop control	1	Run	0	Stop	0	R/W	

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 1 comparison data A set-up register	0048188 (HW)	DF	CR1A15	16-bit timer 1 comparison data A CR1A15 = MSB CR1A0 = LSB	0 to 65535	X	R/W		
		DE	CR1A14						
		DD	CR1A13						
		DC	CR1A12						
		DB	CR1A11						
		DA	CR1A10						
		D9	CR1A9						
		D8	CR1A8						
		D7	CR1A7						
		D6	CR1A6						
		D5	CR1A5						
		D4	CR1A4						
		D3	CR1A3						
		D2	CR1A2						
		D1	CR1A1						
		D0	CR1A0						
16-bit timer 1 comparison data B set-up register	004818A (HW)	DF	CR1B15	16-bit timer 1 comparison data B CR1B15 = MSB CR1B0 = LSB	0 to 65535	X	R/W		
		DE	CR1B14						
		DD	CR1B13						
		DC	CR1B12						
		DB	CR1B11						
		DA	CR1B10						
		D9	CR1B9						
		D8	CR1B8						
		D7	CR1B7						
		D6	CR1B6						
		D5	CR1B5						
		D4	CR1B4						
		D3	CR1B3						
		D2	CR1B2						
		D1	CR1B1						
		D0	CR1B0						
16-bit timer 1 counter data register	004818C (HW)	DF	TC115	16-bit timer 1 counter data TC115 = MSB TC10 = LSB	0 to 65535	X	R		
		DE	TC114						
		DD	TC113						
		DC	TC112						
		DB	TC111						
		DA	TC110						
		D9	TC19						
		D8	TC18						
		D7	TC17						
		D6	TC16						
		D5	TC15						
		D4	TC14						
		D3	TC13						
		D2	TC12						
		D1	TC11						
		D0	TC10						
16-bit timer 1 control register	004818E (B)	D7	–	reserved	–	0	–	0 when being read.	
		D6	SELF M1	16-bit timer 1 fine mode selection	1 Fine mode	0 Normal	0	R/W	
		D5	SEL CRB1	16-bit timer 1 comparison buffer	1 Enabled	0 Disabled	0	R/W	
		D4	OUT INV1	16-bit timer 1 output inversion	1 Invert	0 Normal	0	R/W	
		D3	CK SL1	16-bit timer 1 input clock selection	1 External clock	0 Internal clock	0	R/W	
		D2	PTM1	16-bit timer 1 clock output control	1 On	0 Off	0	R/W	
		D1	PRESET1	16-bit timer 1 reset	1 Reset	0 Invalid	0	W	0 when being read.
		D0	PRUN1	16-bit timer 1 Run/Stop control	1 Run	0 Stop	0	R/W	



Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
16-bit timer 2 comparison data A set-up register	0048190 (HW)	DF	CR2A15	16-bit timer 2 comparison data A CR2A15 = MSB CR2A0 = LSB	0 to 65535	X	R/W				
		DE	CR2A14								
		DD	CR2A13								
		DC	CR2A12								
		DB	CR2A11								
		DA	CR2A10								
		D9	CR2A9								
		D8	CR2A8								
		D7	CR2A7								
		D6	CR2A6								
		D5	CR2A5								
		D4	CR2A4								
		D3	CR2A3								
		D2	CR2A2								
		D1	CR2A1								
		D0	CR2A0								
16-bit timer 2 comparison data B set-up register	0048192 (HW)	DF	CR2B15	16-bit timer 2 comparison data B CR2B15 = MSB CR2B0 = LSB	0 to 65535	X	R/W				
		DE	CR2B14								
		DD	CR2B13								
		DC	CR2B12								
		DB	CR2B11								
		DA	CR2B10								
		D9	CR2B9								
		D8	CR2B8								
		D7	CR2B7								
		D6	CR2B6								
		D5	CR2B5								
		D4	CR2B4								
		D3	CR2B3								
		D2	CR2B2								
		D1	CR2B1								
		D0	CR2B0								
16-bit timer 2 counter data register	0048194 (HW)	DF	TC215	16-bit timer 2 counter data TC215 = MSB TC20 = LSB	0 to 65535	X	R				
		DE	TC214								
		DD	TC213								
		DC	TC212								
		DB	TC211								
		DA	TC210								
		D9	TC29								
		D8	TC28								
		D7	TC27								
		D6	TC26								
		D5	TC25								
		D4	TC24								
		D3	TC23								
		D2	TC22								
		D1	TC21								
		D0	TC20								
16-bit timer 2 control register	0048196 (B)	D7	–	reserved	–	0	–	0 when being read.			
		D6	SELFM2	16-bit timer 2 fine mode selection	1	Fine mode	0	Normal	0	R/W	
		D5	SELCRB2	16-bit timer 2 comparison buffer	1	Enabled	0	Disabled	0	R/W	
		D4	OUTINV2	16-bit timer 2 output inversion	1	Invert	0	Normal	0	R/W	
		D3	CKSL2	16-bit timer 2 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	PTM2	16-bit timer 2 clock output control	1	On	0	Off	0	R/W	
		D1	PRESET2	16-bit timer 2 reset	1	Reset	0	Invalid	0	W	0 when being read.
		D0	PRUN2	16-bit timer 2 Run/Stop control	1	Run	0	Stop	0	R/W	

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 3 comparison data A set-up register	0048198 (HW)	DF	CR3A15	16-bit timer 3 comparison data A CR3A15 = MSB CR3A0 = LSB	0 to 65535	X	R/W		
		DE	CR3A14						
		DD	CR3A13						
		DC	CR3A12						
		DB	CR3A11						
		DA	CR3A10						
		D9	CR3A9						
		D8	CR3A8						
		D7	CR3A7						
		D6	CR3A6						
		D5	CR3A5						
		D4	CR3A4						
		D3	CR3A3						
		D2	CR3A2						
		D1	CR3A1						
		D0	CR3A0						
16-bit timer 3 comparison data B set-up register	004819A (HW)	DF	CR3B15	16-bit timer 3 comparison data B CR3B15 = MSB CR3B0 = LSB	0 to 65535	X	R/W		
		DE	CR3B14						
		DD	CR3B13						
		DC	CR3B12						
		DB	CR3B11						
		DA	CR3B10						
		D9	CR3B9						
		D8	CR3B8						
		D7	CR3B7						
		D6	CR3B6						
		D5	CR3B5						
		D4	CR3B4						
		D3	CR3B3						
		D2	CR3B2						
		D1	CR3B1						
		D0	CR3B0						
16-bit timer 3 counter data register	004819C (HW)	DF	TC315	16-bit timer 3 counter data TC315 = MSB TC30 = LSB	0 to 65535	X	R		
		DE	TC314						
		DD	TC313						
		DC	TC312						
		DB	TC311						
		DA	TC310						
		D9	TC39						
		D8	TC38						
		D7	TC37						
		D6	TC36						
		D5	TC35						
		D4	TC34						
		D3	TC33						
		D2	TC32						
		D1	TC31						
		D0	TC30						
16-bit timer 3 control register	004819E (B)	D7	–	reserved	–	0	–	0 when being read.	
		D6	SELF3	16-bit timer 3 fine mode selection	1 Fine mode	0 Normal	0	R/W	
		D5	SELCRB3	16-bit timer 3 comparison buffer	1 Enabled	0 Disabled	0	R/W	
		D4	OUTINV3	16-bit timer 3 output inversion	1 Invert	0 Normal	0	R/W	
		D3	CKSL3	16-bit timer 3 input clock selection	1 External clock	0 Internal clock	0	R/W	
		D2	PTM3	16-bit timer 3 clock output control	1 On	0 Off	0	R/W	
		D1	PRESET3	16-bit timer 3 reset	1 Reset	0 Invalid	0	W	0 when being read.
		D0	PRUN3	16-bit timer 3 Run/Stop control	1 Run	0 Stop	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
16-bit timer 4 comparison data A set-up register	00481A0 (HW)	DF	CR4A15	16-bit timer 4 comparison data A	0 to 65535	X	R/W				
		DE	CR4A14	CR4A15 = MSB							
		DD	CR4A13	CR4A0 = LSB							
		DC	CR4A12								
		DB	CR4A11								
		DA	CR4A10								
		D9	CR4A9								
		D8	CR4A8								
		D7	CR4A7								
		D6	CR4A6								
		D5	CR4A5								
		D4	CR4A4								
		D3	CR4A3								
D2	CR4A2										
D1	CR4A1										
D0	CR4A0										
16-bit timer 4 comparison data B set-up register	00481A2 (HW)	DF	CR4B15	16-bit timer 4 comparison data B	0 to 65535	X	R/W				
		DE	CR4B14	CR4B15 = MSB							
		DD	CR4B13	CR4B0 = LSB							
		DC	CR4B12								
		DB	CR4B11								
		DA	CR4B10								
		D9	CR4B9								
		D8	CR4B8								
		D7	CR4B7								
		D6	CR4B6								
		D5	CR4B5								
		D4	CR4B4								
		D3	CR4B3								
D2	CR4B2										
D1	CR4B1										
D0	CR4B0										
16-bit timer 4 counter data register	00481A4 (HW)	DF	TC415	16-bit timer 4 counter data	0 to 65535	X	R				
		DE	TC414	TC415 = MSB							
		DD	TC413	TC40 = LSB							
		DC	TC412								
		DB	TC411								
		DA	TC410								
		D9	TC49								
		D8	TC48								
		D7	TC47								
		D6	TC46								
		D5	TC45								
		D4	TC44								
		D3	TC43								
D2	TC42										
D1	TC41										
D0	TC40										
16-bit timer 4 control register	00481A6 (B)	D7	–	reserved	–	0	–	0 when being read.			
		D6	SELFM4	16-bit timer 4 fine mode selection	1	Fine mode	0	Normal	0	R/W	
		D5	SELCRB4	16-bit timer 4 comparison buffer	1	Enabled	0	Disabled	0	R/W	
		D4	OUTINV4	16-bit timer 4 output inversion	1	Invert	0	Normal	0	R/W	
		D3	CKSL4	16-bit timer 4 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	PTM4	16-bit timer 4 clock output control	1	On	0	Off	0	R/W	
		D1	PRESET4	16-bit timer 4 reset	1	Reset	0	Invalid	0	W	0 when being read.
		D0	PRUN4	16-bit timer 4 Run/Stop control	1	Run	0	Stop	0	R/W	

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
16-bit timer 5 comparison data A set-up register	00481A8 (HW)	DF	CR5A15	16-bit timer 5 comparison data A	0 to 65535	X	R/W				
		DE	CR5A14	CR5A15 = MSB							
		DD	CR5A13	CR5A0 = LSB							
		DC	CR5A12								
		DB	CR5A11								
		DA	CR5A10								
		D9	CR5A9								
		D8	CR5A8								
		D7	CR5A7								
		D6	CR5A6								
		D5	CR5A5								
		D4	CR5A4								
		D3	CR5A3								
		D2	CR5A2								
		D1	CR5A1								
		D0	CR5A0								
16-bit timer 5 comparison data B set-up register	00481AA (HW)	DF	CR5B15	16-bit timer 5 comparison data B	0 to 65535	X	R/W				
		DE	CR5B14	CR5B15 = MSB							
		DD	CR5B13	CR5B0 = LSB							
		DC	CR5B12								
		DB	CR5B11								
		DA	CR5B10								
		D9	CR5B9								
		D8	CR5B8								
		D7	CR5B7								
		D6	CR5B6								
		D5	CR5B5								
		D4	CR5B4								
		D3	CR5B3								
		D2	CR5B2								
		D1	CR5B1								
		D0	CR5B0								
16-bit timer 5 counter data register	00481AC (HW)	DF	TC515	16-bit timer 5 counter data	0 to 65535	X	R				
		DE	TC514	TC515 = MSB							
		DD	TC513	TC50 = LSB							
		DC	TC512								
		DB	TC511								
		DA	TC510								
		D9	TC59								
		D8	TC58								
		D7	TC57								
		D6	TC56								
		D5	TC55								
		D4	TC54								
		D3	TC53								
		D2	TC52								
		D1	TC51								
		D0	TC50								
16-bit timer 5 control register	00481AE (B)	D7	–	reserved	–		0	–	0 when being read.		
		D6	SELF5	16-bit timer 5 fine mode selection	1	Fine mode	0	Normal	0	R/W	
		D5	SELCRB5	16-bit timer 5 comparison buffer	1	Enabled	0	Disabled	0	R/W	
		D4	OUTINV5	16-bit timer 5 output inversion	1	Invert	0	Normal	0	R/W	
		D3	CKSL5	16-bit timer 5 input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	PTM5	16-bit timer 5 clock output control	1	On	0	Off	0	R/W	
		D1	PRESET5	16-bit timer 5 reset	1	Reset	0	Invalid	0	W	0 when being read.
		D0	PRUN5	16-bit timer 5 Run/Stop control	1	Run	0	Stop	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IDMA base address low-order register	0048200 (HW)	DF	DBASEL15	IDMA base address		0	R/W	
		DE	DBASEL14	low-order 16 bits		0		
		DD	DBASEL13	(Initial value: 0x0C003A0)		0		
		DC	DBASEL12			0		
		DB	DBASEL11			0		
		DA	DBASEL10			0		
		D9	DBASEL9			1		
		D8	DBASEL8			1		
		D7	DBASEL7			1		
		D6	DBASEL6			0		
		D5	DBASEL5			1		
		D4	DBASEL4			0		
		D3	DBASEL3			0		
		D2	DBASEL2			0		
		D1	DBASEL1			0		
		D0	DBASEL0			0		
IDMA base address high-order register	0048202 (HW)	DF-C	–	reserved	–	–	–	Undefined in read.
		DB	DBASEH11	IDMA base address		0	R/W	
		DA	DBASEH10	high-order 12 bits		0		
		D9	DBASEH9	(Initial value: 0x0C003A0)		0		
		D8	DBASEH8			0		
		D7	DBASEH7			1		
		D6	DBASEH6			1		
		D5	DBASEH5			0		
		D4	DBASEH4			0		
		D3	DBASEH3			0		
		D2	DBASEH2			0		
		D1	DBASEH1			0		
		D0	DBASEH0			0		
IDMA start register	0048204 (B)	D7	DSTART	IDMA start	1 IDMA start 0 Stop	0	R/W	
		D6-0	DCHN	IDMA channel number	0 to 127	0	R/W	
IDMA enable register	0048205 (B)	D7-1	–	reserved	–	–	–	
		D0	IDMAEN	IDMA enable	1 Enabled 0 Disabled	0	R/W	
High-speed DMA Ch.0 transfer counter register	0048220 (HW)	DF	TC0_L7	Ch.0 transfer counter[7:0]		X	R/W	
		DE	TC0_L6	(block transfer mode)		X		
		DD	TC0_L5			X		
		DC	TC0_L4	Ch.0 transfer counter[15:8]		X		
		DB	TC0_L3	(single/successive transfer mode)		X		
		DA	TC0_L2			X		
		D9	TC0_L1			X		
		D8	TC0_L0			X		
		D7	BLKLEN07	Ch.0 block length		X	R/W	
		D6	BLKLEN06	(block transfer mode)		X		
		D5	BLKLEN05			X		
		D4	BLKLEN04	Ch.0 transfer counter[7:0]		X		
		D3	BLKLEN03	(single/successive transfer mode)		X		
		D2	BLKLEN02			X		
D1	BLKLEN01			X				
D0	BLKLEN00			X				
High-speed DMA Ch.0 control register	0048222 (HW)	DF	DUALM0	Ch.0 address mode selection	1 Dual addr 0 Single addr	0	R/W	
		DE	D0DIR	D) Invalid	–	–	–	
				S) Ch.0 transfer direction control	1 Memory WR 0 Memory RD	0	R/W	
		DD-8	–	reserved	–	–	–	Undefined in read.
		D7	TC0_H7	Ch.0 transfer counter[15:8]		X	R/W	
		D6	TC0_H6	(block transfer mode)		X		
		D5	TC0_H5			X		
		D4	TC0_H4	Ch.0 transfer counter[23:16]		X		
		D3	TC0_H3	(single/successive transfer mode)		X		
		D2	TC0_H2			X		
D1	TC0_H1			X				
D0	TC0_H0			X				

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks						
High-speed DMA Ch.0 low-order source address set-up register	0048224 (HW)	DF	S0ADRL15	D) Ch.0 source address[15:0]			X	R/W						
		DE	S0ADRL14	S) Ch.0 memory address[15:0]										
		DD	S0ADRL13											
		DC	S0ADRL12											
		DB	S0ADRL11											
		DA	S0ADRL10											
		D9	S0ADRL9											
		D8	S0ADRL8											
		D7	S0ADRL7											
		D6	S0ADRL6											
		D5	S0ADRL5											
		D4	S0ADRL4											
		D3	S0ADRL3											
		D2	S0ADRL2											
D1	S0ADRL1													
D0	S0ADRL0													
High-speed DMA Ch.0 high-order source address set-up register	0048226 (HW)	DF	--	reserved		--	--							
		DE	DATSIZE0	Ch.0 transfer data size	1   Half word	0   Byte	0	R/W						
		DD	S0IN1	D) Ch.0 source address control	S0IN[1:0]		Inc/dec	0	R/W					
		DC	S0IN0	S) Ch.0 memory address control	1   1		Inc.(no init)	0						
					1   0		Inc.(init)							
					0   1		Dec.(no init)							
					0   0		Fixed							
		DB	S0ADRH11	D) Ch.0 source address[27:16]				X	R/W					
		DA	S0ADRH10	S) Ch.0 memory address[27:16]										
		D9	S0ADRH9											
		D8	S0ADRH8											
		D7	S0ADRH7											
		D6	S0ADRH6											
		D5	S0ADRH5											
D4	S0ADRH4													
D3	S0ADRH3													
D2	S0ADRH2													
D1	S0ADRH1													
D0	S0ADRH0													
High-speed DMA Ch.0 low-order destination address set-up register	0048228 (HW)	DF	D0ADRL15	D) Ch.0 destination address[15:0]								X	R/W	
		DE	D0ADRL14	S) Invalid										
		DD	D0ADRL13											
		DC	D0ADRL12											
		DB	D0ADRL11											
		DA	D0ADRL10											
		D9	D0ADRL9											
		D8	D0ADRL8											
		D7	D0ADRL7											
		D6	D0ADRL6											
		D5	D0ADRL5											
		D4	D0ADRL4											
		D3	D0ADRL3											
		D2	D0ADRL2											
D1	D0ADRL1													
D0	D0ADRL0													
High-speed DMA Ch.0 high-order destination address set-up register	004822A (HW)	DF	D0MOD1	Ch.0 transfer mode	D0MOD[1:0]		Mode	0	R/W					
		DE	D0MOD0		1   1		Invalid	0						
					1   0		Block							
					0   1		Successive							
				0   0		Single								
		DD	D0IN1	D) Ch.0 destination address control	D0IN[1:0]		Inc/dec	0	R/W					
		DC	D0IN0	S) Invalid	1   1		Inc.(no init)	0						
					1   0		Inc.(init)							
					0   1		Dec.(no init)							
					0   0		Fixed							
		DB	D0ADRH11	D) Ch.0 destination address[27:16]				X	R/W					
		DA	D0ADRH10	S) Invalid										
		D9	D0ADRH9											
		D8	D0ADRH8											
D7	D0ADRH7													
D6	D0ADRH6													
D5	D0ADRH5													
D4	D0ADRH4													
D3	D0ADRH3													
D2	D0ADRH2													
D1	D0ADRH1													
D0	D0ADRH0													

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks					
High-speed DMA Ch.0 enable register	004822C (HW)	DF-1	–	reserved	–	–	–	Undefined in read.					
		D0	HS0_EN	Ch.0 enable	1   Enable   0   Disable	0	R/W						
High-speed DMA Ch.0 trigger flag register	004822E (HW)	DF-1	–	reserved	–	–	–	Undefined in read.					
		D0	HS0_TF	Ch.0 trigger flag clear (writing) Ch.0 trigger flag status (reading)	1   Clear   0   No operation 1   Set   0   Cleared	0	R/W						
High-speed DMA Ch.1 transfer counter register	0048230 (HW)	DF	TC1_L7	Ch.1 transfer counter[7:0] (block transfer mode)			X	R/W					
		DE	TC1_L6				X						
		DD	TC1_L5				X						
		DC	TC1_L4				X						
		DB	TC1_L3	Ch.1 transfer counter[15:8] (single/successive transfer mode)			X						
		DA	TC1_L2				X						
		D9	TC1_L1				X						
		D8	TC1_L0				X						
		D7	BLKLEN17	Ch.1 block length (block transfer mode)			X		R/W				
		D6	BLKLEN16				X						
		D5	BLKLEN15				X						
		D4	BLKLEN14				Ch.1 transfer counter[7:0] (single/successive transfer mode)			X			
		D3	BLKLEN13	X									
		D2	BLKLEN12	X									
		D1	BLKLEN11	X									
		D0	BLKLEN10	X									
High-speed DMA Ch.1 control register	0048232 (HW)	DF	DUALM1	Ch.1 address mode selection	1   Dual addr   0   Single addr	0	R/W						
		DE	D1DIR	D) Invalid S) Ch.1 transfer direction control	– 1   Memory WR   0   Memory RD	– 0	– R/W						
		DD-8	–	reserved	–	–	–	Undefined in read.					
		D7	TC1_H7	Ch.1 transfer counter[15:8] (block transfer mode)				X		R/W			
		D6	TC1_H6					X					
		D5	TC1_H5					X					
		D4	TC1_H4					Ch.1 transfer counter[23:16] (single/successive transfer mode)			X		
		D3	TC1_H3	X									
		D2	TC1_H2	X									
		D1	TC1_H1	X									
D0	TC1_H0	X											
High-speed DMA Ch.1 low-order source address set-up register	0048234 (HW)	DF	S1ADRL15	D) Ch.1 source address[15:0] S) Ch.1 memory address[15:0]								X	R/W
		DE	S1ADRL14									X	
		DD	S1ADRL13		X								
		DC	S1ADRL12		X								
		DB	S1ADRL11		X								
		DA	S1ADRL10		X								
		D9	S1ADRL9		X								
		D8	S1ADRL8		X								
		D7	S1ADRL7		X								
		D6	S1ADRL6		X								
		D5	S1ADRL5		X								
		D4	S1ADRL4		X								
		D3	S1ADRL3		X								
		D2	S1ADRL2		X								
D1	S1ADRL1	X											
D0	S1ADRL0	X											
High-speed DMA Ch.1 high-order source address set-up register	0048236 (HW)	DF	–	reserved	–	–	–						
		DE	DATSIZE1	Ch.1 transfer data size	1   Half word   0   Byte	0	R/W						
		DD	S1IN1	D) Ch.1 source address control	S1IN[1:0]	Inc/dec	0	R/W					
		DC	S1IN0	S) Ch.1 memory address control					1   1   Inc.(no init)				
				1   0   Inc.(init)									
				0   1   Dec.(no init)									
				0   0   Fixed									
		DB	S1ADRH11	D) Ch.1 source address[27:16] S) Ch.1 memory address[27:16]					X	R/W			
		DA	S1ADRH10						X				
		D9	S1ADRH9						X				
		D8	S1ADRH8						X				
		D7	S1ADRH7						X				
		D6	S1ADRH6						X				
		D5	S1ADRH5						X				
		D4	S1ADRH4						X				
		D3	S1ADRH3						X				
D2	S1ADRH2	X											
D1	S1ADRH1	X											
D0	S1ADRH0	X											

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks					
High-speed DMA Ch.1 low-order destination address set-up register  Note: D) Dual address mode S) Single address mode	0048238 (HW)	DF	D1ADRL15	D) Ch.1 destination address[15:0] S) Invalid			X	R/W					
		DE	D1ADRL14										
		DD	D1ADRL13										
		DC	D1ADRL12										
		DB	D1ADRL11										
		DA	D1ADRL10										
		D9	D1ADRL9										
		D8	D1ADRL8										
		D7	D1ADRL7										
		D6	D1ADRL6										
		D5	D1ADRL5										
		D4	D1ADRL4										
		D3	D1ADRL3										
		D2	D1ADRL2										
		D1	D1ADRL1										
D0	D1ADRL0												
High-speed DMA Ch.1 high-order destination address set-up register  Note: D) Dual address mode S) Single address mode	004823A (HW)	DF	D1MOD1	Ch.1 transfer mode	D1MOD[1:0]	Mode	0	R/W					
		DE	D1MOD0										
										1	1	Invalid	
										1	0	Block	
										0	1	Successive	
				0	0	Single							
				DD	D1IN1	D) Ch.1 destination address control S) Invalid	D1IN[1:0]	Inc/dec	0	R/W			
			DC	D1IN0									
					1							1	Inc.(no init)
					1							0	Inc.(init)
					0							1	Dec.(no init)
					0	0	Fixed						
				DB	D1ADRH11	D) Ch.1 destination address[27:16] S) Invalid			X	R/W			
			DA	D1ADRH10									
			D9	D1ADRH9									
	D8	D1ADRH8											
	D7	D1ADRH7											
	D6	D1ADRH6											
	D5	D1ADRH5											
	D4	D1ADRH4											
	D3	D1ADRH3											
	D2	D1ADRH2											
	D1	D1ADRH1											
	D0	D1ADRH0											
High-speed DMA Ch.1 enable register	004823C (HW)	DF-1	-	reserved	-	-	-	-	Undefined in read.				
		D0	HS1_EN	Ch.1 enable	1	Enable	0	Disable	0	R/W			
High-speed DMA Ch.1 trigger flag register	004823E (HW)	DF-1	-	reserved	-	-	-	-	Undefined in read.				
		D0	HS1_TF	Ch.1 trigger flag clear (writing) Ch.1 trigger flag status (reading)	1	Clear	0	No operation	0	R/W			
High-speed DMA Ch.2 transfer counter register	0048240 (HW)	DF	TC2_L7	Ch.2 transfer counter[7:0] (block transfer mode)			X	R/W					
		DE	TC2_L6										
		DD	TC2_L5										
		DC	TC2_L4										
		DB	TC2_L3										
		DA	TC2_L2										
		D9	TC2_L1										
		D8	TC2_L0										
				D7	BLKLEN27	Ch.2 block length (block transfer mode)		X	R/W				
				D6	BLKLEN26								
				D5	BLKLEN25								
				D4	BLKLEN24								
				D3	BLKLEN23								
				D2	BLKLEN22								
				D1	BLKLEN21								
		D0	BLKLEN20										
High-speed DMA Ch.2 control register  Note: D) Dual address mode S) Single address mode	0048242 (HW)	DF	DUALM2	Ch.2 address mode selection	1	Dual addr	0	Single addr	0	R/W			
				DE	D2DIR	D) Invalid	-	-	-	-	-		
						S) Ch.2 transfer direction control	1	Memory WR	0	Memory RD	0	R/W	
				DD-8	-	reserved	-	-	-	-	-	Undefined in read.	
				D7	TC2_H7	Ch.2 transfer counter[15:8] (block transfer mode)		X	R/W				
				D6	TC2_H6								
				D5	TC2_H5	Ch.2 transfer counter[23:16] (single/successive transfer mode)		X					
				D4	TC2_H4								
				D3	TC2_H3								
				D2	TC2_H2								
		D1	TC2_H1										
		D0	TC2_H0										



Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
High-speed DMA Ch.2 low-order source address set-up register	0048244 (HW)	DF	S2ADRL15	D) Ch.2 source address[15:0]			X	R/W			
		DE	S2ADRL14	S) Ch.2 memory address[15:0]			X				
		DD	S2ADRL13				X				
		DC	S2ADRL12				X				
		DB	S2ADRL11				X				
		DA	S2ADRL10				X				
		D9	S2ADRL9				X				
		D8	S2ADRL8				X				
		D7	S2ADRL7				X				
		D6	S2ADRL6				X				
		D5	S2ADRL5				X				
		D4	S2ADRL4				X				
		D3	S2ADRL3				X				
		D2	S2ADRL2				X				
D1	S2ADRL1				X						
D0	S2ADRL0				X						
High-speed DMA Ch.2 high-order source address set-up register	0048246 (HW)	DF	–	reserved		–	–				
		DE	DATSIZE2	Ch.2 transfer data size	1   Half word   0   Byte	0	R/W				
		DD	S2IN1	D) Ch.2 source address control	S2IN[1:0]	Inc/dec	0	R/W			
		DC	S2IN0	S) Ch.2 memory address control	1   1	Inc.(no init)	0				
					1   0	Inc.(init)					
					0   1	Dec.(no init)					
					0   0	Fixed					
		DB	S2ADRH11	D) Ch.2 source address[27:16]			X	R/W			
		DA	S2ADRH10	S) Ch.2 memory address[27:16]			X				
		D9	S2ADRH9				X				
		D8	S2ADRH8				X				
		D7	S2ADRH7				X				
		D6	S2ADRH6				X				
		D5	S2ADRH5				X				
D4	S2ADRH4				X						
D3	S2ADRH3				X						
D2	S2ADRH2				X						
D1	S2ADRH1				X						
D0	S2ADRH0				X						
High-speed DMA Ch.2 low-order destination address set-up register	0048248 (HW)	DF	D2ADRL15	D) Ch.2 destination address[15:0]			X	R/W			
		DE	D2ADRL14	S) Invalid			X				
		DD	D2ADRL13				X				
		DC	D2ADRL12				X				
		DB	D2ADRL11				X				
		DA	D2ADRL10				X				
		D9	D2ADRL9				X				
		D8	D2ADRL8				X				
		D7	D2ADRL7				X				
		D6	D2ADRL6				X				
		D5	D2ADRL5				X				
		D4	D2ADRL4				X				
		D3	D2ADRL3				X				
		D2	D2ADRL2				X				
D1	D2ADRL1				X						
D0	D2ADRL0				X						
High-speed DMA Ch.2 high-order destination address set-up register	004824A (HW)	DF	D2MOD1	Ch.2 transfer mode	D2MOD[1:0]	Mode	0	R/W			
		DE	D2MOD0				1   1			Invalid	0
		DC	D2IN0	D) Ch.2 destination address control	S) Invalid	D2IN[1:0]	Inc/dec	1   0	Block	0	R/W
								0   1	Successive		
								0   0	Single		
								1   1	Inc.(no init)	0	
		D9	D2ADRH9	S) Invalid			X	R/W			
		D8	D2ADRH8				X				
		D7	D2ADRH7				X				
		D6	D2ADRH6				X				
		D5	D2ADRH5				X				
		D4	D2ADRH4				X				
		D3	D2ADRH3				X				
		D2	D2ADRH2				X				
D1	D2ADRH1				X						
D0	D2ADRH0				X						

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks											
High-speed DMA Ch.2 enable register	004824C (HW)	DF-1	–	reserved	–	–	–	Undefined in read.											
		D0	HS2_EN	Ch.2 enable	1 Enable 0 Disable	0	R/W												
High-speed DMA Ch.2 trigger flag register	004824E (HW)	DF-1	–	reserved	–	–	–	Undefined in read.											
		D0	HS2_TF	Ch.2 trigger flag clear (writing) Ch.2 trigger flag status (reading)	1 Clear 0 No operation 1 Set 0 Cleared	0	R/W												
High-speed DMA Ch.3 transfer counter register	0048250 (HW)	DF	TC3_L7	Ch.3 transfer counter[7:0] (block transfer mode)				R/W											
		DE	TC3_L6																
		DD	TC3_L5																
		DC	TC3_L4																
		DB	TC3_L3																
		DA	TC3_L2																
		D9	TC3_L1																
		D8	TC3_L0																
		D7	BLKLEN37	Ch.3 block length (block transfer mode)									R/W						
		D6	BLKLEN36																
		D5	BLKLEN35																
		D4	BLKLEN34	Ch.3 transfer counter[7:0] (single/successive transfer mode)														R/W	
		D3	BLKLEN33																
		D2	BLKLEN32																
D1	BLKLEN31																		
D0	BLKLEN30																		
D0	BLKLEN30																		
High-speed DMA Ch.3 control register	0048252 (HW)	DF	DUALM3	Ch.3 address mode selection	1 Dual addr 0 Single addr	0	R/W												
		DE	D3DIR	D) Invalid S) Ch.3 transfer direction control	– 1 Memory WR 0 Memory RD	– 0	– R/W												
		DD-8	–	reserved	–	–	–	Undefined in read.											
		D7	TC3_H7	Ch.3 transfer counter[15:8] (block transfer mode)															R/W
		D6	TC3_H6																
		D5	TC3_H5	Ch.3 transfer counter[23:16] (single/successive transfer mode)									R/W						
		D4	TC3_H4																
		D3	TC3_H3																
		D2	TC3_H2																
		D1	TC3_H1																
D0	TC3_H0																		
D0	TC3_H0																		
High-speed DMA Ch.3 low-order source address set-up register	0048254 (HW)	DF	S3ADRL15											D) Ch.3 source address[15:0]				R/W	
		DE	S3ADRL14																
		DD	S3ADRL13																
		DC	S3ADRL12																
		DB	S3ADRL11																
		DA	S3ADRL10																
		D9	S3ADRL9																
		D8	S3ADRL8																
		D7	S3ADRL7																
		D6	S3ADRL6																
		D5	S3ADRL5																
		D4	S3ADRL4																
		D3	S3ADRL3																
		D2	S3ADRL2																
D1	S3ADRL1																		
D0	S3ADRL0																		
High-speed DMA Ch.3 high-order source address set-up register	0048256 (HW)	DF	–	reserved	–	–	–												
		DE	DATSIZE3	Ch.3 transfer data size	1 Half word 0 Byte	0	R/W												
		DD	S3IN1	D) Ch.3 source address control	S3IN[1:0] 1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed				R/W										
		DC	S3IN0																
		DC	S3IN0																
		DC	S3IN0																
		DB	S3ADRH11	D) Ch.3 source address[27:16]					R/W										
		DA	S3ADRH10																
		D9	S3ADRH9																
		D8	S3ADRH8																
D7	S3ADRH7																		
D6	S3ADRH6																		
D5	S3ADRH5																		
D4	S3ADRH4																		
D3	S3ADRH3																		
D2	S3ADRH2																		
D1	S3ADRH1																		
D0	S3ADRH0																		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
High-speed DMA Ch.3 low-order destination address set-up register  Note: D) Dual address mode S) Single address mode	0048258 (HW)	DF	D3ADRL15	D) Ch.3 destination address[15:0]			X	R/W		
		DE	D3ADRL14	S) Invalid						
		DD	D3ADRL13							
		DC	D3ADRL12							
		DB	D3ADRL11							
		DA	D3ADRL10							
		D9	D3ADRL9							
		D8	D3ADRL8							
		D7	D3ADRL7							
		D6	D3ADRL6							
		D5	D3ADRL5							
		D4	D3ADRL4							
		D3	D3ADRL3							
		D2	D3ADRL2							
		D1	D3ADRL1							
		D0	D3ADRL0							
High-speed DMA Ch.3 high-order destination address set-up register  Note: D) Dual address mode S) Single address mode	004825A (HW)	DF	D3MOD1	Ch.3 transfer mode	D3MOD[1:0]	Mode	0	R/W		
		DE	D3MOD0				1 1 Invalid			0
						1 0 Block				
						0 1 Successive				
						0 0 Single				
		DD	D3IN1	D) Ch.3 destination address control S) Invalid	D3IN[1:0]	Inc/dec	0	R/W		
		DC	D3IN0				1 1 Inc.(no init)			0
						1 0 Inc.(init)				
						0 1 Dec.(no init)				
						0 0 Fixed				
		DB	D3ADRH11	D) Ch.3 destination address[27:16] S) Invalid				X	R/W	
		DA	D3ADRH10							
		D9	D3ADRH9					X		
		D8	D3ADRH8					X		
		D7	D3ADRH7					X		
		D6	D3ADRH6					X		
D5	D3ADRH5					X				
D4	D3ADRH4					X				
D3	D3ADRH3					X				
D2	D3ADRH2					X				
D1	D3ADRH1					X				
D0	D3ADRH0					X				
High-speed DMA Ch.3 enable register	004825C (HW)	DF-1	--	reserved	--	--	--	Undefined in read.		
		D0	HS3_EN	Ch.3 enable	1 Enable 0 Disable	0	R/W			
High-speed DMA Ch.3 trigger flag register	004825E (HW)	DF-1	--	reserved	--	--	--	Undefined in read.		
		D0	HS3_TF	Ch.3 trigger flag clear (writing)	1 Clear 0 No operation	0	R/W			
				Ch.3 trigger flag status (reading)	1 Set 0 Cleared					

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Core ID register	0300000 (B)	D7	CID7	Chip core ID	0x02	0	R		
		D6	CID6	0x02: C33 standard macro core		0			
		D5	CID5	0x03: C33 mini-macro core		0			
		D4	CID4	0x04: C33 advanced macro core		0			
		D3	CID3			0			
		D2	CID2			0			
		D1	CID1			1			
		D0	CID0			0			
		Product series ID register	0300001 (B)	D7		MID7	Product series ID		0x03
D6	MID6			0x03: S1C333xx	0				
D5	MID5			0x15: S1C33Lxx	0				
D4	MID4				0				
D3	MID3				0				
D2	MID2				0				
D1	MID1				1				
D0	MID0				1				
Model ID register	0300002 (B)			D7	NAME7	Model ID	0x01	0	
		D6	NAME6	0x01: S1C33301	0				
		D5	NAME5		0				
		D4	NAME4		0				
		D3	NAME3		0				
		D2	NAME2		0				
		D1	NAME1		0				
		D0	NAME0		1				
		Version register	0300003 (B)	D7	VER3	Version code		0x00	0
D6	VER2			0x00: version 1.0	0				
D5	VER1				0				
D4	VER0				0				
D3-0	—			reserved	—	—	0 when being read.		
P4 I/O port data register	0300020 (B)	D7	P47D	P47 I/O port data	1 High	0 Low	0	R/W	
		D6	P46D	P46 I/O port data			0	R/W	
		D5	P45D	P45 I/O port data			0	R/W	
		D4	P44D	P44 I/O port data			0	R/W	
		D3	P43D	P43 I/O port data			0	R/W	
		D2	P42D	P42 I/O port data			0	R/W	
		D1	P41D	P41 I/O port data			0	R/W	
		D0	P40D	P40 I/O port data			0	R/W	
		P4 I/O control register	0300021 (B)	D7			IOC47	P47 I/O control	
D6	IOC46			P46 I/O control	0	R/W			
D5	IOC45			P45 I/O control	0	R/W			
D4	IOC44			P44 I/O control	0	R/W			
D3	IOC43			P43 I/O control	0	R/W			
D2	IOC42			P42 I/O control	0	R/W			
D1	IOC41			P41 I/O control	0	R/W			
D0	IOC40			P40 I/O control	0	R/W			
P5 I/O port data register	0300022 (B)			D7-6	—	reserved	1 High	0 Low	—
		D5	P55D	P55 I/O port data	0	R/W			
		D4	P54D	P54 I/O port data	0	R/W			
		D3	P53D	P53 I/O port data	0	R/W			
		D2	P52D	P52 I/O port data	0	R/W			
		D1	P51D	P51 I/O port data	0	R/W			
		D0	P50D	P50 I/O port data	0	R/W			
P5 I/O control register	0300023 (B)	D7-6	—	reserved	1 Output	0 Input	—	—	0 when being read. This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)
		D5	IOC55	P55 I/O control			0	R/W	
		D4	IOC54	P54 I/O control			0	R/W	
		D3	IOC53	P53 I/O control			0	R/W	
		D2	IOC52	P52 I/O control			0	R/W	
		D1	IOC51	P51 I/O control			0	R/W	
		D0	IOC50	P50 I/O control			0	R/W	
P6 I/O port data register	0300024 (B)	D7-4	—	reserved	1 High	0 Low	—	—	0 when being read.
		D3	P63D	P63 I/O port data			0	R/W	
		D2	P62D	P62 I/O port data			0	R/W	
		D1	P61D	P61 I/O port data			0	R/W	
		D0	P60D	P60 I/O port data			0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
P6 I/O control register	0300025 (B)	D7-4	–	reserved	–	–	–	0 when being read.			
		D3	IOC63	P63 I/O control	1 Output	0 Input	0	R/W	This register indicates the values of the I/O control signals of the ports when it is read. (See detailed explanation.)		
		D2	IOC62	P62 I/O control			0	R/W			
		D1	IOC61	P61 I/O control			0	R/W			
		D0	IOC60	P60 I/O control			0	R/W			
P00–P03 port function extension register	0300040 (B)	D7 D6	EFP031 EFP030	P03 port extended function	EFP03[1:0]		Function			0	R/W
					1	*	reserved	0			
							0	1	#FSRDY0		
						0	0	P03/#SRDY0			
		D5 D4	EFP021 EFP020	P02 port extended function	EFP02[1:0]		Function		0	R/W	
					1	*	reserved	0			
							0	1	#FSCLK0		
						0	0	P02/#SCLK0			
		D3 D2	EFP011 EFP010	P01 port extended function	EFP01[1:0]		Function		0	R/W	
					1	*	reserved	0			
							0	1	FSOUT0		
						0	0	P01/SOUT0			
D1 D0	EFP001 EFP000	P00 port extended function	EFP00[1:0]		Function		0	R/W			
			1	*	reserved	0					
					0	1	FSIN0				
				0	0	P00/SIN0					
P34–P35 port function extension register	0300047 (B)	D7-4	–	reserved	–	–	–	0 when being read.			
		D3 D2	EFP351 EFP350	P35 port extended function	EFP35[1:0]		Function		0	R/W	
					1	*	reserved	0			
							0	1	#SMRE		
						0	0	P35, etc.			
D1 D0	EFP341 EFP340	P34 port extended function	EFP34[1:0]		Function		0	R/W			
			1	*	reserved	0					
					0	1	#SMWE				
				0	0	P34, etc.					
P40–P43 port function extension register	0300048 (B)	D7 D6	EFP431 EFP430	P43 port extended function	EFP43[1:0]		Function		0	R/W	
					1	*	reserved	0			
							0	1	P43		
						0	0	A22			
		D5 D4	EFP421 EFP420	P42 port extended function	EFP42[1:0]		Function		0	R/W	
					1	*	reserved	0			
							0	1	P42		
						0	0	A23			
		D3 D2	EFP411 EFP410	P41 port extended function	EFP41[1:0]		Function		0	R/W	
					1	1	reserved	0			
						1	0	A24			
						0	1	P41			
				0	0	#LCAS					
D1 D0	EFP401 EFP400	P40 port extended function	EFP40[1:0]		Function		0	R/W			
			1	1	reserved	0					
				1	0	A25					
				0	1	P40					
				0	0	#HCAS					
P44–P47 port function extension register	0300049 (B)	D7 D6	EFP471 EFP470	P47 port extended function	EFP47[1:0]		Function		0	R/W	
					1	*	reserved	0			
							0	1	P47		
						0	0	A18			
		D5 D4	EFP461 EFP460	P46 port extended function	EFP46[1:0]		Function		0	R/W	
					1	*	reserved	0			
							0	1	P46		
						0	0	A19			
D3 D2	EFP451 EFP450	P45 port extended function	EFP45[1:0]		Function		0	R/W			
			1	*	reserved	0					
					0	1	P45				
				0	0	A20					
D1 D0	EFP441 EFP440	P44 port extended function	EFP44[1:0]		Function		0	R/W			
			1	*	reserved	0					
					0	1	P44				
				0	0	A21					

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
P50–P53 port function extension register	030004A (B)	D7	EFP531	P53 port extended function	EFP53[1:0]		Function	0	R/W		
		D6	EFP530		1	*	reserved	0			
					0	1	P53				
				0	0	#CE7, etc.					
		D5	EFP521	P52 port extended function	EFP52[1:0]		Function	0	R/W		
D4	EFP520	1	*		reserved	0					
		0	1		P52						
					0	0	#CE6, etc.				
		D3	EFP511	P51 port extended function	EFP51[1:0]		Function	0	R/W		
D2	EFP510	1	*		reserved	0					
		0	1		P51						
					0	0	#CE5, etc.				
		D1	EFP501	P50 port extended function	EFP50[1:0]		Function	0	R/W		
D0	EFP500	1	*		reserved	0					
		0	1		P50						
					0	0	#CE4, etc.				
P54–P55 port function extension register	030004B (B)	D7–4	–	reserved	–		–	–	0 when being read.		
		D3	EFP551	P55 port extended function	EFP55[1:0]		Function	0	R/W		
		D2	EFP550		1	*	reserved	0			
					0	1	P55				
				0	0	#CE9, etc.					
		D1	EFP541	P54 port extended function	EFP54[1:0]		Function	0	R/W		
D0	EFP540	1	*		reserved	0					
		0	1		P54						
					0	0	#CE8, etc.				
P60 port function extension register	030004C (B)	D7–2	–	reserved	–		–	–	0 when being read.		
		D1	EFP601	P60 port extended function	EFP60[1:0]		Function	0	R/W		
		D0	EFP600		1	1	reserved	0			
					1	0	FOSC1				
				0	1	P60					
					0	0	BCLK				
SmartMedia I/F #CE area select register	0300100 (B)	D7–2	–	reserved	–		–	–	0 when being read.		
		D1	SMCES1	SmartMedia I/F #CE area selection (according of CEFUNC)	SMCES[1:0]		#CE area	0	R/W		
		D0	SMCES0		1	1	#CE9/#CE17(+18)	0			
					1	0	#CE8/#CE14				
				0	1	#CE7/#CE13					
					0	0	#CE4/#CE11(+12)				
FIFO serial I/F Ch.0 transmit data register	0300200 (B)	D7	TXD07	FIFO serial I/F Ch.0 transmit data	0x0 to 0xFF(0x7F)		X	R/W	7-bit asynchronous mode does not use TXD07.		
		D6	TXD06	TXD07(06) = MSB							
		D5	TXD05	TXD00 = LSB							
		D4	TXD04								
		D3	TXD03								
		D2	TXD02								
		D1	TXD01								
		D0	TXD00								
FIFO serial I/F Ch.0 receive data register	0300201 (B)	D7	RXD07	FIFO serial I/F Ch.0 receive data	0x0 to 0xFF(0x7F)		X	R	7-bit asynchronous mode does not use RXD07 (fixed at 0).		
		D6	RXD06	RXD07(06) = MSB							
		D5	RXD05	RXD00 = LSB							
		D4	RXD04								
		D3	RXD03								
		D2	RXD02								
		D1	RXD01								
		D0	RXD00								
FIFO serial I/F Ch.0 status register	0300202 (B)	D7	RXD0NUM1	Number of Ch.0 receive data in FIFO	RXD0NUM[1:0]		Number of data		0	R/W	
		D6	RXD0NUM0		1	1	4	0			
					1	0	3				
					0	1	2				
					0	0	1 or 0				
		D5	TEND0	Ch.0 transmit-completion flag	1	Transmitting	0	End	0		R
		D4	FER0	Ch.0 flaming error flag	1	Error	0	Normal	0		R/W
		D3	PER0	Ch.0 parity error flag	1	Error	0	Normal	0		R/W
		D2	OER0	Ch.0 overrun error flag	1	Error	0	Normal	0		R/W
		D1	TDBE0	Ch.0 transmit data buffer status	1	Not full	0	Buffer full	1		R
D0	RDBF0	Ch.0 receive data buffer status	1	Not empty	0	Empty	0	R			

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks					
FIFO serial I/F Ch.0 control register	0300203 (B)	D7	TXENO	Ch.0 transmit enable	1	Enabled	0	Disabled	0	R/W				
		D6	RXENO	Ch.0 receive enable	1	Enabled	0	Disabled	0	R/W				
		D5	EPRO	Ch.0 parity enable	1	With parity	0	No parity	X	R/W	Valid only in asynchronous mode.			
		D4	PMD0	Ch.0 parity mode selection	1	Odd	0	Even	X	R/W				
		D3	STPB0	Ch.0 stop bit selection	1	2 bits	0	1 bit	X	R/W				
		D2	SCLK0	Ch.0 input clock selection	1	#FSCLK0	0	Internal clock	X	R/W				
		D1	SMD01	Ch.0 transfer mode selection	SMD0[1:0]	Transfer mode	X	X	R/W	X	R/W			
		D0	SMD00										1	1
		1	0										7-bit asynchronous	
		0	1										Clock sync. Slave	
				0	0	Clock sync. Master								
FIFO serial I/F Ch.0 IrDA register	0300204 (B)	D7	SRDYCTL0	Ch.0 #FSRDY control	1	High mask	0	Normal	0	R/W				
		D6	FIFOINT01	Ch.0 receive FIFO level setting	FIFOINT0[1:0]	Level	0	R/W	0	R/W				
		D5	FIFOINT00									1	1	4
												1	0	3
												0	1	2
						0	0	1						
		D4	DIVMD0	Ch.0 async. clock division ratio	1	1/8	0	1/16	X	R/W				
		D3	IRTL0	Ch.0 IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.			
		D2	IRRL0	Ch.0 IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W				
		D1	IRMD01	Ch.0 interface mode selection	IRMD0[1:0]	I/F mode	X	R/W	X	R/W				
D0	IRMD00	1	1									reserved		
		1	0									IrDA 1.0		
		0	1									reserved		
				0	0	General I/F								
FIFO serial I/F Ch.0 baud-rate timer control register	0300205 (B)	D7-1	--	reserved	--		--	--	0 when being read.					
		D0	BRTRUN	Baud-rate timer Run/Stop control	1	Run	0	Stop	0	R/W				
FIFO serial I/F Ch.0 baud-rate timer reload data register (LSB)	0300206 (B)	D7	BRTRD07	FIFO serial I/F Ch.0	0x0 to 0xFF		0	R/W						
		D6	BRTRD06	baud-rate timer reload data [7:0]	(BRTRD0[9:0] = 0x0 to 0x3FF)	0	0	R/W						
		D5	BRTRD05											
		D4	BRTRD04											
		D3	BRTRD03											
		D2	BRTRD02											
		D1	BRTRD01											
		D0	BRTRD00											
FIFO serial I/F Ch.0 baud-rate timer reload data register (MSB)	0300207 (B)	D7-2	--						reserved	--		--	--	0 when being read.
		D1	BRTRD09	FIFO serial I/F Ch.0	0x0 to 0x3		0	R/W						
		D0	BRTRD08	baud-rate timer reload data [9:8]	(BRTRD0[9:0] = 0x0 to 0x3FF)		0	R/W						
FIFO serial I/F Ch.0 baud-rate timer count data register (LSB)	0300208 (B)	D7	BRTCD07	FIFO serial I/F Ch.0	0x0 to 0xFF		0	R						
		D6	BRTCD06	baud-rate timer count data [7:0]	(BRTCD0[9:0] = 0x0 to 0x3FF)	0	0	R						
		D5	BRTCD05											
		D4	BRTCD04											
		D3	BRTCD03											
		D2	BRTCD02											
		D1	BRTCD01											
		D0	BRTCD00											
FIFO serial I/F Ch.0 baud-rate timer count data register (MSB)	0300209 (B)	D7-2	--						reserved	--		--	--	0 when being read.
		D1	BRTCD09	FIFO serial I/F Ch.0	0x0 to 0x3		0	R						
		D0	BRTCD08	baud-rate timer count data [9:8]	(BRTCD0[9:0] = 0x0 to 0x3FF)		0	R						
Bus signal pull-up control register	0300F00 (B)	D7-4	--	reserved	--		--	--	0 when being read.					
		D3	PUPCAS	#LCAS, #HCAS, BCLK (P40, P41, P60) pull-up	1	Pulled up	0	No pull-up	1	R/W				
		D2	PUPCE	#CE10-#CE4 (P55-P50) pull-up					1	R/W				
		D1	PUPAD	A23-A0 (P42-P47) pull-up					1	R/W				
		D0	PUPRW	#WRH, #WRL, #RD pull-up					1	R/W				

APPENDIX I/O MAP

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Bus signal low drive control register	0300F01 (B)	D7-5	–	reserved	–		–	–	0 when being read.	
		D4	LDRVDB	D15–D0 low drive	1	Low drive	0	Normal output	0	R/W
		D3	LDRVCAS	#LCAS, #HCAS, BCLK low drive					0	R/W
		D2	LDRVCE	#CE10–#CE4 low drive					0	R/W
		D1	LDRVAD	A23–A0 low drive					0	R/W
		D0	LDRVVRW	#WRH, #WRL, #RD low drive					0	R/W
Input port pull-up control register	0300F02 (B)	D7-3	–	reserved	–		–	–	0 when being read.	
		D2	PUPK6H	K67–K64 pull-up *1	1	Pulled up	0	No pull-up	1	R/W
		D1	PUPK6L	K63–K60 pull-up					1	R/W
		D0	PUPK5	K54–K50 pull-up					1	R/W
I/O port pull-up control register	0300F04 (B)	D7	–	reserved	–		–	–	0 when being read.	
		D6	PUPP6	P63–P60 pull-up *1	1	Pulled up	0	No pull-up	1	R/W
		D5-4	–	reserved					–	
		D3	PUPP3	P35–P30 pull-up	1	Pulled up	0	No pull-up	1	R/W
		D2	PUPP2	P27–P20 pull-up					1	R/W
		D1	PUPP1	P16–P10 pull-up					1	R/W
		D0	PUPP0	P07–P00 pull-up					1	R/W

\*1 In case of PFBGA-121pin package, PUPK6H (K67–K64 pull-up control) and PUPP6 (P63–P60 pull-up control) must be set to "1" (pulled up).



# EPSON International Sales Operations

---

## AMERICA

---

### EPSON ELECTRONICS AMERICA, INC.

#### - HEADQUARTERS -

150 River Oaks Parkway  
San Jose, CA 95134, U.S.A.  
Phone: +1-408-922-0200 Fax: +1-408-922-0238

#### - SALES OFFICES -

##### West

1960 E. Grand Avenue  
El Segundo, CA 90245, U.S.A.  
Phone: +1-310-955-5300 Fax: +1-310-955-5400

##### Central

101 Virginia Street, Suite 290  
Crystal Lake, IL 60014, U.S.A.  
Phone: +1-815-455-7630 Fax: +1-815-455-7633

##### Northeast

301 Edgewater Place, Suite 120  
Wakefield, MA 01880, U.S.A.  
Phone: +1-781-246-3600 Fax: +1-781-246-5443

##### Southeast

3010 Royal Blvd. South, Suite 170  
Alpharetta, GA 30005, U.S.A.  
Phone: +1-877-EEA-0020 Fax: +1-770-777-2637

## EUROPE

---

### EPSON EUROPE ELECTRONICS GmbH

#### - HEADQUARTERS -

Riesstrasse 15  
80992 Munich, GERMANY  
Phone: +49-(0)89-14005-0 Fax: +49-(0)89-14005-110

#### DÜSSELDORF BRANCH OFFICE

Altstadtstrasse 176  
51379 Leverkusen, GERMANY  
Phone: +49-(0)2171-5045-0 Fax: +49-(0)2171-5045-10

#### UK & IRELAND BRANCH OFFICE

Unit 2.4, Doncastle House, Doncastle Road  
Bracknell, Berkshire RG12 8PE, ENGLAND  
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

#### FRENCH BRANCH OFFICE

1 Avenue de l'Atlantique, LP 915 Les Conquerants  
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE  
Phone: +33-(0)1-64862350 Fax: +33-(0)1-64862355

#### BARCELONA BRANCH OFFICE

##### Barcelona Design Center

Edificio Testa, Avda. Alcalde Barrils num. 64-68  
E-08190 Sant Cugat del Vallès, SPAIN  
Phone: +34-93-544-2490 Fax: +34-93-544-2491

##### Scotland Design Center

Integration House, The Alba Campus  
Livingston West Lothian, EH54 7EG, SCOTLAND  
Phone: +44-1506-605040 Fax: +44-1506-605041

## ASIA

---

### EPSON (CHINA) CO., LTD.

23F, Beijing Silver Tower 2# North RD DongSanHuan  
ChaoYang District, Beijing, CHINA  
Phone: 64106655 Fax: 64107319

#### SHANGHAI BRANCH

7F, High-Tech Bldg., 900, Yishan Road  
Shanghai 200233, CHINA  
Phone: 86-21-5423-5577 Fax: 86-21-5423-4677

### EPSON HONG KONG LTD.

20/F., Harbour Centre, 25 Harbour Road  
Wanchai, Hong Kong  
Phone: +852-2585-4600 Fax: +852-2827-4346  
Telex: 65542 EPSCO HX

### EPSON TAIWAN TECHNOLOGY & TRADING LTD.

14F, No. 7, Song Ren Road, Taipei 110  
Phone: 02-8786-6688 Fax: 02-8786-6660

#### HSINCHU OFFICE

13F-3, No. 295, Kuang-Fu Road, Sec. 2  
HsinChu 300  
Phone: 03-573-9900 Fax: 03-573-9169

### EPSON SINGAPORE PTE., LTD.

No. 1 Temasek Avenue, #36-00  
Millenia Tower, SINGAPORE 039192  
Phone: +65-6337-7911 Fax: +65-6334-2716

### SEIKO EPSON CORPORATION KOREA OFFICE

50F, KLI 63 Bldg., 60 Yoido-dong  
Youngdeungpo-Ku, Seoul, 150-763, KOREA  
Phone: 02-784-6027 Fax: 02-767-3677

#### GUMI OFFICE

6F, Good Morning Securities Bldg.  
56 Songjeong-Dong, Gumi-City, 730-090, KOREA  
Phone: 054-454-6027 Fax: 054-454-6093

### SEIKO EPSON CORPORATION

#### ELECTRONIC DEVICES MARKETING DIVISION

##### IC Marketing Department

##### IC Marketing & Engineering Group

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN  
Phone: +81-(0)42-587-5816 Fax: +81-(0)42-587-5624

##### ED International Marketing Department

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN  
Phone: +81-(0)42-587-5814 Fax: +81-(0)42-587-5117

**SEIKO EPSON CORPORATION**  
**ELECTRONIC DEVICES MARKETING DIVISION**

■ EPSON Electronic Devices Website

<http://www.epsondevice.com>