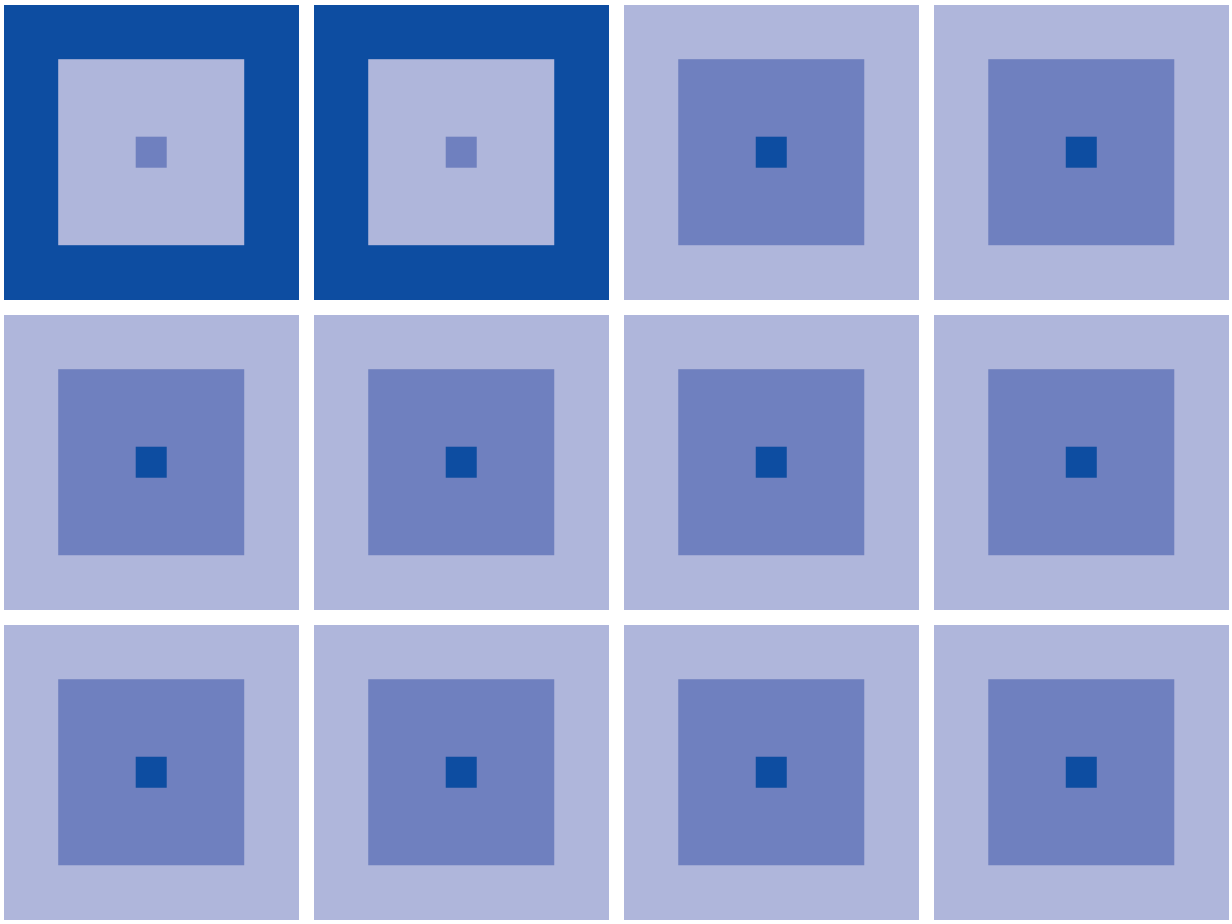
EPSON

CMOS 32-BIT SINGLE CHIP MICROCOMPUTER

S1C33L05

Technical Manual



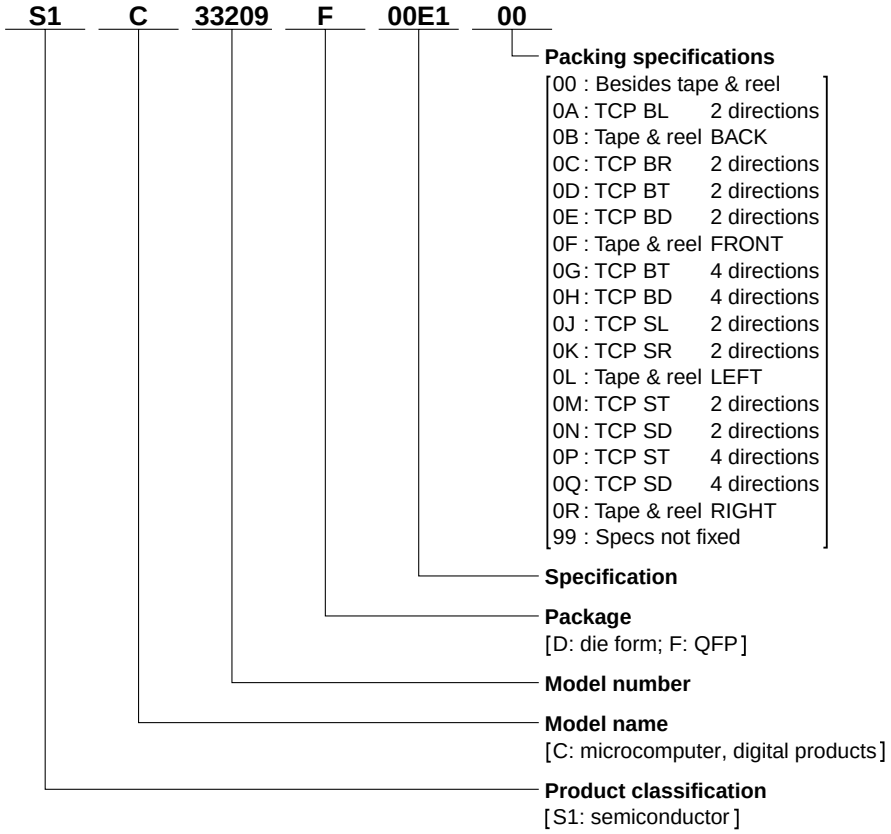
SEIKO EPSON CORPORATION

NOTICE

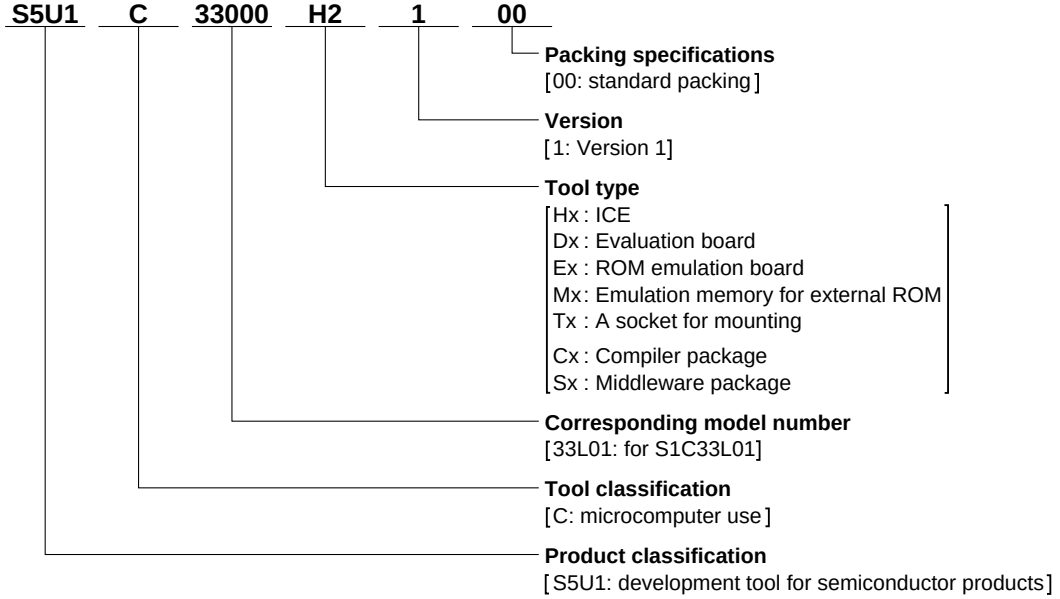
No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained 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



Development tools



S1C33L05 Technical Manual

CONTENTS

I S1C33L05

III PERIPHERAL BLOCK

IV ANALOG BLOCK

V DMA BLOCK

VI USB BLOCK

VII LCDC BLOCK

Appendix

S1C33L05 Technical Manual

I S1C33L05

I-1 Outline

The S1C33L05 is a Seiko Epson original 32-bit microcomputer that features high speed, low power consumption, and low-voltage operation. The S1C33L05 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, a color STN LCD controller that supports 64K color display, SDRAM controller, USB1.1 function controller, sequential ROM interface, MMC (SPI mode) interface and NAND flash interface, and also an embedded RAM. Two oscillation circuits and a PLL are also included, supporting advanced operation, power-saving operation, and high-performance realtime clock functions. The S1C33L05 is ideal for portable products that require high-speed data processing. Especially it is suitable for the application processor embedded in PDAs, electronic dictionary and e-Book readers.

Table I.1.1 Model Lineup

Model	Package	Internal RAM	Internal VRAM	Internal ROM
S1C33L05F00	QFP21-176pin	16K bytes	40K bytes	None
S1C33L05D00	Die form (167 pads)	16K bytes	40K bytes	None

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
- 20.83 ns of minimum instruction execution time at 48 MHz operation

Internal memory

General-purpose RAM: 16K bytes (1-cycle-access)
Video-RAM: 40K bytes (usable for general-purpose RAM, 2-cycle-access)

Internal peripheral circuits

OSC3 oscillation circuit/PLL: When PLL is disabled
- Crystal oscillator: 5 MHz min. to 48 MHz max.
- Ceramic oscillator: 48 MHz (fixed)
- External clock input: 2 MHz min. to 48 MHz max.

When PLL is enabled
- Crystal oscillator: 20 MHz min. to 48 MHz max.
- Ceramic oscillator: 48 MHz (fixed)
- External clock input: 20 MHz min. to 48 MHz max.

Generates the main clock for the bus and the CPU.
The software controllable PLL multiplies the high-speed (OSC3) oscillation clock frequency.
- PLL input clock: 10 MHz min. to 24 MHz max.
- PLL output clock: 10 MHz min. to 48 MHz max.

OSC1 oscillation circuit: Crystal oscillator or external clock input 32.768 kHz typ.
Generates the source clock for the realtime clock function, etc.

Timers:
- 8-bit timer: 6 channels
- 16-bit timer: 6 channels
- Watchdog timer: 1 channel (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-byte receive-data buffer and a 2-byte transmit-data buffer) and no buffer type

A/D converter: 10 bits × 5 channels

LCD controller: 4 or 8-bit monochrome/color LCD interface
Panels supported
- Single-panel, single drive passive display
- 4/8-bit monochrome LCD interface
- 4/8-bit color LCD interface

Display modes
- 16-bpp mode: 64K colors or 64-level gray scale display
- 12-bpp mode: 4096 colors or 16-level gray scale display
- 8-bpp mode: 256 colors or 64-level gray scale display
- 4-bpp mode: 16 colors or 16-level gray scale display
- 2-bpp mode: 4 colors or 4-level gray scale display
- 1-bpp mode: 2 colors or 2-level gray scale display

∗ A 256 × 3 × 6-bit Look-Up Table (256K-color palette) is provided for displaying 256 colors simultaneously. The LUT can be bypassed to send display data from VRAM directly to the LCD.

∗ Gray scale display uses FRM (Frame Rate Modulation) and dithering.

Resolution (programmable)
Typical resolutions when only the internal VRAM is used:
- 320 × 240 pixels in 4-bpp mode
- 160 × 240 pixels in 8-bpp mode
- 160 × 160 pixels in 12-bpp mode

Typical resolutions when an external VRAM is used via the UMA:
- 320 × 240 pixels in 8-bpp mode
- 320 × 240 pixels in 16-bpp mode

SDRAM controller: 48 MHz synchronous clock max.
Supports up to 256M-bit (32MB) SDRAM with 16-bit data width.
16-stage IQB (32-byte Instruction Queue Buffer) and 2-stage DQB (4-byte Data Queue Buffer) are provided.
Allows LCDC DMA controller to access SDRAM directly as an external VRAM.

MMC (SPI mode) interface: 1 channel
Supports 1 to 16-bit serial data transfer in master mode.
Compatible with MMC.

NAND flash interface: Generates the #SMWE and #SMRE signals using the BCU signals to interface directly with SmartMedia cards or NAND flash memories.
Supports 8/16-bit Nand flash devices. Also the Nand flash booting function and the ECC function when a Nand flash is read/written are supported.

Sequential ROM interface: Supports MX23L12813 (manufactured by Macronix International Co., Ltd.).
Generates the SQUALE, SQLALE and #SQRD signals using the BCU signals to interface directly with the sequential mask ROM.

USB1.1 function controller: Endpoint: EP0, EPa, EPb, EPc, EPd (4 channels); FIFO: 1,024 bytes

DMA controller:
High-speed DMA 4 channels
High-speed DMA Ch. 3 has been reserved for the internal USB1.1 function controller.
Intelligent DMA 128 channels

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
	LCD controller interrupt	1 type
	SPI interrupt	1 type
	USB function controller interrupt	1 type
General-purpose input and output ports:	Shared with the I/O pins for internal peripheral circuits Input port 9 bits (max.) I/O port 69 bits (max.) ∗ The K54 and K65–K67 pins are not available in the S1C33L05. ∗ Two LED direct output (8 mA) ports (P27 and P26) are available. ∗ The number of the ports varies depending on the peripheral functions used.	

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/big-endian memory access; endian type 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_{DD})	1.65 V to 1.95 V (1.8 V ±0.15 V) (when crystal oscillator is used) 1.70 V to 1.90 V (1.8 V ±0.10 V) (when ceramic oscillator is used)
	I/O (V_{DDE}, AV_{DDE})	2.70 V to 3.60 V (when USB is not used) 3.00 V to 3.60 V (when USB is used)
Operating clock frequency:	CPU	48 MHz max. Note 1
	Bus (BCU)	40 MHz max.
	LCD controller	48 MHz max.
	USB function controller	48 MHz
	SDRAM	48 MHz
Operating temperature:	-40 to 85°C	(when crystal oscillator is used)
	0 to 70°C	(when ceramic oscillator is used)
Power consumption:	During SLEEP	12 µW typ.
	During HALT	18 mW typ. (48 MHz, LCDC and USB not included)
	During execution	42 mW typ. Note 2 (48 MHz, LCDC and USB not included)
	LCD controller - During display	1.8 mW typ. (LCDC clock = 8 MHz, 16 bpp, IVRAM mode, V_{DD}, LCDC block only)
	USB controller - Idle state	14 mW typ. (V_{DD}, USB block only)

Supply form

QFP21-176pin plastic package (24 mm × 24 mm × 1.4 mm, 0.5-mm pitch)
or die form 167-PAD (5.25 mm × 4.85 mm, 100 µm pitch)

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.

THIS PAGE IS BLANK.

I-2 Block Diagram

Macro Block Diagram

The S1C33L05 consists of eight major blocks: C33 Core Block, C33 Peripheral Block, C33 Analog Block, C33 DMA Block, C33 Internal Memory Block, LCDC Block, SDRAMC Block and USB Block. Although the S1C33L05 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 macro blocks.

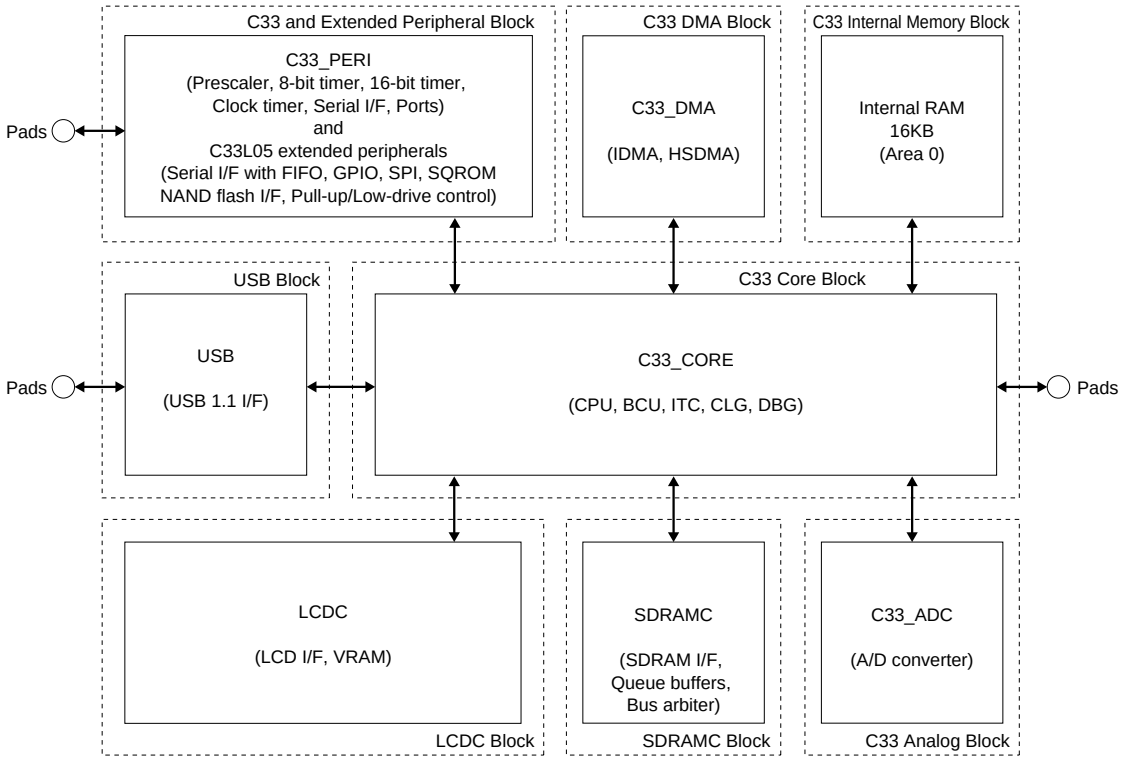


Figure I.2.1 Macro Block Configuration

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)	53 types of interrupts are available.
CLG (Clock Generator)	OSC3 oscillation circuit (48 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	9 bits of input ports and 29 bits of I/O ports (used for peripheral I/O)
Clock timer	1 channel with alarm function

S1C33L05 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.
NAND flash interface	Generates the #SMWE and #SMRE signals using the BCU signals to interface directly with SmartMedia cards and NAND flash memories.
MMC (SPI mode) interface	Supports 1 to 16-bit serial data transfer in master mode.
Sequential ROM interface	Generates the SQUALE, SQLALE and #SQRD signals to interface directly with the sequential mask ROM (MX23L12813).
I/O ports	I/O ports are extended to 69 bits (including 29 bits of standard macro I/O ports).
Pull-up/low-drive control circuit	Controls the pull-up resistors at the bus and I/O port pins and drives the bus and ports low.
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 5 input channels The S1C33L05 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 High-speed DMA Ch. 3 has been reserved for the internal USB1.1 function controller.
IDMA (Intelligent DMA)	128 channels

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

C33 Memory Block

RAM	A 16KB SRAM built-in

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

USB Block

USB1.1 function controller	Provides a USB1.1 interface that supports control, bulk, isochronous and interrupt transfers.

Refer to Chapter VI, "USB Block", for details.

LCDC Block

LCD controller	Provides a passive LCD interface for a 4 or 8-bit LCD panel, 40KB built-in VRAM or 1MB external VRAM via UMA.

Refer to Chapter VII, "LCDC Block", for details.

SDRAMC Block

SDRAM controller	Up to a 256M-bit SDRAM (32MB) can be connected directly. 16-stage IQB (32-byte Instruction Queue Buffer) and 2-stage DQB (4-byte Data Queue Buffer) are provided.

Refer to Chapter VIII, "SDRAMC Block", for details.

Functional Block Diagram

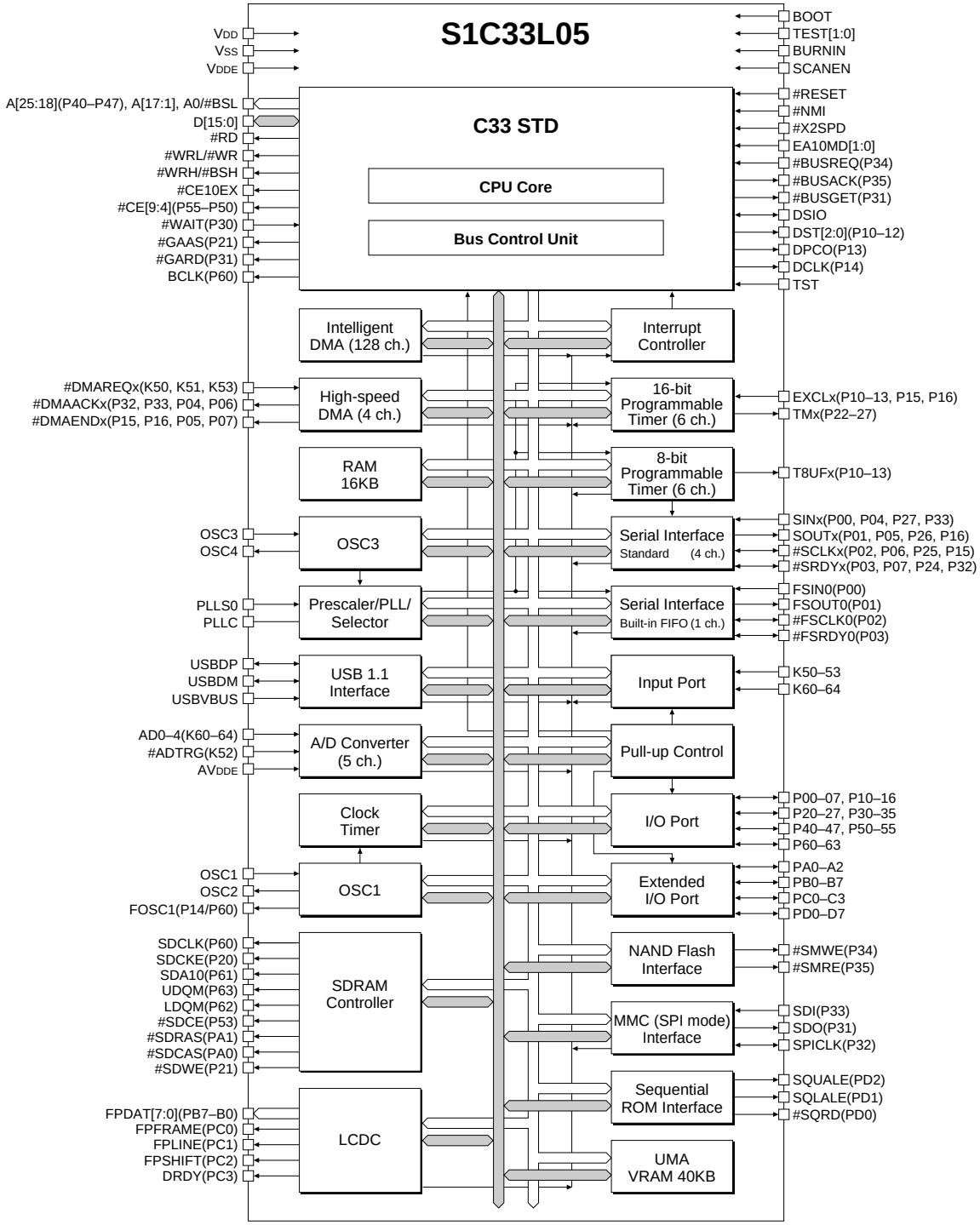


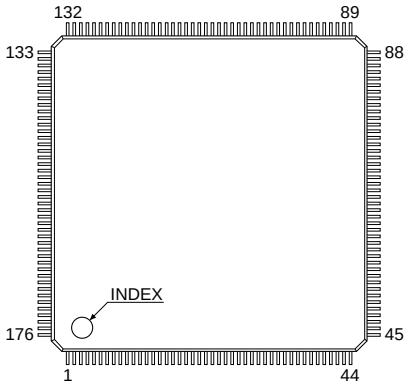
Figure I.2.2 S1C33L05 Functional Block Diagram

THIS PAGE IS BLANK.

I-3 Pin Description

Pin Layout Diagram (plastic package)

QFP21-176pin



No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
1	**D9**	45	**P15**/EXCL4/#DMAEND0/#SCLK3	89	**P01**/SOUT0/FSOUT0	133	**A23**/P42
2	**D8**	46	N.C.	90	**P00**/SIN0/FSIN0	134	N.C.
3	**VDDE**	47	**DSIO**	91	**USBDP**	135	**A22**/P43
4	**D7**	48	**VDDE**	92	**USBDM**	136	**A21**/P44
5	**D6**	49	**DCLK**/P14/FOSC1	93	N.C.	137	**A20**/P45
6	**D5**	50	**DPCO**/P13/EXCL3/T8UF3	94	**USBVBUS**	138	**A19**/P46
7	**D4**	51	**DST2**/P12/EXCL2/T8UF2	95	**VDDE**	139	**A18**/P47
8	**D3**	52	**DST1**/P11/EXCL1/T8UF1	96	**P31**/#BUSGET/#GARD/SDO	140	**A17**
9	**D2**	53	**DST0**/P10/EXCL0/T8UF0	97	**P32**/#DMAACK0/#SRDY3/SPICLK	141	**A16**
10	**D1**	54	**VDD**	98	**VSS**	142	**VDDE**
11	**D0**	55	**#NMI**	99	**P33**/#DMAACK1/SIN3/SDI	143	**A15**
12	**VSS**	56	**#RESET**	100	**P34**/#BUSREQ/#CE6/#SMWE	144	**A14**
13	**P30**/#WAIT/#CE4&5/PA2	57	N.C.	101	**P35**/#BUSACK/#SMRE	145	N.C.
14	**PD0**/#SQRD	58	**VSS**	102	**VDD**	146	**A13**
15	**PD1**/SQLALE	59	**K60**/AD0	103	**#X2SPD**	147	**A12**
16	**PD2**/SQUALE	60	**K61**/AD1	104	**EA10MD0**	148	**VSS**
17	**PD3**	61	**K62**/AD2	105	**EA10MD1**	149	**A11**
18	**PD4**	62	**K63**/AD3	106	**VDDE**	150	**A10**
19	**PD5**	63	**K64**/AD4	107	**PLLC**	151	**VDD**
20	**PD6**	64	**TEST0**	108	**VSS**	152	**A9**
21	**PD7**	65	**AVDDE**	109	**PLLS0**	153	**A8**
22	**VDDE**	66	**K53**/#DMAREQ2	110	**TST**	154	**A7**
23	**P22**/TM0	67	**K52**/#ADTRG	111	**BOOT**	155	**A6**
24	**P23**/TM1	68	**K51**/#DMAREQ1	112	**#CE4**/#CE11/#CE11&12/P50	156	**A5**
25	**P24**/TM2/#SRDY2	69	**K50**/#DMAREQ0	113	**VSS**	157	**VSS**
26	**P25**/TM3/#SCLK2	70	**VSS**	114	**#CE5**/#CE15/#CE15&16/P51	158	**A4**
27	**P26**/TM4/SOUT2	71	**OSC1**	115	**#CE6**/#CE7&8/P52	159	**A3**
28	**P27**/TM5/SIN2	72	**OSC2**	116	**#CE7**/#RAS0/#CE13/#RAS2/P53/#SDCE	160	**VDDE**
29	**VSS**	73	**VDDE**	117	**#CE8**/#RAS1/#CE14/#RAS3/P54	161	**A2**
30	**PB7**/FPDAT7	74	**BURNIN**	118	**#CE9**/#CE17/#CE17&18/P55	162	**A1**
31	**PB6**/FPDAT6	75	**SCANEN**	119	**#CE10EX**/#CE9&10EX	163	**A0**/#BSL
32	**PB5**/FPDAT5	76	**TEST1**	120	**P61**/SDA10	164	**#WRH**/#BSH
33	**PB4**/FPDAT4	77	N.C.	121	**P62**/LDQM	165	N.C.
34	**VDD**	78	**VDD**	122	**P63**/UDQM	166	**#WRL**/#WR/#WE
35	**PB3**/FPDAT3	79	**OSC3**	123	**P21**/#DWE/#GAAS/#SDWE	167	**#RD**
36	**PB2**/FPDAT2	80	**OSC4**	124	**#LCAS**/PA0/#SDCAS	168	**VSS**
37	**PB1**/FPDAT1	81	**VSS**	125	**VDD**	169	**D15**
38	**PB0**/FPDAT0	82	**P07**/#SRDY1/#DMAEND3	126	**#HCAS**/PA1/#SDRAS	170	**D14**
39	**PC3**/DRDY	83	**P06**/#SCLK1/#DMAACK3	127	**P20**/#DRD/SDCKE	171	**D13**
40	**PC2**/FPSHIFT	84	**P05**/#SOUT1/#DMAEND2	128	**VDDE**	172	**D12**
41	**PC1**/FPLINE	85	**P04**/#SIN1/#DMAACK2	129	**BCLK**/P60/FOSC1/SDCLK	173	**D11**
42	**PC0**/FPFRAME	86	**P03**/#SRDY0/#FSRDY0	130	**A25**/P40	174	**VDD**
43	**VSS**	87	**P02**/#SCLK0/#FSCLK0	131	**VSS**	175	**D10**
44	**P16**/EXCL5/#DMAEND1/SOUT3	88	N.C.	132	**A24**/P41	176	N.C.

Boldface: Default setting

Figure I.3.1 Pin Layout Diagram (QFP21-176pin)

Pin Functions

Table I.3.1 List of Pins for Power Supply System

Pin name	Pin No.	I/O	Pull-up	Function
VDD	34,54, 78,102, 125,151, 174	–	–	Power supply (+) for the internal logic
VSS	12,29, 43,58, 70,81, 98,108, 113,131, 148,157, 168	–	–	Power supply (-); GND
VDDE	3,22,48, 73,95, 106,128, 142,160	–	–	Power supply (+) for the I/O block
AVDDE	65	–	–	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
A0 #BSL	163	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]	162,161, 159,158, 156–152, 150,149, 147,146, 144,143, 141,140	**O**	∗1	Address bus (A1–A17)
A18 P47	139	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	138	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	137	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	136	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	135	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	133	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"
A24 P41	132	I/**O**	∗1	A24: Address bus (A24) when EFP41[1:0](D[3:2]/0x300048) = "00" (default) P41: I/O port when EFP41[1:0](D[3:2]/0x300048) = "01"
A25 P40	130	I/**O**	∗1	A25: Address bus (A25) when EFP40[1:0](D[1:0]/0x300048) = "00" (default) P40: I/O port when EFP40[1:0](D[1:0]/0x300048) = "01"
#LCAS PA0 #SDCAS	124	I/**O** (H)	∗1	#LCAS: DRAM column address strobe (low byte) signal when FPA0[1:0](D[1:0]/0x300F60) = "00" (default) PA0: I/O port when FPA0[1:0](D[1:0]/0x300F60) = "01" #SDCAS: SDRAM column address strobe signal when FPA0[1:0](D[1:0]/0x300F60) = "10"
#HCAS PA1 #SDRAS	126	I/**O** (H)	∗1	#HCAS: DRAM column address strobe (high byte) signal when FPA1[1:0](D[3:2]/0x300F60) = "00" (default) PA1: I/O port when FPA1[1:0](D[3:2]/0x300F60) = "01" #SDRAS: SDRAM row address strobe signal when FPA1[1:0](D[3:2]/0x300F60) = "10"
D[15:0]	11–4, 2,1,175, 173–169	**I**/O	∗2	Data bus (D0–D15)

Pin name	Pin No.	I/O	Pull-up	Function
#CE10EX #CE9&10EX	119	**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	118	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"
#CE8 #RAS1 #CE14 #RAS3 P54	117	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 #SDCE	116	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" #SDCE: SDRAM chip enable when EFP53[1:0](D[7:6]/0x30004A) = "10"
#CE6 #CE7&8 P52	115	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	114	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	112	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	167	**O**	*1	Read signal
#WRL #WR #WE	166	**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	164	**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 SDCLK	129	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" SDCLK: SDRAM clock output when EFP60[1:0](D[1:0]/0x30004C) = "11"

Table I.3.3 List of Port Pins

Pin name	Pin No.	I/O	Pull-up	Function
K50 #DMAREQ0	69	**I**	∗1	K50: Input port when CFK50(D0/0x402C0) = "0" (default) #DMAREQ0: HSDMA Ch. 0 request input when CFK50(D0/0x402C0) = "1"
K51 #DMAREQ1	68	**I**	∗1	K51: Input port when CFK51(D1/0x402C0) = "0" (default) #DMAREQ1: HSDMA Ch. 1 request input when CFK51(D1/0x402C0) = "1"
K52 #ADTRG	67	**I**	∗1	K52: Input port when CFK52(D2/0x402C0) = "0" (default) #ADTRG: A/D converter trigger input when CFK52(D2/0x402C0) = "1"
K53 #DMAREQ2	66	**I**	∗1	K53: Input port when CFK53(D3/0x402C0) = "0" (default) #DMAREQ2: HSDMA Ch. 2 request input when CFK53(D3/0x402C0) = "1"
K60 AD0	59	**I**	∗1	K60: Input port when CFK60(D0/0x402C3) = "0" (default) AD0: A/D converter Ch. 0 input when CFK60(D0/0x402C3) = "1"
K61 AD1	60	**I**	∗1	K61: Input port when CFK61(D1/0x402C3) = "0" (default) AD1: A/D converter Ch. 1 input when CFK61(D1/0x402C3) = "1"
K62 AD2	61	**I**	∗1	K62: Input port when CFK62(D2/0x402C3) = "0" (default) AD2: A/D converter Ch. 2 input when CFK62(D2/0x402C3) = "1"
K63 AD3	62	**I**	∗1	K63: Input port when CFK63(D3/0x402C3) = "0" (default) AD3: A/D converter Ch. 3 input when CFK63(D3/0x402C3) = "1"
K64 AD4	63	**I**	∗1	K64: Input port when CFK64(D4/0x402C3) = "0" (default) AD4: A/D converter Ch. 4 input when CFK64(D4/0x402C3) = "1"
P00 SIN0 FSIN0	90	**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"
P01 SOUT0 FSOUT0	89	**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"
P02 #SCLK0 #FSCLK0	87	**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"
P03 #SRDY0 #FSRDY0	86	**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"
P04 SIN1 #DMAACK2	85	**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"
P05 SOUT1 #DMAEND2	84	**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"
P06 #SCLK1 #DMAACK3	83	**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"
P07 #SRDY1 #DMAEND3	82	**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"
P15 EXCL4 #DMAEND0 #SCLK3	45	**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"

Pin name	Pin No.	I/O	Pull-up	Function	
P16 EXCL5 #DMAEND1 SOUT3	44	I/O	*1	P16: EXCL5: #DMAEND1: SOUT3:	I/O port when CFP16(D6/0x402D4) = "0" (default) 16-bit timer 5 event counter input when CFP16(D6/0x402D4) = "1" and IOC16(D6/0x402D6) = "0" HSDMA Ch. 1 end-of-transfer signal output when CFP16(D6/0x402D4) = "1" and IOC16(D6/0x402D6) = "1" Serial I/F Ch. 3 data output when SSOUT3(D1/0x402D7) = "1" and CFP16(D6/0x402D4) = "0"
P20 #DRD SDCKE	127	I/O	*1	P20: #DRD: SDCKE:	I/O port when CFP20(D0/0x402D8) = "0" and EFP20[1:0](D[1:0]/0x300044) = "00" (default) DRAM read signal output for successive RAS mode when CFP20(D0/0x402D8) = "1" and EFP20[1:0](D[1:0]/0x300044) = "00" SDRAM clock enable output when EFP20[1:0](D[1:0]/0x300044) = "01"
P21 #DWE #GAAS #SDWE	123	I/O	*1	P21: #DWE: #GAAS: #SDWE:	I/O port when CFP21(D1/0x402D8) = "0", CFEX2(D2/0x402DF) = "0" and EFP21[1:0](D[3:2]/0x300044) = "00" (default) DRAM write signal output for successive RAS mode when CFP21(D1/0x402D8) = "1", CFEX2(D2/0x402DF) = "0" and EFP21[1:0](D[3:2]/0x300044) = "00" Area address strobe output for GA when CFEX2(D2/0x402DF) = "1" and EFP21[1:0](D[3:2]/0x300044) = "00" SDRAM write signal output when EFP21[1:0](D[3:2]/0x300044) = "01"
P22 TM0	23	I/O	*1	P22: TM0:	I/O port when CFP22(D2/0x402D8) = "0" (default) 16-bit timer 0 output when CFP22(D2/0x402D8) = "1"
P23 TM1	24	I/O	*1	P23: TM1:	I/O port when CFP23(D3/0x402D8) = "0" (default) 16-bit timer 1 output when CFP23(D3/0x402D8) = "1"
P24 TM2 #SRDY2	25	I/O	*1	P24: TM2: #SRDY2:	I/O port when CFP24(D4/0x402D8) = "0" (default) 16-bit timer 2 output when CFP24(D4/0x402D8) = "1" Serial I/F Ch. 2 ready signal input/output when SSRDY2(D3/0x402DB) = "1" and CFP24(D4/0x402D8) = "0"
P25 TM3 #SCLK2	26	I/O	*1	P25: TM3: #SCLK2:	I/O port when CFP25(D5/0x402D8) = "0" (default) 16-bit timer 3 output when CFP25(D5/0x402D8) = "1" Serial I/F Ch. 2 clock input/output when SSCLK2(D2/0x402DB) = "1" and CFP25(D5/0x402D8) = "0"
P26 *3 TM4 SOUT2	27	I/O	*1	P26: TM4: SOUT2:	I/O port when CFP26(D6/0x402D8) = "0" (default) 16-bit timer 4 output when CFP26(D6/0x402D8) = "1" Serial I/F Ch. 2 data output when SSOUT2(D1/0x402DB) = "1" and CFP26(D6/0x402D8) = "0"
P27 *3 TM5 SIN2	28	I/O	*1	P27: TM5: SIN2:	I/O port when CFP27(D7/0x402D8) = "0" (default) 16-bit timer 5 output when CFP27(D7/0x402D8) = "1" Serial I/F Ch. 2 data input when SSIN2(D0/0x402DB) = "1" and CFP27(D7/0x402D8) = "0"
P30 *4 #WAIT #CE4&5 PA2	13	I/O	*1	P30: #WAIT: #CE4&5: PA2:	I/O port when CFP30(D0/0x402DC) = "0" and FPA2[1:0](D[5:4]/0x300F60) = "00" (default) Wait cycle request input when CFP30(D0/0x402DC) = "1" and FPA2[1:0](D[5:4]/0x300F60) = "00" Areas 4&5 chip enable when CFP30(D0/0x402DC) = "1", IOC30(D0/0x402DE) = "1" and FPA2[1:0](D[5:4]/0x300F60) = "00" I/O port when FPA2[1:0](D[5:4]/0x300F60) = "01"
P31 #BUSGET #GARD SDO	96	I/O	*1	P31: #BUSGET: #GARD: SDO:	I/O port when CFP31(D1/0x402DC) = "0", CFEX3(D3/0x402DF) = "0" and EFP31[1:0](D[3:2]/0x300046) = "00" (default) Bus status monitor signal output for bus release request when CFP31(D1/0x402DC)="1", CFEX3(D3/0x402DF) = "0" and EFP31[1:0](D[3:2]/0x300046) = "00" Area read signal output for GA when CFEX3(D3/0x402DF) = "1" and EFP31[1:0](D[3:2]/0x300046) = "00" SPI data output when EFP31[1:0](D[3:2]/0x300046) = "01"
P32 #DMAACK0 #SRDY3 SPICLK	97	I/O	*1	P32: #DMAACK0: #SRDY3: SPICLK:	I/O port when CFP32(D2/0x402DC) = "0" and EFP32[1:0](D[5:4]/0x300046) = "00" (default) HSDMA Ch. 0 acknowledge output when CFP32(D2/0x402DC) = "1" and EFP32[1:0](D[5:4]/0x300046) = "00" Serial I/F Ch. 3 ready signal input/output when SSRDY3(D3/0x402D7) = "1", CFP32(D2/0x402DC) = "0" and EFP32[1:0](D[5:4]/0x300046) = "00" SPI clock input/output when EFP32[1:0](D[5:4]/0x300046) = "01"

Pin name	Pin No.	I/O	Pull-up	Function	
P33 #DMAACK1 SIN3 SDI	99	I/O	*1	P33: #DMAACK1: SIN3: SDI:	I/O port when CFP33(D3/0x402DC) = "0" and EFP33[1:0](D[7:6]/0x300046) = "00" (default) HSDMA Ch. 1 acknowledge output when CFP33(D3/0x402DC) = "1" and EFP33[1:0](D[7:6]/0x300046) = "00" Serial I/F Ch. 3 data input when SSIN3(D0/0x402D7) = "1", CFP33(D3/0x402DC) = "0" and EFP33[1:0](D[7:6]/0x300046) = "00" SPI data input when EFP33[1:0](D[7:6]/0x300046) = "01"
P34 #BUSREQ #CE6 #SMWE	100	I/O	*1	P34: #BUSREQ: #CE6: #SMWE:	I/O port when CFP34(D4/0x402DC) = "0" and EFP34[1:0](D[1:0]/0x300047) = "00" (default) Bus release request input when CFP34(D4/0x402DC)="1" and EFP34[1:0](D[1:0]/0x300047) = "00" Area 6 chip enable when CFP34(D4/0x402DC)="1", IOC34(D4/0x402DE) = "1" and EFP34[1:0](D[1:0]/0x300047) = "00" NAND flash write signal when EFP34[1:0](D[1:0]/0x300047) = "01"
P35 #BUSACK #SMRE	101	I/O	*1	P35: #BUSACK: #SMRE:	I/O port when CFP35(D5/0x402DC) = "0" and EFP35[1:0](D[3:2]/0x300047) = "00" (default) Bus acknowledge output when CFP35(D5/0x402DC) = "1" and EFP35[1:0](D[3:2]/0x300047) = "00" NAND flash read signal when EFP35[1:0](D[3:2]/0x300047) = "01"
P61 SDA10	120	I/O	*1	P61: SDA10:	I/O port when EFP61[1:0](D[3:2]/0x30004C) = "00" (default) SDRAM address A10 when EFP61[1:0](D[3:2]/0x30004C) = "01"
P62 LDQM	121	I/O	*1	P62: LDQM	I/O port when EFP62[1:0](D[5:4]/0x30004C) = "00" (default) SDRAM lower byte output mask when EFP62[1:0](D[5:4]/0x30004C) = "01"
P63 UDQM	122	I/O	*1	P63: UDQM	I/O port when EFP63[1:0](D[7:6]/0x30004C) = "00" (default) SDRAM upper byte output mask when EFP63[1:0](D[7:6]/0x30004C) = "01"
PB0 FPDAT0	38	I/O	*1	PB0: FPDAT0:	I/O port when FPB0[1:0](D[1:0]/0x300F62) = "00" (default) LCD display data output when FPB0[1:0](D[1:0]/0x300F62) = "01"
PB1 FPDAT1	37	I/O	*1	PB1: FPDAT1:	I/O port when FPB1[1:0](D[3:2]/0x300F62) = "00" (default) LCD display data output when FPB1[1:0](D[3:2]/0x300F62) = "01"
PB2 FPDAT2	36	I/O	*1	PB2: FPDAT2:	I/O port when FPB2[1:0](D[5:4]/0x300F62) = "00" (default) LCD display data output when FPB2[1:0](D[5:4]/0x300F62) = "01"
PB3 FPDAT3	35	I/O	*1	PB3: FPDAT3:	I/O port when FPB3[1:0](D[7:6]/0x300F62) = "00" (default) LCD display data output when FPB3[1:0](D[7:6]/0x300F62) = "01"
PB4 FPDAT4	33	I/O	*1	PB4: FPDAT4:	I/O port when FPB4[1:0](D[1:0]/0x300F63) = "00" (default) LCD display data output when FPB4[1:0](D[1:0]/0x300F63) = "01"
PB5 FPDAT5	32	I/O	*1	PB5: FPDAT5:	I/O port when FPB5[1:0](D[3:2]/0x300F63) = "00" (default) LCD display data output when FPB5[1:0](D[3:2]/0x300F63) = "01"
PB6 FPDAT6	31	I/O	*1	PB6: FPDAT6:	I/O port when FPB6[1:0](D[5:4]/0x300F63) = "00" (default) LCD display data output when FPB6[1:0](D[5:4]/0x300F63) = "01"
PB7 FPDAT7	30	I/O	*1	PB7: FPDAT7:	I/O port when FPB7[1:0](D[7:6]/0x300F63) = "00" (default) LCD display data output when FPB7[1:0](D[7:6]/0x300F63) = "01"
PC0 FPFRAME	42	I/O	*1	PC0: FPFRAME:	I/O port when FPC0[1:0](D[1:0]/0x300F64) = "00" (default) LCD vertical scan start pulse output when FPC0[1:0](D[1:0]/0x300F64) = "01"
PC1 FPLINE	41	I/O	*1	PC1: FPLINE:	I/O port when FPC1[1:0](D[3:2]/0x300F64) = "00" (default) LCD display data latch clock output when FPC1[1:0](D[3:2]/0x300F64) = "01"
PC2 FPSHIFT	40	I/O	*1	PC2: FPSHIFT:	I/O port when FPC2[1:0](D[5:4]/0x300F64) = "00" (default) LCD display data shift clock output when FPC2[1:0](D[5:4]/0x300F64) = "01"
PC3 DRDY	39	I/O	*1	PC3: DRDY:	I/O port when FPC3[1:0](D[7:6]/0x300F64) = "00" (default) LCD backplane bias signal output when FPC3[1:0](D[7:6]/0x300F64) = "01"
PD0 #SQRD	14	I/O	*1	PD0: #SQRD:	I/O port when FPD0[1:0](D[1:0]/0x300F66) = "00" (default) Sequential ROM read signal output when FPD0[1:0](D[1:0]/0x300F66) = "01"
PD1 SQLALE	15	I/O	*1	PD1: SQLALE:	I/O port when FPD1[1:0](D[3:2]/0x300F66) = "00" (default) Sequential ROM lower byte address latch enable output when FPD1[1:0](D[3:2]/0x300F66) = "01"
PD2 SQUALE	16	I/O	*1	PD2: SQUALE:	I/O port when FPD2[1:0](D[5:4]/0x300F66) = "00" (default) Sequential ROM upper byte address latch enable output when FPD2[1:0](D[5:4]/0x300F66) = "01"
PD3	17	I/O	*1	I/O port	
PD4	18	I/O	*1	I/O port	
PD5	19	I/O	*1	I/O port	
PD6	20	I/O	*1	I/O port	
PD7	21	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
DST0 P10 EXCL0 T8UF0	53	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"
DST1 P11 EXCL1 T8UF1	52	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"
DST2 P12 EXCL2 T8UF2	51	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"
DPCO P13 EXCL3 T8UF3	50	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"
DCLK P14 FOSC1	49	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"
DSIO	47	**I**/O	Pull-up	Serial I/O pin for debugging (with a 50kΩ pull-up resistor) This pin is used to communicate with the debugging tool S5U1C33000H/S5U1C33001H.

Table I.3.5 List of USB Pins

Pin name	Pin No.	I/O	Pull-up	Function
USBDP	91	**I**/O	–	USB D+ pin
USBDM	92	**I**/O	–	USB D- pin
USBVBUS	94	**I**	–	USB VBUS pin. Allows input of 5 V.

Table I.3.6 List of Other Pins

Pin name	Pin No.	I/O	Pull-up	Function
OSC1	71	**I**	–	Low-speed (OSC1) oscillation input (32 kHz crystal oscillator or external clock input)
OSC2	72	**O**	–	Low-speed (OSC1) oscillation output
OSC3	79	**I**	–	High-speed (OSC3) oscillation input for CPU (crystal (ceramic) oscillator or external clock input)
OSC4	80	**O**	–	High-speed (OSC3) oscillation output for CPU
PLLS0	109	**I**	∗6	PLL set-up pins 0: PLL not used, 1: Frequency-doubling mode
PLLC	107	–	–	Capacitor connecting pin for PLL
EA10MD1 **EA10MD0**	105 104	**I** **I**	Pull-up Pull-up	Area 10 boot mode selection pins (with a 50 kΩ pull-up resistor) EA10MD1 / EA10MD0 / Mode 1 / 1 / External ROM mode 1 / 0 / Internal ROM mode Connect the both pins to V_{DDE} during normal operation.
#X2SPD	103	**I**	∗7	Clock doubling mode set-up pin (with a 50 kΩ pull-up resistor) 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).
#NMI	55	**I**	Pull-up	NMI request input pin (with a 50 kΩ pull-up resistor)
#RESET	56	**I**	Pull-up	Initial reset input pin (with a 50 kΩ pull-up resistor)
BOOT	111	**I**	∗1	NAND flash booting select pin (with a 50 kΩ pull-up resistor)
TST	110	**I**	Pull-down	Test input pin (with a 50 kΩ pull-down resistor). Connect to V_{SS} during normal operation.
TEST1	76	**I**	Pull-down	Test input pin (with a 50 kΩ pull-down resistor). Connect to V_{SS} during normal operation.
TEST0	64	**I**	Pull-down	Test input pin (with a 120 kΩ pull-down resistor). Connect to V_{SS} during normal operation.
BURNIN	74	**I**	–	Test input pin. Connect to V_{SS} during normal operation.
SCANEN	75	**I**	Pull-down	Test input pin (with a 50 kΩ pull-down resistor). Connect to V_{SS} 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.
∗3: The P26 and P27 ports are implemented using an 8 mA I/O cell. This makes it possible to directly drive an LED without an external driver.
∗4: When the SDRAM controller or Bus Arbiter is used, the P30 I/O port function is disabled because SWAITE (D0/0x4812E) must be set (enabled). PA2 can be used as a general-purpose I/O port in this case. When the SDRAM controller and Bus Arbiter is not used, the P30 I/O port function can be used because SWAITE (D0/0x4812E) can be cleared (disabled).
∗5: The K54 and K65–K67 ports are not available in the S1C33L05. In the S1C33L05, HSDMA Ch. 3 is reserved for the USB function controller and the A/D inputs are limited to 5 channels.
∗6: A pull-down resistor is built in and is enabled only in vender test mode.
∗7: A pull-up resistor is built in and is enabled only in vender test mode.

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.

I-4 Power Supply

This section explains the operating voltage of the S1C33L05.

Power Supply Pins

The S1C33L05 has the power supply pins shown in Table I.4.1.

Table I.4.1 Power Supply Pins

Pin name	Pin No.	I/O	Function
VDD	34,54,78,102,125,151,174	–	Power supply (+) for the internal logic
VSS	12,29,43,58,70,81,98,108, 113,131,148,157,168	–	Power supply (-); GND
VDDE	3,22,48,73,95,106,128,142,160	–	Power supply (+) for the I/O block
AVDDE	65	–	Analog system power supply (+)

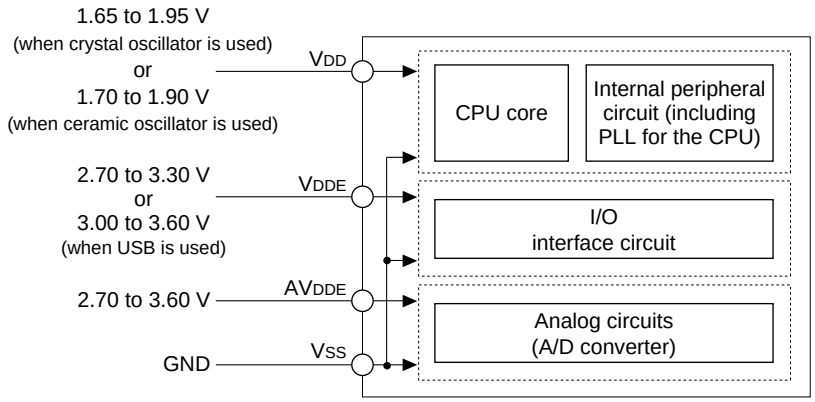


Figure I.4.1 Power Supply System

Operating Voltage (VDD, Vss)

The core CPU and internal peripheral circuits operate with a voltage supplied between the VDD and VSS pins. The following operating voltage can be used:

VDD = 1.65 V to 1.95 V (1.8 V ± 0.15 V, VSS = GND) when crystal oscillator is used
1.70 V to 1.90 V (1.8 V ± 0.10 V, VSS = GND) when ceramic oscillator is used

Note: The S1C33L05 has 7 VDD pins and 13 VSS pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.

Power Supply for I/O Interface (VDDE)

The VDDE voltage is used for interfacing with external I/O signals. For the output interface of the S1C33L05, the VDDE voltage is used as high level and the VSS voltage as low level. The VSS pin is used for the ground common with VDD. The following voltage is enabled for VDDE:

VDDE = 3.00 V to 3.60 V (3.3 V ± 0.3 V, VSS = GND)

When the USB function controller is not used, the following voltage is enabled for VDDE:

VDDE = 3.00 V to 3.60 V (3.3 V ± 0.3 V, VSS = GND)
or 2.70 V to 3.30 V (3.0 V ± 0.3 V, VSS = GND)

Notes: • The S1C33L05 has 9 VDDE 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 VDD.

Power Supply for Analog Circuits (AV_{DDE})

The analog power supply pin (AV_{DDE}) is provided separately from the V_{DD} and V_{DDE} pins in order that the digital circuits do not affect the analog circuit (A/D converter). The AV_{DDE} pin is used to supply an analog power voltage and the V_{SS} pin is used as the analog ground.
The following voltage is enabled for AV_{DDE}:

AV_{DDE} = 2.70 V to 3.60 V (3.0/3.3 V ± 0.3 V, V_{SS} = GND)

Note: Be sure to supply V_{DDE} to the AV_{DDE} 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.

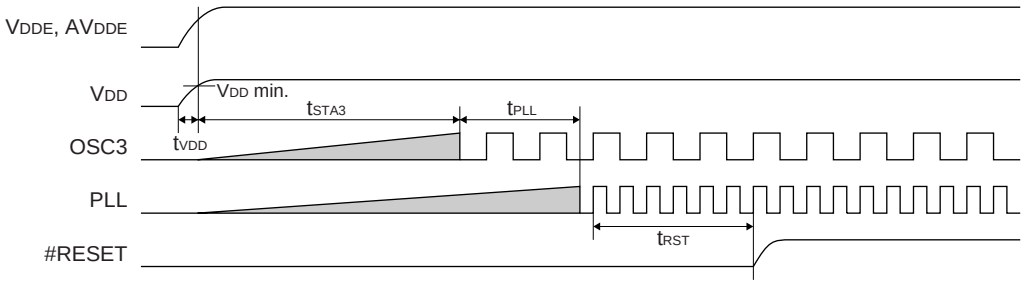


Figure I.4.2 Power-On Sequence

(1) tVDD: 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) 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.

Power-off sequence

Shut off the power supply in the following sequence (or simultaneously).
Power-off: Turn off the input signal → AVDDE and VDDE (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 PNPN 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.

THIS PAGE IS BLANK.

I-5 Internal Memory

This section explains the internal memory configuration.

Figures I.5.1 and I.5.2 show the S1C33L05 memory map.

Area	Address	Internal Areas	External Areas
Areas 18–12	0xFFFFFFF 0x1800000	(Reserved)	External Memory 8MB or more
Area 11	0x17FFFFF 0x1000000	USB register area	(Reserved for USB registers) ∗1
Areas 10–9	0x0FFFFFF 0x0800000	(Reserved)	External Memory 4MB × 2
Areas 8–7	0x07FFFFF 0x0400000	(Reserved)	External Memory 2MB × 2
Area 6	0x03FFFFF 0x0380000	Internal I/O area (16-bit) LCDC, SDRAMC, etc.	(Reserved for LCDC, SDRAMC, etc.)
	0x037FFFF 0x0300000	Internal I/O area (8-bit) FSIO, GPIO, etc.	(Reserved for FSIO, GPIO, etc.)
Area 5	0x02FFFFF 0x0200000	(Reserved)	External Memory 1MB
Area 4	0x01FFFFF 0x0100000	USB DMA area	(Reserved for USB DMA) ∗1
Area 3	0x00FFFFF 0x0080000	(Reserved)	
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 0x0004000	(Mirror of internal RAM)	
	0x0003FFF 0x0000000	Internal RAM (16KB)	

∗1: These areas can be used for external memory when the USB function controller is not used.

Figure I.5.1 Memory Map

Area	Address	
Area 11	0x17FFFFF 0x1100000	(Reserved)
	0x10FFFFF 0x1000000	USB control registers
	⋮	
Area 6 16-bit access area	0x03FFFFF 0x03CA000	(Reserved)
	0x03C9FFF 0x03C0000	Video RAM for LCDC
	0x03BFFFF 0x03A0300	(Reserved)
	0x03A02FF 0x03A0200	SDRAMC control registers
	0x03A01FF 0x03A0100	Sequential ROM I/F control registers
	0x03A00FF 0x03A0000	MMC (SPI mode) control registers
	0x039FFFF 0x0390100	(Reserved)
	0x03900FF 0x0390000	Bus arbiter control registers
	0x038FFFF 0x0380000	LCDC control registers
Area 6 8-bit access area	0x037FFFF 0x0301000	(Reserved)
	0x0300FFF 0x0300F40	Extended GPIO (PA–PD) control registers
	0x0300F3F 0x0300F20	Miscellaneous registers
	0x0300F1F 0x0300F00	Pull-up control registers
	0x0300EFF 0x0300300	(Reserved)
	0x03002FF 0x0300200	Serial interface with FIFO control registers
	0x03001FF 0x0300100	NAND flash interface control registers
	0x03000FF 0x0300060	(Reserved)
	0x030005F 0x0300040	P0–P6 port function extension registers
	0x030003F 0x0300020	P4–P6 I/O port control registers
	0x030001F 0x0300000	Chip ID registers
	⋮	
Area 4	0x01FFFFF 0x0100000	USB DMA area

Figure I.5.2 Detailed Internal Area Map

Area 2

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

Area 6

Area 6 is reserved for the extended peripheral circuits, such as LCDC, SDRAMC, Sequential ROM interface, SPI, bus arbiter, etc.
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 S1C33L05 extended peripheral circuits and LCD controller 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) = "002"
   This setting is required only for setting the SDRAMC block in x1 speed mode.
4. A6RH (D1) / Read cycle hold time control register (0x4813C) = "1"
   This extends the access time for reading Area 6 by 1 cycle. This setting is required only for using the USB function.
5. SWAITE (D0) / Bus control register (0x4812E) = "1"
   This enables wait control using the external or internal #WAIT signal.
   This setting is required to access the LCDC block.

Note: The bus speed is limited, up to 40 MHz. When running the CPU with a 40 MHz or higher operating clock, set the #X2SPD pin to "0", A6BS (D1/0x4813E) to "0", and A1X1MD (D3/0x4813A) to "1".

Areas 4 and 11

Areas 4 and 11 are reserved for the USB function controller.
Areas 4 and 11 are configured for external memory access at initial reset. When using the USB function controller, set up the access conditions for these areas as follows:

1. A12IO (DC) / Access control register (0x48132) = "1"
   A5IO (D8) / Access control register (0x48132) = "1"
   Areas 12–11 and 5–4 are configured for internal devices.
2. A12EC (D4) / Access control register (0x48132) = "0"
   A5EC (D0) / Access control register (0x48132) = "0"
   Areas 12–11 and 5–4 are configured for little endian access.
3. A12SZ (D6) / Areas 12–11 setup register (0x48124) = "1"
   The device size of Areas 12–11 is set to 8 bits.
4. A12WT[2:0] (D[2:0]) / Areas 12–11 setup register (0x48124) = "010" or "001"
   The number of wait cycles for accessing Areas 12–11 (USB control registers) is set to 2 (when the CPU is running with a 40-MHz or higher clock) or 1 (when the CPU is running with a less than 40-MHz clock and in x1 speed mode).
5. A5SZ (D6) / Areas 6–4 setup register (0x4812A) = "1"
   The device size of Areas 5–4 is set to 8 bits.
6. A5WT[2:0] (D[2:0]) / Areas 6–4 setup register (0x4812A) = "001"
   The number of wait cycles for accessing Areas 5–4 (USB DMA area) is set to 1.
7. SWAITE (D0) / Bus control register (0x4812E) = "1"
   The wait control using the #WAIT signal is enabled.

8. USBWT[2:0] (D[2:0]) / Macro control register (0x300F20)
   The USB function controller in Area 11 needs a different wait control from other internal blocks and the USB bus wait control bits (USBWT[2:0]) are provided in the macro control register. Wait states selected by USBWT[2:0] will be inserted into the USB register access cycle along with the wait states selected by A12WT[2:0].
   The USBWT[2:0] should be set as follows according to the CPU operating clock frequency:
   USBWT[2:0] = "111" (CPU operating clock frequency = 40 MHz or more)
   = "101" (CPU operating clock frequency = 24 MHz)
   = "010" (CPU operating clock frequency = 12 MHz)
   = "000" (CPU operating clock frequency = 6 MHz or less)

Note: The bus speed is limited, up to 40 MHz. When running the CPU with a 40 MHz or higher operating clock, set the #X2SPD pin to "0", A5BS (D0/0x4813E) to "0", A12BS (D4/0x4813E) to "0", and A1X1MD (D3/0x4813A) to "1".

When the USB function is not used, you can use Areas 11 and 4 for external memory devices.
Furthermore, we recommend that CEFUNC (D[A:9]/0x48130) is set to "10" or "11", since this setting allows you to use all #CE pins to configure large memory space. For details of the CEFUNC settings, refer to "External Memory Map and Chip Enable" in Section II-4, "BCU (Bus Control Unit)".

ROM and Boot Address

The S1C33L05 has a built-in ROM with 128 instructions, which is used to boot from NAND flash (NAND flash boot mode). For successful booting, an external ROM/NOR-flash should be used in Area 10 or an external NAND flash should be used in Area 5.

Refer to Section II-4, "BCU (Bus Control Unit)", for setting up Area 10. And refer to Section III-14, "NAND Flash Interface", for setting up NAND flash booting.

RAM

The S1C33L05 has a built-in 16KB RAM. The RAM is allocated to Area 0, address 0x0000000 to address 0x0003FFF.
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 S1C33L05 external bus allows output of the low-order 26 bits (max.).

I-6 Electrical Characteristics

Absolute Maximum Rating

(V_{SS}=0V)

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

Recommended Operating Conditions

(V_{SS}=0V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Internal logic power voltage	V_{DD}	when crystal oscillator is used.	1.65	1.80	1.95	V	
		when ceramic oscillator is used.	1.70	1.80	1.90	V	
I/O power voltage	V_{DDE}	when USB controller is used.	3.00	3.30	3.60	V	
		when USB controller is not used.	2.70	–	3.60	V	
Analog power voltage	AV_{DDE}		2.70	–	3.60	V	
Input voltage	HV_I		V_{SS}	–	V_{DDE}	V	
	LV_I		V_{SS}	–	V_{DD}	V	
Analog input voltage	AV_{IN}		V_{SS}	–	AV_{DDE}	V	
CPU operating clock frequency	f_{CPU}		–	–	48	MHz	
Bus operating clock frequency	f_{BUS}		–	–	40	MHz	
OSC3 oscillation frequency	f_{OSC3}	when crystal oscillator is used.	20	–	48	MHz	
		when ceramic oscillator is used.	–	48	–	MHz	
OSC3 external input clock frequency	f_{ECLK3}		2	–	48	MHz	
OSC1 oscillation frequency	f_{OSC1}		–	32.768	–	kHz	
Operating temperature	Ta	when crystal oscillator is used.	-40	25	85	°C	
		when ceramic oscillator is used.	0	25	70	°C	
Input rise time (normal input)	tri		–	–	50	ns	
Input fall time (normal input)	tfi		–	–	50	ns	
Input rise time (Schmitt input)	tri		–	–	5	ms	
Input fall time (Schmitt input)	tfi		–	–	5	ms	

DC Characteristics

(Unless otherwise specified: V_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, Ta=-40°C to +85°C)

Item	Symbol	Condition		Min.	Typ.	Max.	Unit	*
Input leakage current	I_{LI}			-5	–	5	µA	
Off-state leakage current	I_{OZ}			-5	–	5	µA	
High-level output voltage	V_{OH}	I_{OH}=-1.7mA (2mA Type), I_{OH}=-3.5mA (4mA Type), I_{OH}=-7.1mA (8mA Type), V_{DD}=Min.		V_{DD} -0.4	–	–	V	
Low-level output voltage	V_{OL}	I_{OL}=1.7mA (2mA Type), I_{OL}=3.5mA (4mA Type), I_{OL}=7.1mA (8mA Type), V_{DD}=Min.		–	–	0.4	V	
High-level input voltage	V_{IH}	LVTTL level, V_{DDE}=Max.		2.0	–	–	V	
Low-level input voltage	V_{IL}	LVTTL level, V_{DDE}=Min.		–	–	0.7	V	
Positive trigger input voltage	V_{T+}	LVCMOS Schmitt		1.2	–	2.7	V	
Negative trigger input voltage	V_{T-}	LVCMOS Schmitt		0.5	–	1.8	V	
Hysteresis voltage	V_H	LVCMOS Schmitt		0.2	–	–	V	
Pull-up resistor	R_{PU}	V_I=0V	100kΩ Type	50	100	288	kΩ	
			50kΩ Type	25	50	144	kΩ	
Pull-down resistor	R_{PD}	V_I=V_{DDE}	120kΩ Type	60	120	346	kΩ	
			50kΩ Type	25	50	144	kΩ	
High-level latching current	I_{BHH}	Pins with bus-hold latch, V_I=1.9V, V_{DDE}=Min.		–	–	-20	µA	
Low-level latching current	I_{BHL}	Pins with bus-hold latch, V_I=0.8V, V_{DDE}=Min.		–	–	17	µA	
High-level reversal current	I_{BHHO}	Pins with bus-hold latch, V_I=0.8V, V_{DDE}=Max.		-350	–	–	µA	
Low-level reversal current	I_{BHLO}	Pins with bus-hold latch, V_I=1.9V, V_{DDE}=Max.		300	–	–	µA	
Input pin capacitance	C_I	f=1MHz, V_{DDE}=0V		–	–	8	pF	
Output pin capacitance	C_O	f=1MHz, V_{DDE}=0V		–	–	8	pF	
I/O pin capacitance	C_{IO}	f=1MHz, V_{DDE}=0V		–	–	8	pF	

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

USB DC and AC Characteristics

Input levels

(Unless otherwise specified: USBVDD=3.0V to 3.6V, VDD=1.65V to 1.95V, Ta=-40°C to +85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
VBUS input	VBUS		4.40	–	5.25	V	1
High (driven)	VIH		2.0	–	–	V	2
High (floating)	VIHZ		2.7	–	3.6	V	2
Low	VIL		–	–	0.8	V	2
Differential input sensitivity	VDI	|DP - DM|	0.2	–	–	V	
Differential common mode range	VCM	Include VDI range	0.8	–	2.5	V	

∗ note 1) Refer to Section 7.2.1 in the USB2.0 Specification for the conditions.
2) Refer to Section 7.1.4 in the USB2.0 Specification for the conditions.

Output levels

(Unless otherwise specified: USBVDD=3.0V to 3.6V, VDD=1.65V to 1.95V, Ta=-40°C to +85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Low	VOL		0.0	–	0.3	V	3
High (driven)	VOH		2.8	–	3.6	V	3
Output signal crossover voltage	VCRS		1.3	–	2.0	V	4

∗ note 3) Refer to Section 7.1.1 in the USB2.0 Specification for the conditions.
4) Refer to Figures 7-8 and 7-9 in the USB2.0 Specification for the conditions.

Terminations

(Unless otherwise specified: USBVDD=3.0V to 3.6V, VDD=1.65V to 1.95V, Ta=-40°C to +85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Bus pull-up resistor on upstream facing port (idle Bus)	RPUI		0.9	–	1.575	kΩ	5
Bus pull-up resistor on upstream facing port (receiving)	VPUA		1.425	–	3.090	kΩ	5

∗ note 5) Refer to ECN in the USB2.0 Specification for the conditions.

Driver characteristics

(Unless otherwise specified: USBVDD=3.0V to 3.6V, VDD=1.65V to 1.95V, Ta=-40°C to +85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Rise time	TFR		4	–	20	ns	4
Fall time	TFF		4	–	20	ns	4
Differential rise and fall time matching	TFRFM	TFR/TFF	90	–	111.11	%	
Driver output resistance	ZDRV		28	–	44	Ω	
VBUS input impedance	ZVBUS	R1 + R2	125	–		kΩ	
VBUS resistor ratio		R1 : R2	1 : 2 (nominal)				

∗ note 4) Refer to Figures 7-8 and 7-9 in the USB2.0 Specification for the conditions.

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, Ta=-40°C to +85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Current consumption during CPU running	I_{DD1}	20MHz	–	10	–	mA	1
		25MHz	–	12	–		
		33MHz	–	16	–		
		48MHz	–	23	–		
Current consumption in HALT mode	I_{DD2}	20MHz	–	5	–	mA	2
		25MHz	–	6	–		
		33MHz	–	7	–		
		48MHz	–	10	–		
Current consumption in HALT2 mode	I_{DD3}	20MHz	–	1.5	–	mA	3
		25MHz	–	2	–		
		33MHz	–	2.5	–		
		48MHz	–	4	–		

Current consumption measurement condition: V_{IH}=V_{DDE}, V_{IL}=0V, output pins are open, V_{DD} power current only
Typ. value measurement condition: V_{DDE}=AV_{DDE}=3.3V, V_{DD}=1.8V, Ta=25°C Typ. sample

∗ 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}=1.8V, V_{SS}=0V, Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
V_{DD} current consumption in SLEEP mode	I_{DDSL}	Ta=25°C	–	1	–	µA	
		Ta=85°C	–	11	–		
V_{DDE}+AV_{DDE} current consumption in SLEEP mode	I_{DDSE}	Ta=25°C	–	3	–	µA	
		Ta=85°C	–	5	–		

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, Ta=25°C Typ. sample

Peripheral circuit operating currents

(Unless otherwise specified: V_{DDE}=AV_{DDE}=3.3V, V_{DD}=1.8V, V_{SS}=0V, Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	∗
Clock timer operating current	I_{CT}	OSC1 oscillation: 32kHz	–	2.5	–	µA	4
A/D converter operating current	I_{AD}	When A/D converter is enabled	–	260	–	µA	5
OSC3 operating current	I_{OSC3}	OSC3 oscillation: 48MHz	–	0.6	–	mA	4
OSC3 PLL operating current	I_{PLL}	PLL output clock: 48MHz	–	1	–	mA	4
LCDC display current	I_{LCDCD}	LCDC clock: 8MHz, 16 bpp, IVRAM mode	–	1	–	mA	4
USB controller current	I_{USB}	Idle state with clock enabled	–	7.3	–	mA	4

∗ note 4) V_{DD} power current consumption
5) AV_{DDE} power current consumption

A/D Converter Characteristics

(Unless otherwise specified: V_{DDE}=AV_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=-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_{ZS}		-2	–	2	LSB	
Full scale error	E_{FS}		-2	–	2	LSB	
Integral linearity error	E_L		-3	–	3	LSB	
Differential linearity error	E_D		-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.

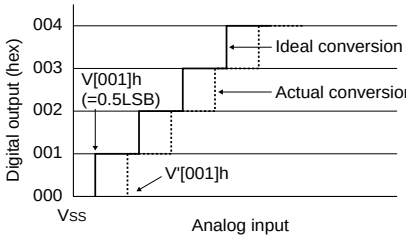
A/D conversion error

V[001]h = Ideal voltage at zero-scale point (=0.5LSB)
V'[001]h = Actual voltage at zero-scale point
V[3FF]h = Ideal voltage at full-scale point (=1022.5LSB)
V'[3FF]h = Actual voltage at full-scale point

$$1LSB = \frac{AV_{DDE} - V_{SS}}{2^{10} - 1}$$

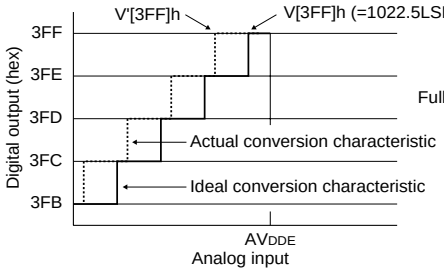
$$1LSB' = \frac{V'[3FF]h - V'[001]h}{2^{10} - 2}$$

■ **Zero scale error**



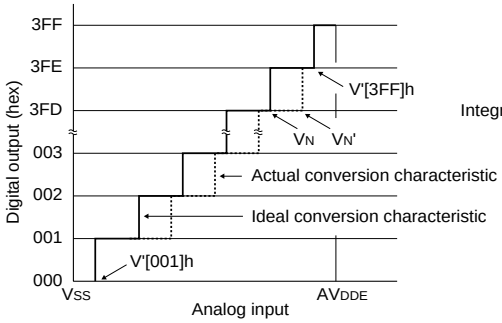
$$\text{Zero scale error } E_{ZS} = \frac{(V'[001]h - 0.5LSB') - (V[001]h - 0.5LSB)}{1LSB} \text{ [LSB]}$$

■ **Full scale error**



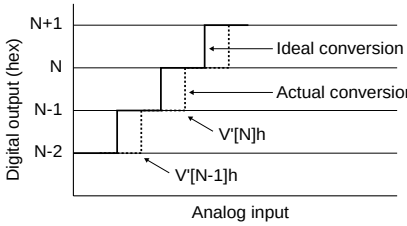
$$\text{Full scale error } E_{FS} = \frac{(V'[3FF]h + 0.5LSB') - (V[3FF]h + 0.5LSB)}{1LSB} \text{ [LSB]}$$

■ **Integral linearity error**



$$\text{Integral linearity error } E_L = \frac{V_N' - V_N}{1LSB'} \text{ [LSB]}$$

■ **Differential linearity error**



$$\text{Differential linearity error } E_D = \frac{V'[N]h - V'[N-1]h}{1LSB'} - 1 \text{ [LSB]}$$

Oscillation Characteristics

Oscillation characteristics change depending on conditions such as components used (oscillator, R_f, R_d, C_G, C_D) 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_f, R_d) and capacitor (C_G, C_D) values are finally decided.

OSC1 crystal oscillation

(Unless otherwise specified: V_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA1}				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_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA3}				25	ms	

PLL Characteristics

Setting the PLLS0 pin (recommended operating condition)

PLLS0	Mode	Fin (OSC3 clock)	Fout
1	Frequency doubling mode	10 to 24MHz	20 to 48MHz *1
0	PLL not used	–	–

*1 When PLLPDWN (D1/0x300F31) = "0" and PLLMODE (D0/0x300F31) = "1"
10 to 24MHz when PLLPDWN (D1/0x300F31) = "0" and PLLMODE (D0/0x300F31) = "0"

PLL characteristics

(Unless otherwise specified: V_{DD}=1.65V to 1.95V, V_{SS}=0V, R_1=4.7kΩ, C_1=100pF, C_2=3pF, Ta=-40°C to +85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Lockup time	t_{PLL}				1	ms	

Pin Characteristics

Pin No.	Pad No.	Signal name	I/O	I/O cell name	Input/output characteristic	Output characteristic	Pull-up/down (register value)	Power source	Remarks
1	1	**D9**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
2	2	**D8**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
3	3	**VDDE**	P	–	–	–	–	–	
4	4	**D7**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
5	5	**D6**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
6	6	**D5**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
7	7	**D4**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
8	8	**D3**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
9	9	**D2**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
10	10	**D1**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
11	11	**D0**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	VDDE	
12	12	**VSS**	P	–	–	–	–	–	
13	13	**P30**/#WAIT/#CE4&5/PA2	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
14	14	**PD0**/#SQRD	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
15	15	**PD1**/SQLALE	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
16	16	**PD2**/SQUALE	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
17	17	**PD3**	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
18	18	**PD4**	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
19	19	**PD5**	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
20	20	**PD6**	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
21	21	**PD7**	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
22	22	**VDDE**	P	–	–	–	–	–	
23	23	**P22**/TM0	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
24	24	**P23**/TM1	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
25	25	**P24**/TM2/#SRDY2	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
26	26	**P25**/TM3/#SCLK2	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
27	27	**P26**/TM4/SOUT2	I/O	HBBH3BP2TY	SCHMITT	8mA	Pull-up(100k)	VDDE	note 1
28	28	**P27**/TM5/SIN2	I/O	HBBH3BP2TY	SCHMITT	8mA	Pull-up(100k)	VDDE	note 1
29	29	**VSS**	P	–	–	–	–	–	
30	30	**PB7**/FPDAT7	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
31	31	**PB6**/FPDAT6	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
32	32	**PB5**/FPDAT5	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
33	33	**PB4**/FPDAT4	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
34	34	**VDD**	P	–	–	–	–	–	
35	35	**PB3**/FPDAT3	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
36	36	**PB2**/FPDAT2	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
37	37	**PB1**/FPDAT1	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
38	38	**PB0**/FPDAT0	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
39	39	**PC3**/DRDY	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
40	40	**PC2**/FPSHIFT	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
41	41	**PC1**/FPLINE	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
42	42	**PC0**/FPFRAME	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
43	43	**VSS**	P	–	–	–	–	–	
44	44	**P16**/EXCL5/#DMAEND1/SOUT3	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
45	45	**P15**/EXCL4/#DMAEND0/#SCLK3	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	VDDE	note 1
46	–	N.C.	–	–	–	–	–	–	
47	46	**DSIO**	I/O	HBBH2BP1TY	SCHMITT	4mA	Pull-up(50k)	VDDE	
48	47	**VDDE**	P	–	–	–	–	–	
49	48	**DCLK**/P14/FOSC1	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	VDDE	note 1
50	49	**DPCO**/P13/EXCL3/T8UF3	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	VDDE	note 1
51	50	**DST2**/P12/EXCL2/T8UF2	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	VDDE	note 1
52	51	**DST1**/P11/EXCL1/T8UF1	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	VDDE	note 1
53	52	**DST0**/P10/EXCL0/T8UF0	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	VDDE	note 1
54	53	**VDD**	P	–	–	–	–	–	
55	54	**#NMI**	I	HIBHP1TY	SCHMITT	–	Pull-up(50k)	VDDE	
56	55	**#RESET**	I	HIBHP1TY	SCHMITT	–	Pull-up(50k)	VDDE	
57	–	N.C.	–	–	–	–	–	–	
58	56	**VSS**	P	–	–	–	–	–	
59	57	**K60**/AD0	I	HIBASP2TY	Special use	–	Pull-up(100k)	AVDDE	note 1, 4
60	58	**K61**/AD1	I	HIBASP2TY	Special use	–	Pull-up(100k)	AVDDE	note 1, 4

Pin No.	Pad No.	Signal name	I/O	I/O cell name	Input/output characteristic	Output characteristic	Pull-up/down (register value)	Power source	Remarks
61	59	**K62**/AD2	I	HIBASP2TY	Special use	–	Pull-up(100k)	AV_{DDE}	note 1, 4
62	60	**K63**/AD3	I	HIBASP2TY	Special use	–	Pull-up(100k)	AV_{DDE}	note 1, 4
63	61	**K64**/AD4	I	HIBASP2TY	Special use	–	Pull-up(100k)	AV_{DDE}	note 1, 4
64	62	**TEST0**	I	LITST1Y	LVCMOS	–	Pull-down(120k)	V_{DDE}	
65	63	**AV_{DDE}**	P	–	–	–	–	–	
66	64	**K53**/#DMAREQ2	I	HIBHP2TY	SCHMITT	–	Pull-up(100k)	V_{DDE}	note 1
67	65	**K52**/#ADTRG	I	HIBHP2TY	SCHMITT	–	Pull-up(100k)	V_{DDE}	note 1
68	66	**K51**/#DMAREQ1	I	HIBHP2TY	SCHMITT	–	Pull-up(100k)	V_{DDE}	note 1
69	67	**K50**/#DMAREQ0	I	HIBHP2TY	SCHMITT	–	Pull-up(100k)	V_{DDE}	note 1
70	68	**V_{SS}**	P	–	–	–	–	–	
71	69	**OSC1**	I	LLINY	Transparent	–	–	V_{DD}	note 3
72	70	**OSC2**	O	LLOTY	Transparent	–	–	V_{DD}	note 3
73	71	**V_{DDE}**	P	–	–	–	–	–	
74	72	**BURNIN**	I	HIBHY	SCHMITT	–	–	V_{DDE}	
75	73	**SCANEN**	I/O	HBBH1BD1TY	SCHMITT	2mA	Pull-down(50k)	V_{DDE}	
76	74	**TEST1**	I	HIBHD1TY	SCHMITT	–	Pull-down(50k)	V_{DDE}	
77	–	N.C.	–	–	–	–	–	–	
78	75	**V_{DD}**	P	–	–	–	–	–	
79	76	**OSC3**	I	LLINY	Transparent	–	–	V_{DDE}	note 3
80	77	**OSC4**	O	LLOTY	Transparent	–	–	–	note 3
81	78	**V_{SS}**	P	–	–	–	–	–	
82	79	**P07**/#SRDY1/#DMAEND3	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	
83	80	**P06**/#SCLK1/#DMAACK3	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
84	81	**P05**/#SOUT1/#DMAEND2	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
85	82	**P04**/#SIN1/#DMAACK2	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
86	83	**P03**/#SRDY0/#FSRDY0	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
87	84	**P02**/#SCLK0/#FSCLK0	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
88	–	N.C.	–	–	–	–	–	–	
89	85	**P01**/SOUT0/FSOUT0	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
90	86	**P00**/SIN0/FSIN0	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
91	87	**USBDP**	I/O	EIFUFPCX711	Special use	–	–	V_{DDE}	
92	88	**USBDM**	I/O	EIFUFPCX711	Special use	–	–	V_{DDE}	
93	89	N.C.	–	–	–	–	–	–	
94	90	**USBVBUS**	I	EIFUFPCX711	Special use	–	–	V_{DDE}	5V input
95	91	**V_{DDE}**	P	–	–	–	–	–	
96	92	**P31**/#BUSGET/#GARD/SDO	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
97	93	**P32**/#DMAACK0/#SRDY3/SPICLK	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
98	94	**V_{SS}**	P	–	–	–	–	–	
99	95	**P33**/#DMAACK1/SIN3/SDI	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
100	96	**P34**/#BUSREQ/#CE6/#SMWE	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
101	97	**P35**/#BUSACK/#SMRE	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
102	98	**V_{DD}**	P	–	–	–	–	–	
103	99	**#X2SPD**	I	HIBHP1TY	SCHMITT	–	Pull-up(50k)	V_{DDE}	
104	100	**EA10MD0**	I	HIBHP1TY	SCHMITT	–	Pull-up(50k)	V_{DDE}	
105	101	**EA10MD1**	I	HIBHP1TY	SCHMITT	–	Pull-up(50k)	V_{DDE}	
106	102	**V_{DDE}**	P	–	–	–	–	–	
107	103	**PLLC**	I	LLINY	Transparent	–	–	V_{DDE}	note 3
108	104	**V_{SS}**	P	–	–	–	–	–	
109	105	**PLLS0**	I	HIBHD1TY	SCHMITT	–	Pull-down(50k)	V_{DDE}	
110	106	**TST**	I	HIBHD1TY	SCHMITT	–	Pull-down(50k)	V_{DDE}	
111	107	**BOOT**	I	HIBHP1TY	SCHMITT	–	Pull-up(50k)	V_{DDE}	
112	108	**#CE4**/#CE11/#CE11&12/P50	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
113	109	**V_{SS}**	P	–	–	–	–	–	
114	110	**#CE5**/#CE15/#CE15&16/P51	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
115	111	**#CE6**/#CE7&8/P52	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
116	112	#**CE7**/#RAS0/#CE13/#RAS2/P53/#SDCE	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
117	113	**#CE8**/#RAS1/#CE14/#RAS3/P54	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
118	114	**#CE9**/#CE17/#CE17&18/P55	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
119	115	**#CE10EX**/#CE9&10EX	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
120	116	**P61**/SDA10	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1

Pin No.	Pad No.	Signal name	I/O	I/O cell name	Input/output characteristic	Output characteristic	Pull-up/down (register value)	Power source	Remarks
121	117	**P62**/LDQM	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
122	118	**P63**/UDQM	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
123	119	**P21**/#DWE/#GAAS/#SDWE	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
124	120	**#LCAS**/PA0/#SDCAS	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
125	121	**V_{DD}**	P	–	–	–	–	–	
126	122	**#HCAS**/PA1/#SDRAS	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
127	123	**P20**/#DRD/SDCKE	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
128	124	**V_{DDE}**	P	–	–	–	Pull-up(100k)	V_{DDE}	note 1
129	125	**BCLK**/P60/FOSC1/SDCLK	I/O	HBBH1BP2TY	SCHMITT	2mA	Pull-up(100k)	V_{DDE}	note 1
130	126	**A25**/P40	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
131	127	**V_{SS}**	P	–	–	–	–	–	
132	128	**A24**/P41	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
133	129	**A23**/P42	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
134	–	N.C.	–	–	–	–	–	–	
135	130	**A22**/P43	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
136	131	**A21**/P44	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
137	132	**A20**/P45	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
138	133	**A19**/P46	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
139	134	**A18**/P47	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
140	135	**A17**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
141	136	**A16**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
142	137	**V_{DDE}**	P	–	–	–	–	–	
143	138	**A15**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
144	139	**A14**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
145	–	N.C.	–	–	–	–	–	–	
146	140	**A13**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
147	141	**A12**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
148	142	**V_{SS}**	P	–	–	–	–	–	
149	143	**A11**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
150	144	**A10**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
151	145	**V_{DD}**	P	–	–	–	–	–	
152	146	**A9**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
153	147	**A8**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
154	148	**A7**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
155	149	**A6**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
156	150	**A5**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
157	151	**V_{SS}**	P	–	–	–	–	–	
158	152	**A4**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
159	153	**A3**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
160	154	**V_{DDE}**	P	–	–	–	–	–	
161	155	**A2**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
162	156	**A1**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
163	157	**A0**/#BSL	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1, 2
164	158	**#WRH**/#BSH	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
165	–	N.C.	–	–	–	–	–	–	
166	159	**#WRL**/#WR/#WE	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
167	160	**#RD**	I/O	HBBH2BP2TY	SCHMITT	4mA	Pull-up(100k)	V_{DDE}	note 1
168	161	**V_{SS}**	P	–	–	–	–	–	
169	162	**D15**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	V_{DDE}	
170	163	**D14**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	V_{DDE}	
171	164	**D13**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	V_{DDE}	
172	165	**D12**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	V_{DDE}	
173	166	**D11**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	V_{DDE}	
174	167	**V_{DD}**	P	–	–	–	–	–	
175	168	**D10**	I/O	HBBT2BHTY	LVTTL	4mA	Bus hold latch	V_{DDE}	
176	–	N.C.	–	–	–	–	–	–	

note 1) The pull-up resistor can be enabled/disabled using the I/O register (pull-up is enabled by default).
note 2) This pin is set as an input pin during device testing. Normally it is an output pin.
note 3) The voltage applied to this pin must be $0V \leq V_{IN} \leq V_{DD}$.
note 4) The voltage applied to this pin must be $0V \leq V_{IN} \leq AV_{DDE}$.

AC Characteristics

Symbol Description

t_{CYC}: Bus-clock cycle time

- In x1 mode, t_{CYC} = 50 ns (20 MHz) when the CPU is operated with a 20-MHz clock
  t_{CYC} = 30 ns (33 MHz) when the CPU is operated with a 33-MHz clock
- In x2 mode, t_{CYC} = 50 ns (20 MHz) when the CPU is operated with a 40-MHz clock
  t_{CYC} = 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	$V_{IH} = V_{DDE} - 0.4$ V
		Low level	$V_{IL} = 0.4$ 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 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 V_{DD} to V_{SS} voltage range.

(Unless otherwise specified: V_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
High-speed clock cycle time	t_{C3}	20.83	500	ns	
OSC3 clock input duty	t_{C3ED}	45	55	%	
OSC3 clock input rise time	t_{IF}		5	ns	
OSC3 clock input fall time	t_{IR}		5	ns	
BCLK high-level output delay time	t_{CD1}		35	ns	
BCLK low-level output delay time	t_{CD2}		35	ns	
Minimum reset pulse width	t_{RST}	6·t_{CYC}		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_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
BCLK clock output duty	t_{CBD}	40	60	%	

Common characteristics

(Unless otherwise specified: V_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
Address delay time	t_{AD}	–	10	ns	1
#CEx delay time (1)	t_{CE1}	–	10	ns	
#CEx delay time (2)	t_{CE2}	–	10	ns	
Wait setup time	t_{WTS}	18	–	ns	
Wait hold time	t_{WTH}	0	–	ns	
Read signal delay time (1)	t_{RDD1}		10	ns	2
Read data setup time	t_{RDS}	18		ns	
Read data hold time	t_{RDH}	0		ns	
Write signal delay time (1)	t_{WRD1}		10	ns	3
Write data delay time (1)	t_{WDD1}		10	ns	
Write data delay time (2)	t_{WDD2}	0	10	ns	
Write data hold time	t_{WDH}	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_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
Read signal delay time (2)	t_{RDD2}		10	ns	
Read signal pulse width	t_{RDW}	t_{CYC}(0.5+WC)-10		ns	
Read address access time (1)	t_{ACC1}		t_{CYC}(1+WC)-25	ns	
Chip enable access time (1)	t_{CEAC1}		t_{CYC}(1+WC)-25	ns	
Read signal access time (1)	t_{RDAC1}		t_{CYC}(0.5+WC)-25	ns	

SRAM write cycle

(Unless otherwise specified: V_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
Write signal delay time (2)	t_{WRD2}		10	ns	
Write signal pulse width	t_{WRW}	t_{CYC}(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, Ta=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
Read address access time (2)	t_{ACC2}		t_{CYC}(1+WC)-25	ns	
Chip enable access time (2)	t_{CEAC2}		t_{CYC}(1+WC)-25	ns	
Read signal access time (2)	t_{RDAC2}		t_{CYC}(0.5+WC)-25	ns	
Burst address access time	t_{ACCB}		t_{CYC}(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, Ta=-40°C to +85°C)

Item	Symbol	Min.	Max.	Unit	*
#BUSREQ signal setup time	t_{BRQS}	18		ns	
#BUSREQ signal hold time	t_{BRQH}	0		ns	
#BUSACK signal output delay time	t_{BAKD}		10	ns	
High-impedance → output delay time	t_{Z2E}		10	ns	
Output → high-impedance delay time	t_{B2Z}		10	ns	
#NMI pulse width	t_{NMIW}	30		ns	

Input, Output and I/O port

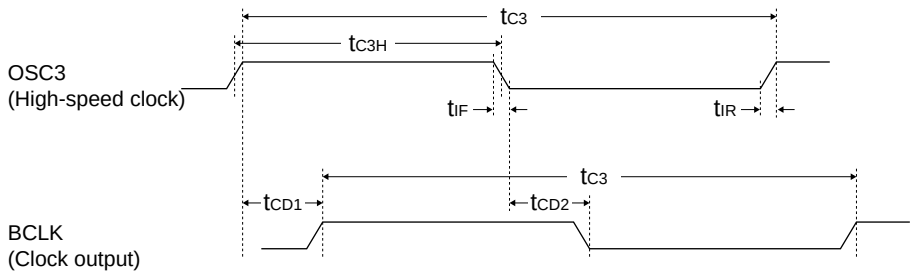
(Unless otherwise specified: V_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, V_{SS}=0V, Ta=-40°C to +85°C)

Item		Symbol	Min.	Max.	Unit	*
Input data setup time		t_{INPS}	20		ns	
Input data hold time		t_{INPH}	10		ns	
Output data delay time		t_{OUTD}		20	ns	
K-port interrupt input pulse width	SLEEP, HALT2 mode	t_{KINW}	30		ns	
	Others		2 × t_{CYC}		ns	

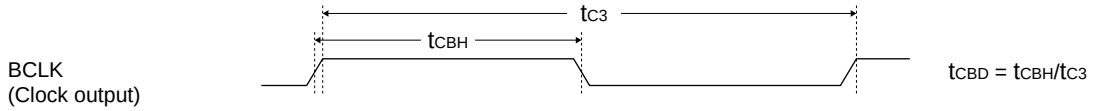
C33 Block AC Characteristic Timing Charts

Clock

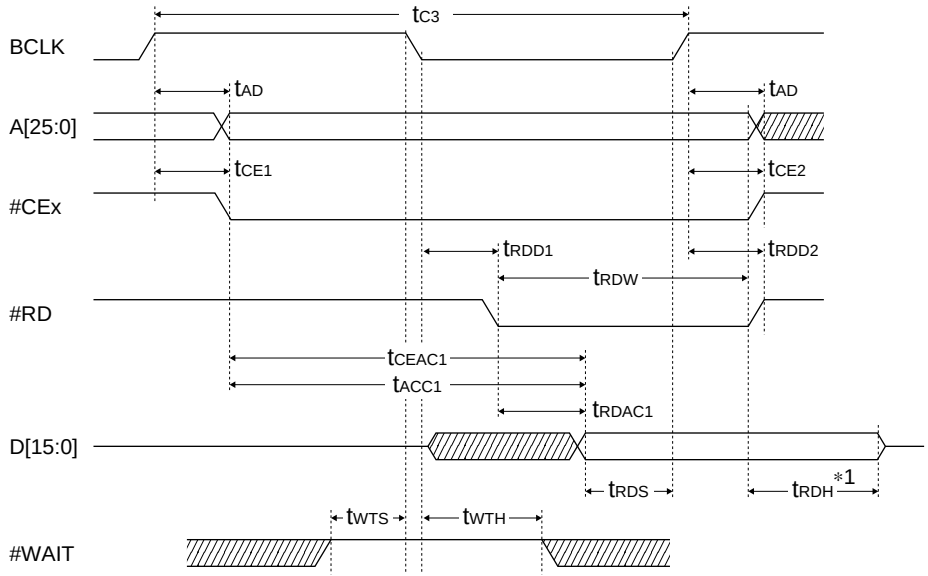
(1) When an external clock is input (in x1 speed mode):



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

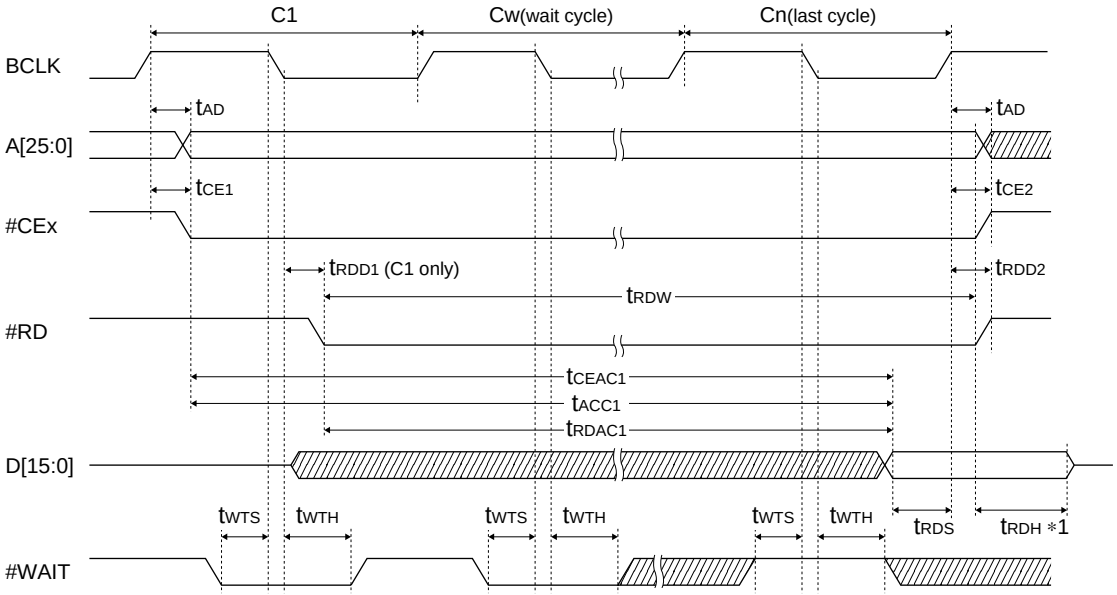


SRAM read cycle (basic cycle: 1 cycle)



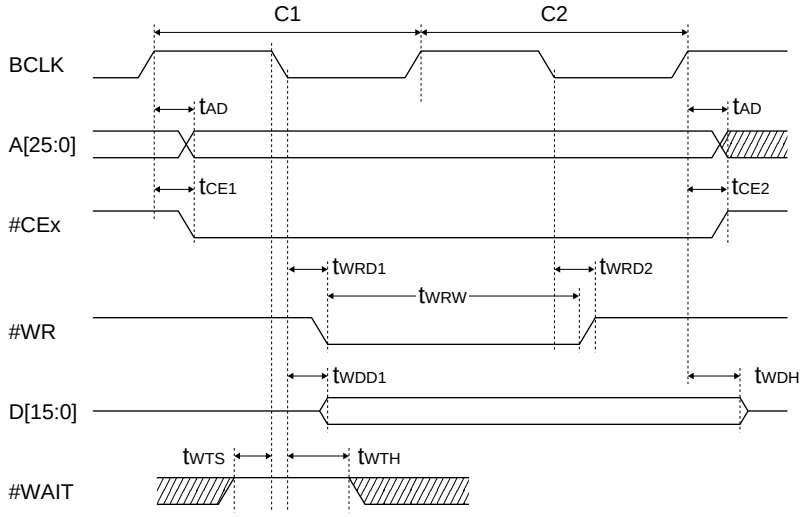
∗1 tRDH is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[25:0] signals.

SRAM read cycle (when a wait cycle is inserted)

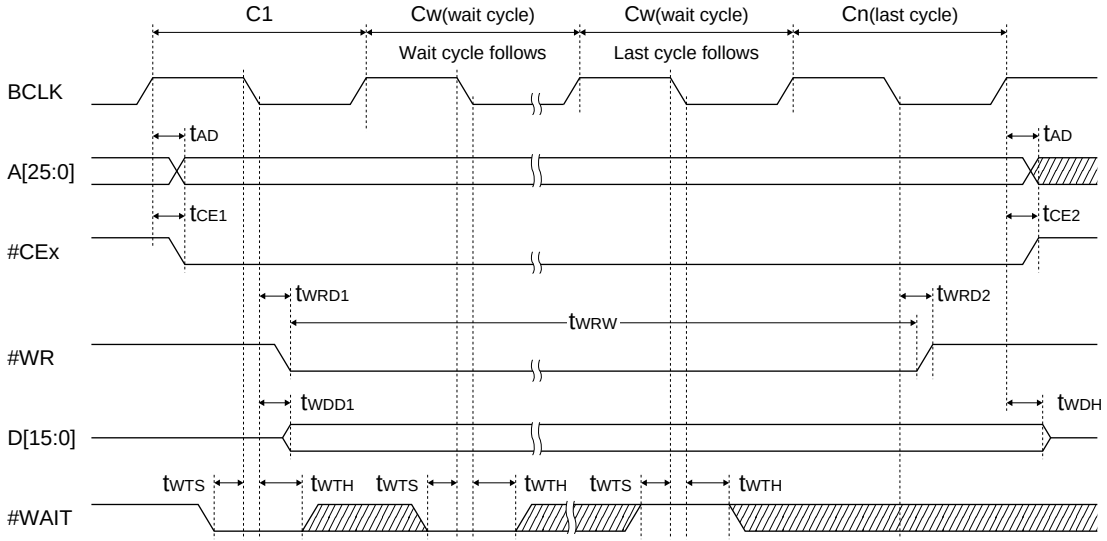


∗1 tRDH is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[25:0] signals.

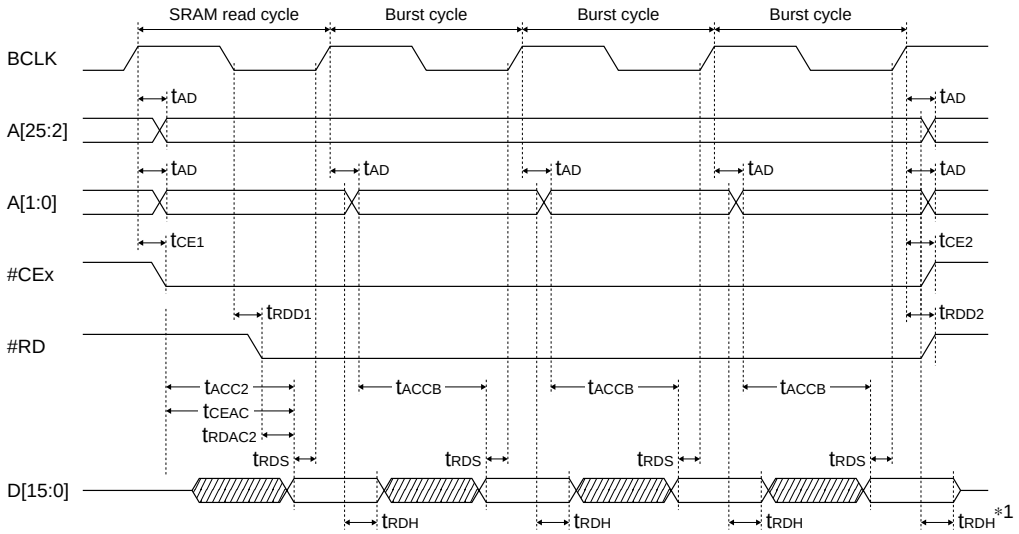
SRAM write cycle (basic cycle: 2 cycles)



SRAM write cycle (when wait cycles are inserted)

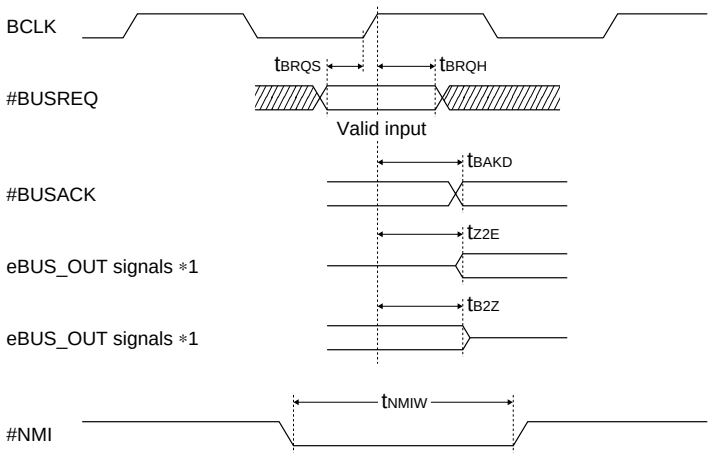


Burst ROM read cycle



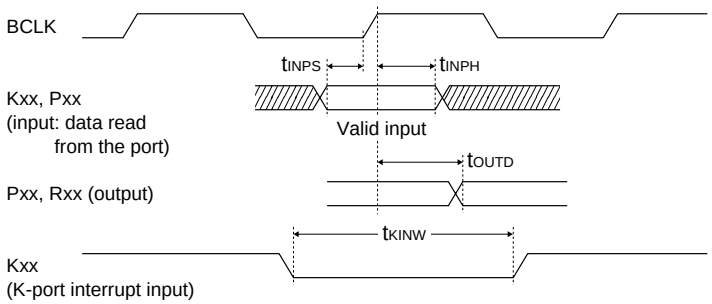
∗1 tRDH is measured with respect to the first signal change (negation) from among the #RD, #CEx and A[25:0] signals.

#BUSREQ, #BUSACK and #NMI timing



∗1 eBUS_OUT indicates the following pins:
A[25:0], #RD, #WRL, #WRH, #HCAS, #LCAS, #CEx, D[15:0]

Input, output and I/O port timing



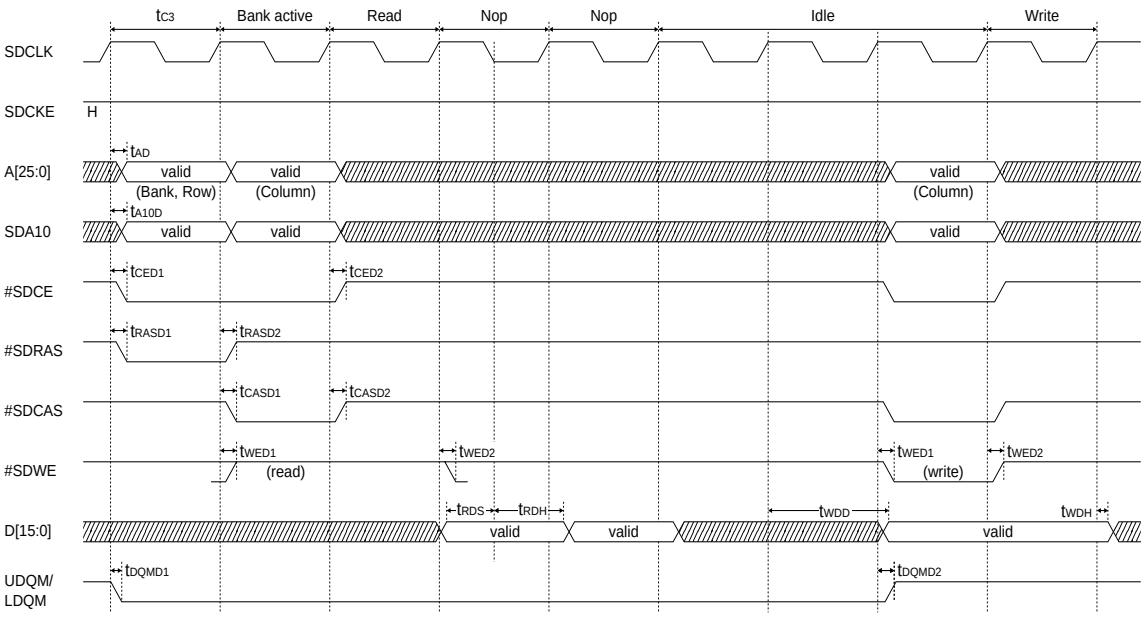
SDRAM AC Characteristics

SDRAM access cycle

(Unless otherwise specified: V_{DDE}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, Ta=-40°C to +85°C)

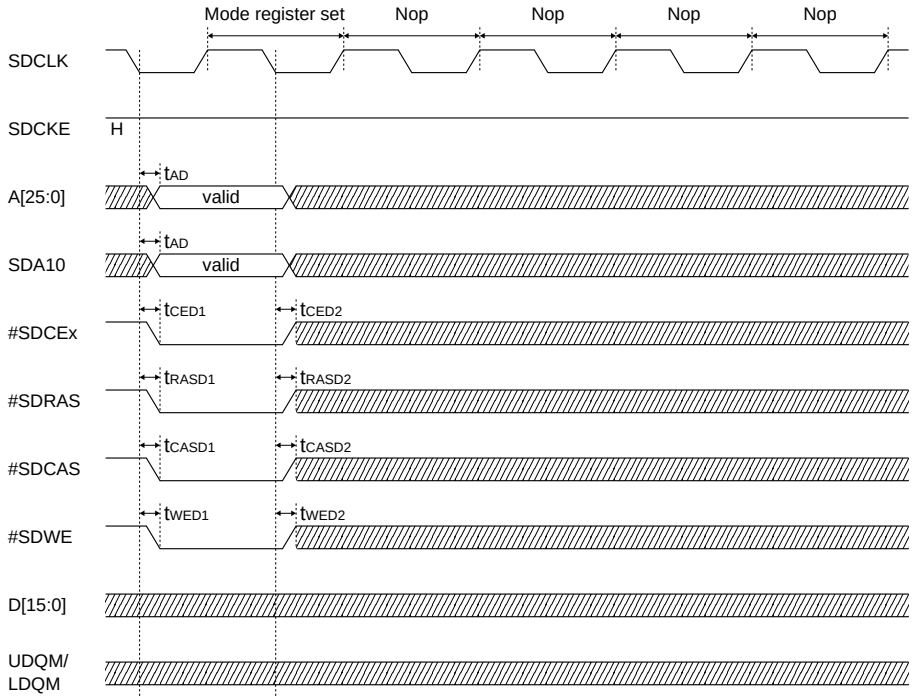
Item	Symbol	Min.	Max.	Unit	*
SDRAM clock cycle time	t_{C3}	18.8	–	ns	
Address delay time	t_{AD}	–	9.1	ns	
SDA10 delay time	t_{A10D}	–	7.4	ns	
#SDCE delay time (1)	t_{CED1}	–	6.6	ns	
#SDCE delay time (2)	t_{CED2}	–	7.4	ns	
#SDRAS signal delay time (1)	t_{RASD1}	–	7.0	ns	
#SDRAS signal delay time (2)	t_{RASD2}	–	6.9	ns	
#SDCAS signal delay time (1)	t_{CASD1}	–	7.0	ns	
#SDCAS signal delay time (2)	t_{CASD2}	–	6.2	ns	
UDQM, LDQM signal delay time (1)	t_{DQMD1}	–	6.6	ns	
UDQM, LDQM signal delay time (2)	t_{DQMD2}	–	9.0	ns	
SDCKE signal delay time (1)	t_{CKED1}	–	5.7	ns	
SDCKE signal delay time (2)	t_{CKED2}	–	6.5	ns	
#SDWE signal delay time (1)	t_{WED1}	–	6.7	ns	
#SDWE signal delay time (2)	t_{WED2}	–	7.5	ns	
Read data setup time	t_{RDS}	7.1	–	ns	
Read data hold time	t_{RDH}	0	–	ns	
Write data delay time	t_{WDD}	–	25.6	ns	
Write data hold time	t_{WDH}	0	3.5	ns	

Note: All the signals except for DATA change at the rising edge of the SDRAM clock.

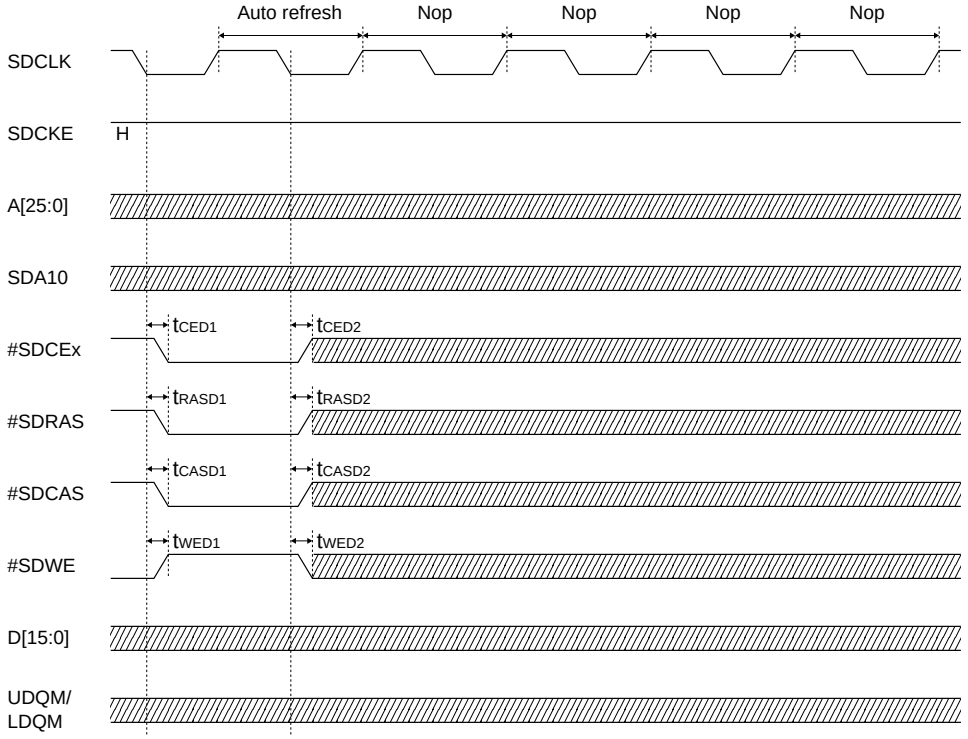


* Read: CAS latency = 2, burst length = 2 Write: single write

SDRAM mode-register-set cycle

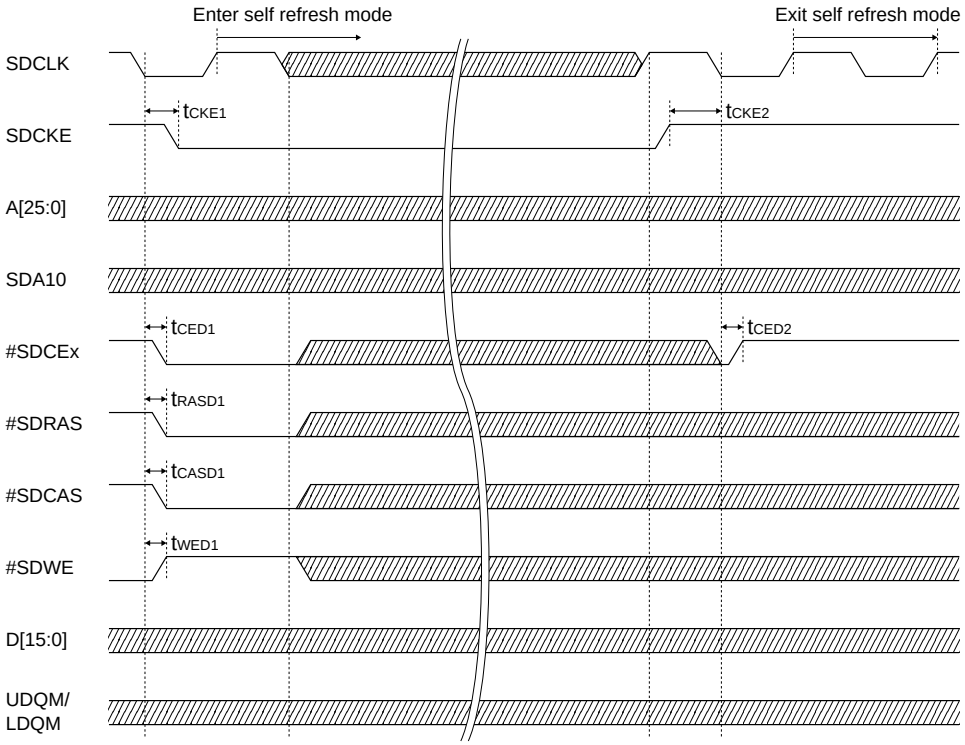


SDRAM auto-refresh cycle



∗ A precharge cycle is necessary before entering the auto refresh mode.

SDRAM self-refresh cycle

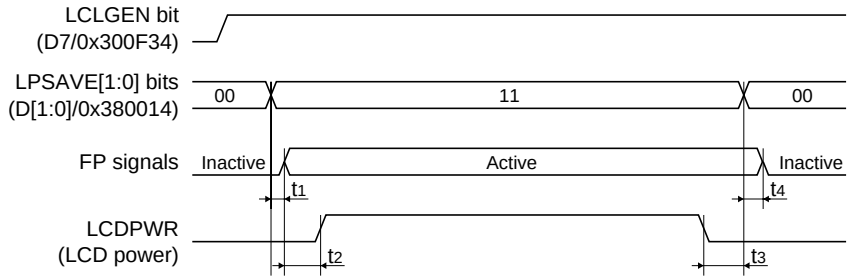


∗ A precharge cycle is necessary before entering the self refresh mode.

LCDC AC Characteristics

Conditions: V_{DDE} = 2.7V to 3.6V
Ta = -40°C to 85°C
T_{rise} and T_{fall} for all outputs should be < 5 ns (10% ~ 90%)
C_L = 60 pF (LCD panel interface)

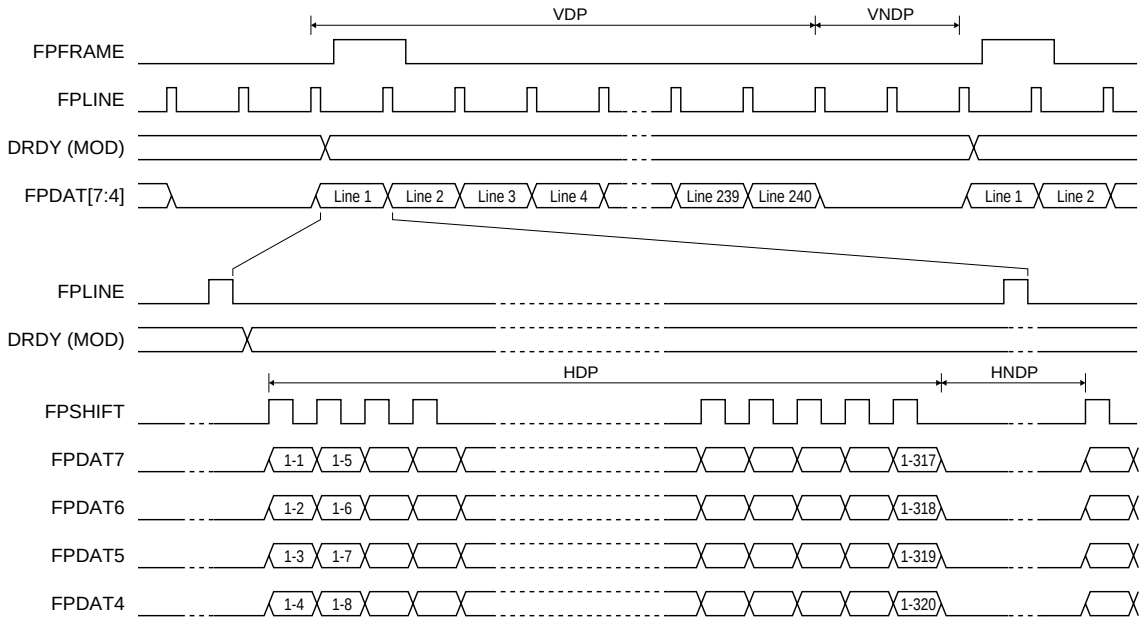
Power up/down timing



Symbol	Parameter	Min.	Typ.	Max.	Unit
t_1	Power Save inactive to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY active			1	Frame
t_2	FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY active to LCDPWR active	1			Frame
t_3	Power Save active to LCDPWR inactive	1			Frame
t_4	Power Save active to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY inactive			1	Frame

note) Any I/O port can be used as the LCDPWR output for controlling the power supply to the LCD panel. Note, however, that the t_2 and t_3 timing conditions must be satisfied when controlling the signal.

4-bit single monochrome panel timing



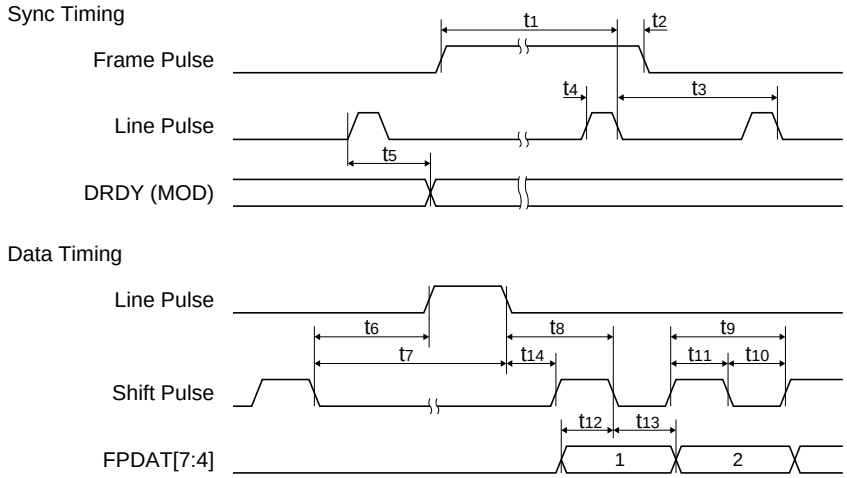
* Diagram drawn with 2 FPLINE vertical blank period
  Example timing for a 320 × 240 panel
  For this timing diagram MASK (DD/0x380202) is set to "1"

VDP = Vertical Display Period = VSIZE[9:0] + 1 (lines)
VSIZE[9:0] (D[9:0]/0x38004C)

VNDP = Vertical Non-Display Period = VNDP[5:0] (lines)
VNDP[5:0] (D[5:0]/0x38004A)

HDP = Horizontal Display Period = (HSIZE[6:0] + 1) × 8 (Ts)
HSIZE[6:0] (D[6:0]/0x380042)

HNDP = Horizontal Non-Display Period = (HNDP[4:0] + 4) × 8 (Ts)
HNDP[4:0] (D[4:0]/0x380040)



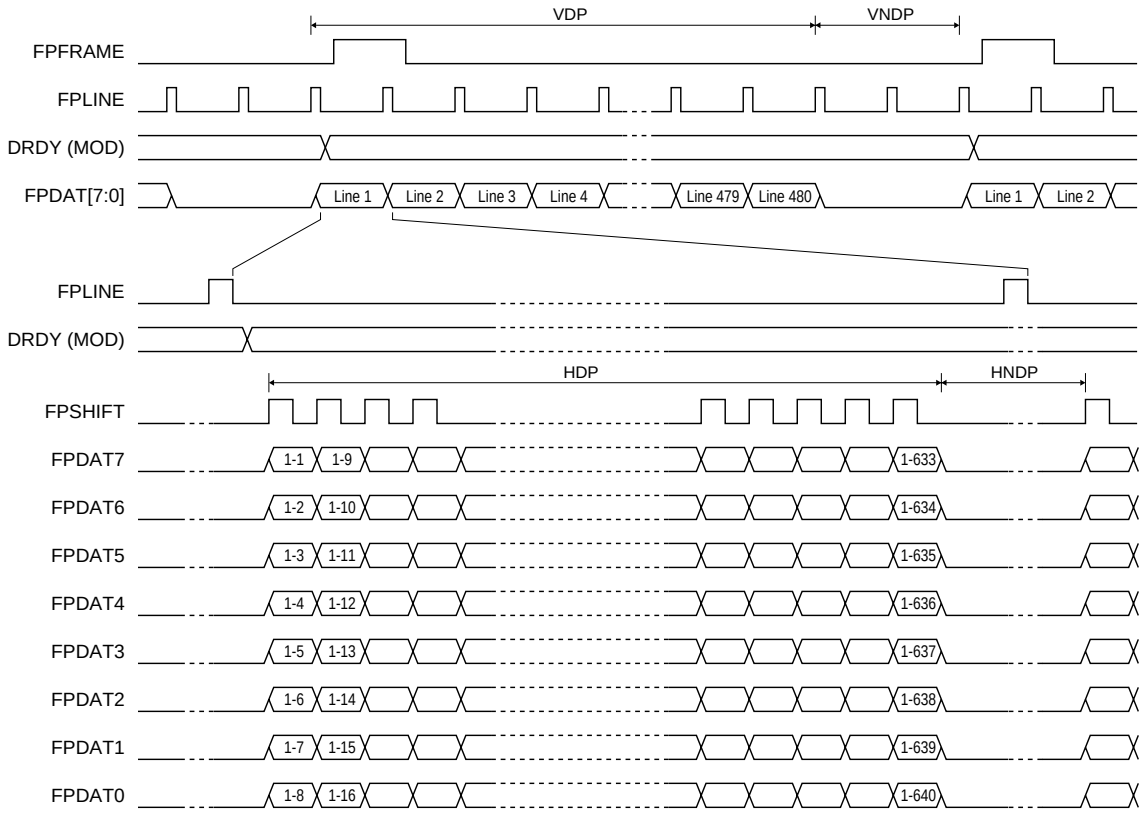
Note: For this timing diagram MASK (DD/0x380202) is set to "1".

4-bit Single Monochrome Panel AC Timing

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t_2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t_3	Line Pulse period	note 3			
t_4	Line Pulse width	9			Ts
t_5	MOD delay from Line Pulse rising edge	1			Ts
t_6	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t_7	Shift Pulse falling edge to Line Pulse falling edge	note 5			
t_8	Line Pulse falling edge to Shift Pulse falling edge	t_{14}+2			Ts
t_9	Shift Pulse period	4			Ts
t_{10}	Shift Pulse width low	2			Ts
t_{11}	Shift Pulse width high	2			Ts
t_{12}	FPDAT[7:4] setup to Shift Pulse falling edge	2			Ts
t_{13}	FPDAT[7:4] hold from Shift Pulse falling edge	2			Ts
t_{14}	Line Pulse falling edge to Shift Pulse rising edge	23			Ts

note) 1. Ts = pixel clock period
2. t_{1min} = t_{3min} - 9 (Ts)
3. t_{3min} = (HSIZE[6:0] + 1) × 8 + (HNDP[4:0] + 4) × 8 (Ts)
4. t_{6min} = HNDP[4:0] × 8 + 2 (Ts)
5. t_{7min} = HNDP[4:0] × 8 + 11 (Ts)

8-bit single monochrome panel timing



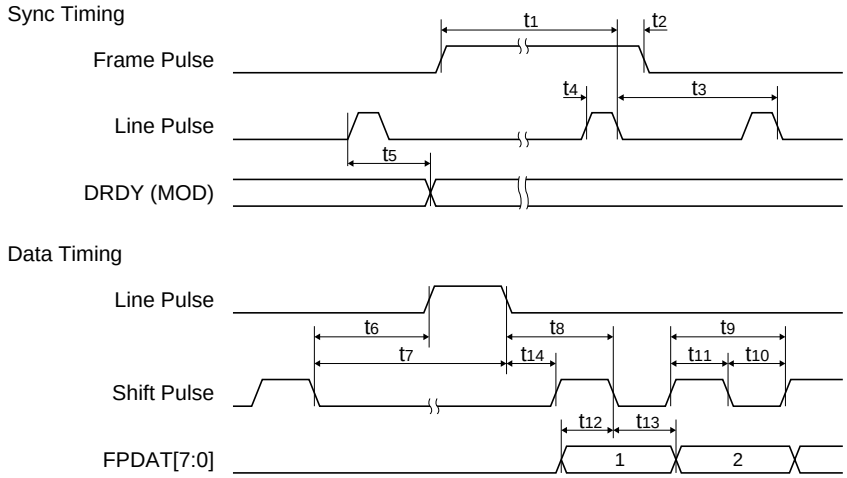
∗ Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640 × 480 panel
For this timing diagram MASK (DD/0x380202) is set to "1"

VDP = Vertical Display Period = VSIZE[9:0] + 1 (lines)
VSIZE[9:0] (D[9:0]/0x38004C)

VNDP = Vertical Non-Display Period = VNDP[5:0] (lines)
VNDP[5:0] (D[5:0]/0x38004A)

HDP = Horizontal Display Period = (HSIZE[6:0] + 1) × 8 (Ts)
HSIZE[6:0] (D[6:0]/0x380042)

HNDP = Horizontal Non-Display Period = (HNDP[4:0] + 4) × 8 (Ts)
HNDP[4:0] (D[4:0]/0x380040)



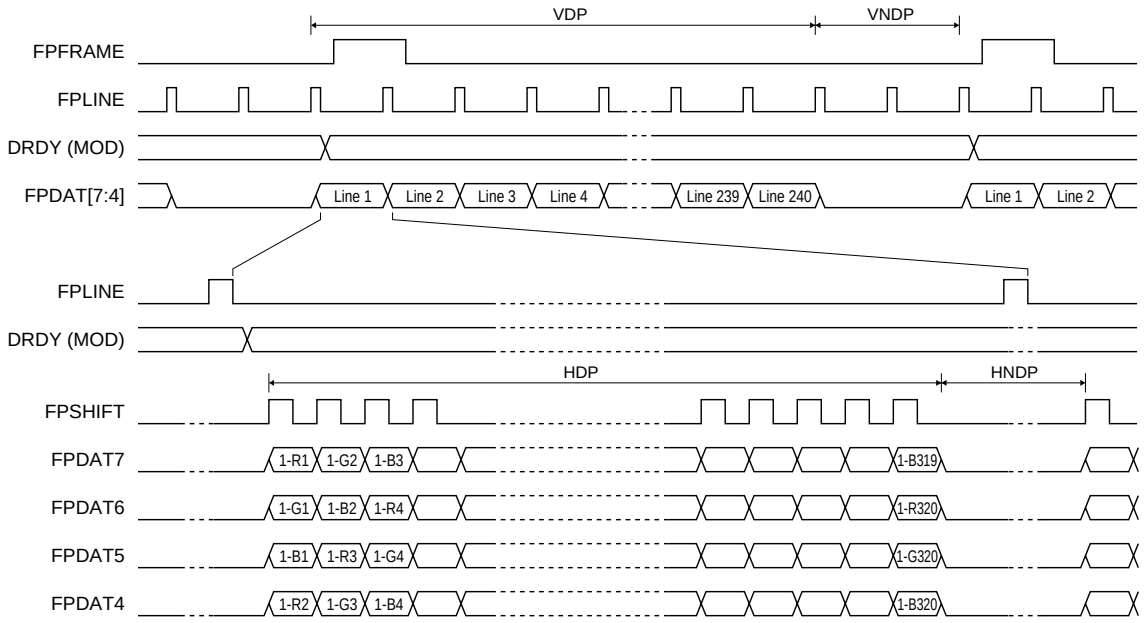
Note: For this timing diagram MASK (DD/0x380202) is set to "1".

8-bit Single Monochrome Panel AC Timing

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t_2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t_3	Line Pulse period	note 3			
t_4	Line Pulse width	9			Ts
t_5	MOD delay from Line Pulse rising edge	1			Ts
t_6	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t_7	Shift Pulse falling edge to Line Pulse falling edge	note 5			
t_8	Line Pulse falling edge to Shift Pulse falling edge	t_{14}+4			Ts
t_9	Shift Pulse period	8			Ts
t_{10}	Shift Pulse width low	4			Ts
t_{11}	Shift Pulse width high	4			Ts
t_{12}	FPDAT[7:0] setup to Shift Pulse falling edge	4			Ts
t_{13}	FPDAT[7:0] hold from Shift Pulse falling edge	4			Ts
t_{14}	Line Pulse falling edge to Shift Pulse rising edge	23			Ts

note) 1. Ts = pixel clock period
2. t_{1min} = t_{3min} - 9 (Ts)
3. t_{3min} = (HSIZE[6:0] + 1) × 8 + (HNDP[4:0] + 4) × 8 (Ts)
4. t_{6min} = HNDP[4:0] × 8 + 4 (Ts)
5. t_{7min} = HNDP[4:0] × 8 + 13 (Ts)

4-bit single color panel timing



∗ Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 320 × 240 panel

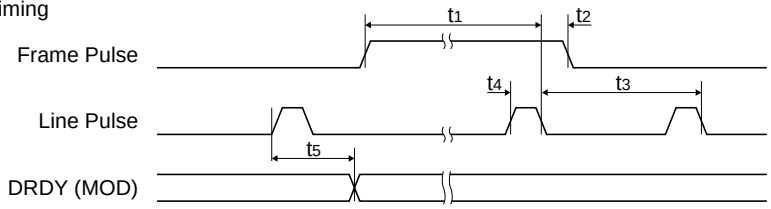
VDP = Vertical Display Period = VSIZE[9:0] + 1 (lines)
VSIZE[9:0] (D[9:0]/0x38004C)

VNDP = Vertical Non-Display Period = VNDP[5:0] (lines)
VNDP[5:0] (D[5:0]/0x38004A)

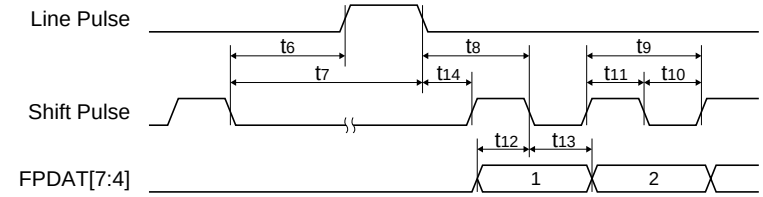
HDP = Horizontal Display Period = (HSIZE[6:0] + 1) × 8 (Ts)
HSIZE[6:0] (D[6:0]/0x380042)

HNDP = Horizontal Non-Display Period = (HNDP[4:0] + 4) × 8 (Ts)
HNDP[4:0] (D[4:0]/0x380040)

Sync Timing



Data Timing

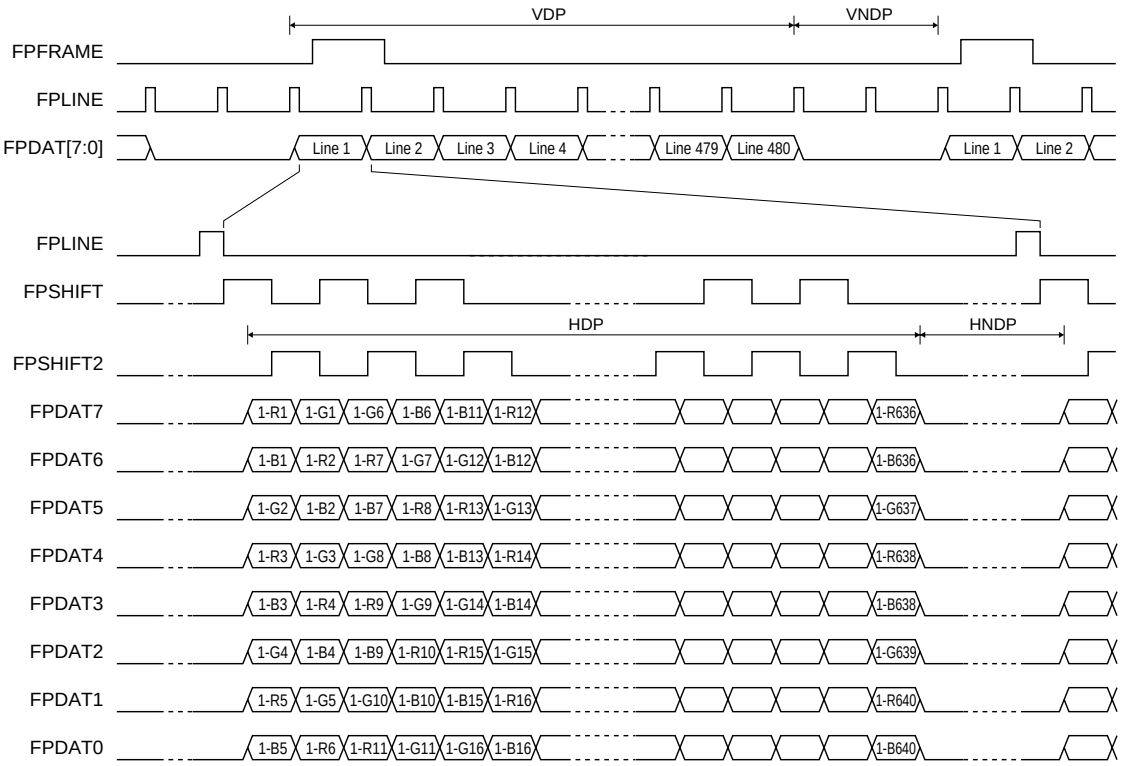


4-bit Single Color Panel AC Timing

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t_2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t_3	Line Pulse period	note 3			
t_4	Line Pulse width	9			Ts
t_5	MOD delay from Line Pulse rising edge	1			Ts
t_6	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t_7	Shift Pulse falling edge to Line Pulse falling edge	note 5			
t_8	Line Pulse falling edge to Shift Pulse falling edge	t_{14}+0.5			Ts
t_9	Shift Pulse period	1			Ts
t_{10}	Shift Pulse width low	0.5			Ts
t_{11}	Shift Pulse width high	0.5			Ts
t_{12}	FPDAT[7:4] setup to Shift Pulse falling edge	0.5			Ts
t_{13}	FPDAT[7:4] hold from Shift Pulse falling edge	0.5			Ts
t_{14}	Line Pulse falling edge to Shift Pulse rising edge	23 (24)			Ts

note) 1. Ts = pixel clock period
2. t_{1min} = t_{3min} - 9 (Ts)
3. t_{3min} = (HSIZE[6:0] + 1) × 8 + (HNDP[4:0] + 4) × 8 (Ts)
4. t_{6min} = HNDP[4:0] × 8 + 1.5 (Ts)
5. t_{7min} = HNDP[4:0] × 8 + 10.5 (Ts)

8-bit single color panel timing (Format 1)



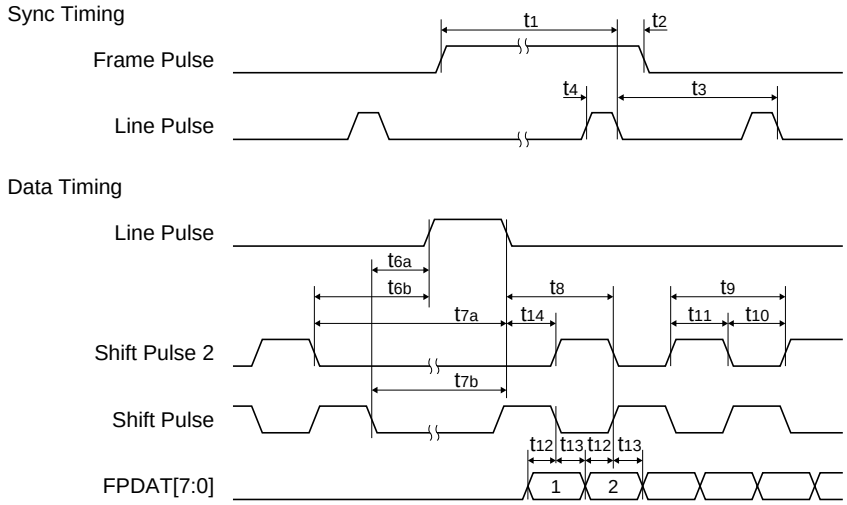
* Diagram drawn with 2 FPLINE vertical blank period
  Example timing for a 640 × 480 panel

VDP = Vertical Display Period = VSIZE[9:0] + 1 (lines)
VSIZE[9:0] (D[9:0]/0x38004C)

VNDP = Vertical Non-Display Period = VNDP[5:0] (lines)
VNDP[5:0] (D[5:0]/0x38004A)

HDP = Horizontal Display Period = (HSIZE[6:0] + 1) × 8 (Ts)
HSIZE[6:0] (D[6:0]/0x380042)

HNDP = Horizontal Non-Display Period = (HNDP[4:0] + 4) × 8 (Ts)
HNDP[4:0] (D[4:0]/0x380040)

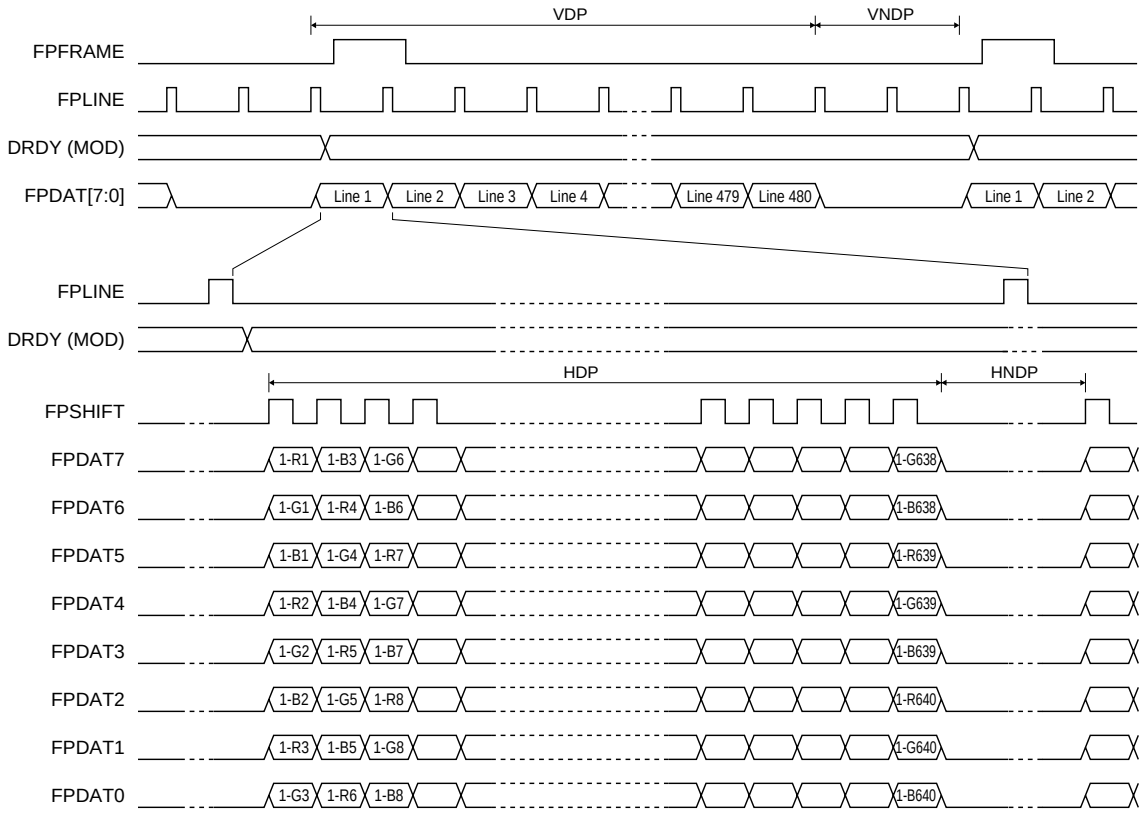


8-bit Single Color Panel AC Timing (Format 1)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t_2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t_3	Line Pulse period	note 3			
t_4	Line Pulse width	9			Ts
t_{6a}	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t_{6b}	Shift Pulse 2 falling edge to Line Pulse rising edge	note 5			
t_{7a}	Shift Pulse 2 falling edge to Line Pulse falling edge	note 6			
t_{7b}	Shift Pulse falling edge to Line Pulse falling edge	note 7			
t_8	Line Pulse falling edge to Shift Pulse rising, Shift Pulse 2 falling edge	t_{14}+2			Ts
t_9	Shift Pulse 2, Shift Pulse period	4			Ts
t_{10}	Shift Pulse 2, Shift Pulse width low	2			Ts
t_{11}	Shift Pulse 2, Shift Pulse width high	2			Ts
t_{12}	FPDAT[7:0] setup to Shift Pulse 2, Shift Pulse falling edge	1			Ts
t_{13}	FPDAT[7:0] hold from Shift Pulse 2, Shift Pulse falling edge	1			Ts
t_{14}	Line Pulse falling edge to Shift Pulse rising edge	23 (25)			Ts

note) 1. Ts = pixel clock period
2. t_{1min} = t_{3min} - 9 (Ts)
3. t_{3min} = (HSIZE[6:0] + 1) × 8 + (HNDP[4:0] + 4) × 8 (Ts)
4. t_{6amin} = HNDP[4:0] × 8 + t_{13} - t_{10} + 1 (Ts)
5. t_{6bmin} = HNDP[4:0] × 8 + t_{13} + 1 (Ts)
6. t_{7amin} = HNDP[4:0] × 8 + 11 (Ts)
7. t_{7bmin} = HNDP[4:0] × 8 + 11 - t_{10} (Ts)

8-bit single color panel timing (Format 2)



∗ Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640 × 480 panel

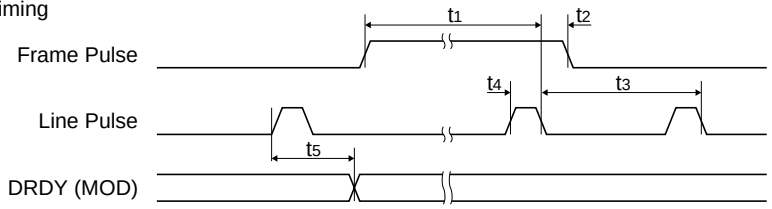
VDP = Vertical Display Period = VSIZE[9:0] + 1 (lines)
VSIZE[9:0] (D[9:0]/0x38004C)

VNDP = Vertical Non-Display Period = VNDP[5:0] (lines)
VNDP[5:0] (D[5:0]/0x38004A)

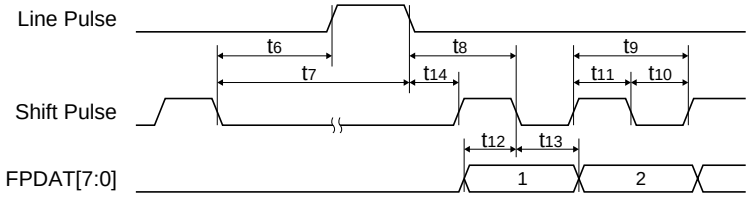
HDP = Horizontal Display Period = (HSIZE[6:0] + 1) × 8 (Ts)
HSIZE[6:0] (D[6:0]/0x380042)

HNDP = Horizontal Non-Display Period = (HNDP[4:0] + 4) × 8 (Ts)
HNDP[4:0] (D[4:0]/0x380040)

Sync Timing



Data Timing



8-bit Single Color Panel AC Timing (Format 2)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t_2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t_3	Line Pulse period	note 3			
t_4	Line Pulse width	9			Ts
t_5	MOD delay from Line Pulse rising edge	1			Ts
t_6	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t_7	Shift Pulse falling edge to Line Pulse falling edge	note 5			
t_8	Line Pulse falling edge to Shift Pulse falling edge	t_{14}+2			Ts
t_9	Shift Pulse period	2 (3)			Ts
t_{10}	Shift Pulse width low	1			Ts
t_{11}	Shift Pulse width high	1			Ts
t_{12}	FPDAT[7:0] setup to Shift Pulse falling edge	1			Ts
t_{13}	FPDAT[7:0] hold from Shift Pulse falling edge	1			Ts
t_{14}	Line Pulse falling edge to Shift Pulse rising edge	23			Ts

note) 1. Ts = pixel clock period
2. t_{1min} = t_{3min} - 9 (Ts)
3. t_{3min} = (HSIZE[6:0] + 1) × 8 + (HNDP[4:0] + 4) × 8 (Ts)
4. t_{6min} = HNDP[4:0] × 8 + 1 (Ts)
5. t_{7min} = HNDP[4:0] × 8 + 10 (Ts)

<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

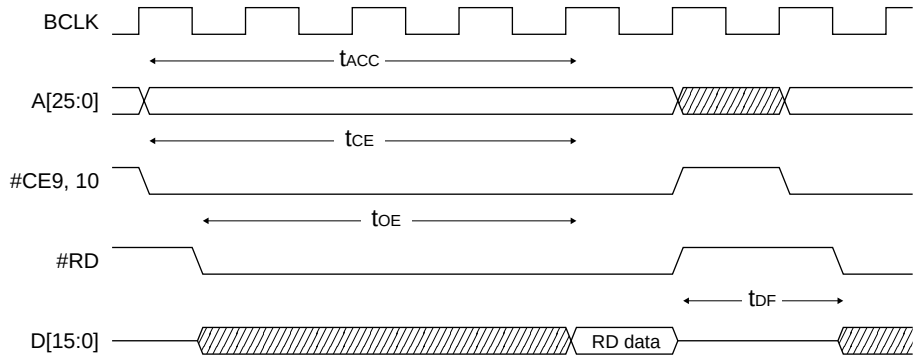
Burst ROM and mask ROM interface setup examples

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

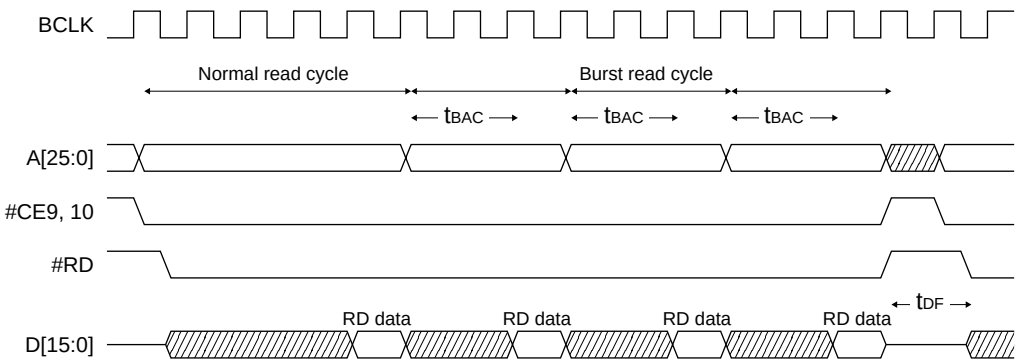
Burst ROM and mask ROM interface timing

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

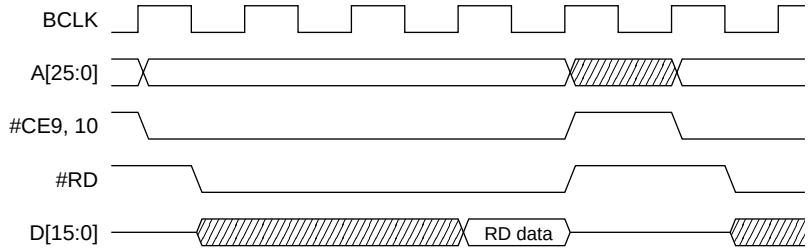
ROM: 100ns, CPU: 33MHz, normal read



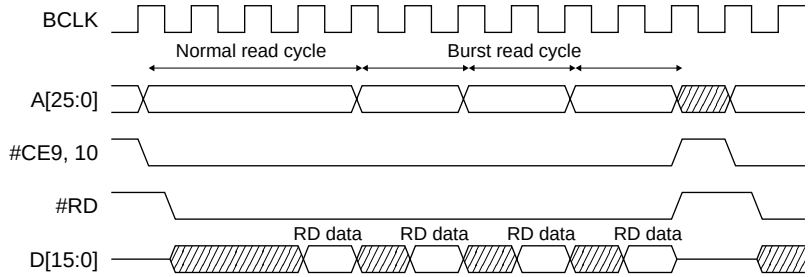
ROM: 100ns, CPU: 33MHz, burst read



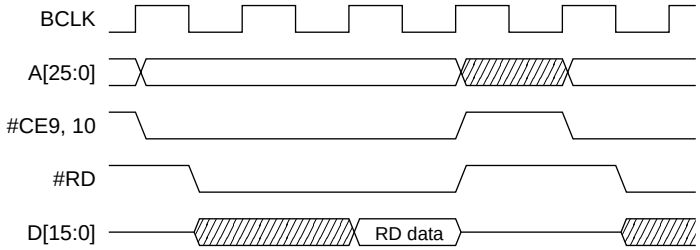
ROM: 100ns, CPU: 25MHz, normal read



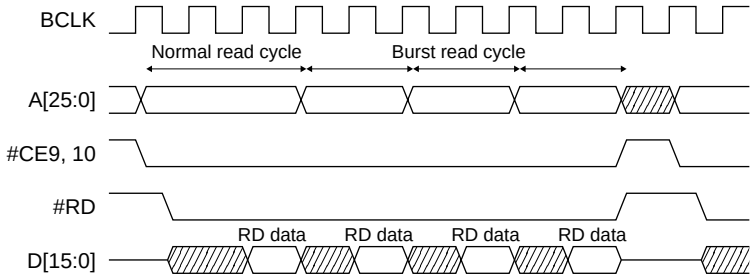
ROM: 100ns, CPU: 25MHz, burst read



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



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



SRAM (55ns)

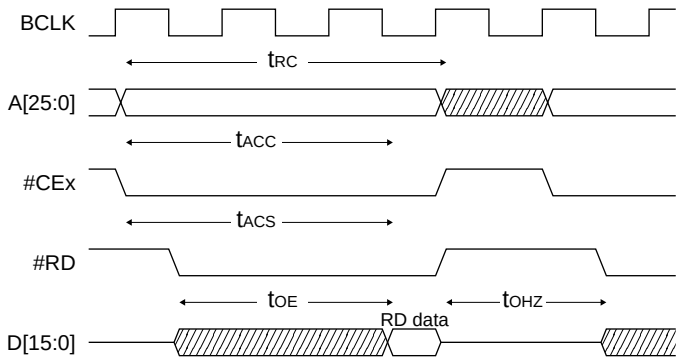
SRAM interface setup examples – 55ns

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

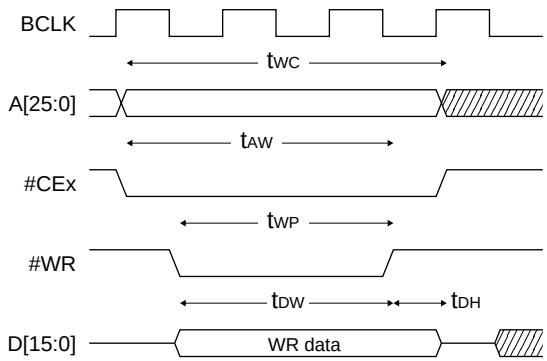
SRAM interface timing – 55ns

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

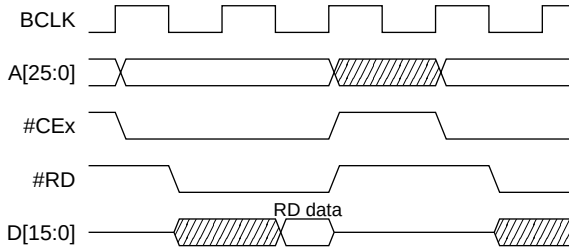
SRAM: 55ns, CPU: 33/25MHz, read cycle



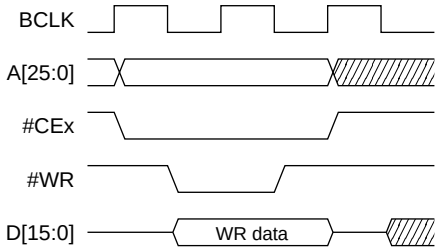
SRAM: 55ns, CPU: 33/25MHz, write cycle



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



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



SRAM (70ns)

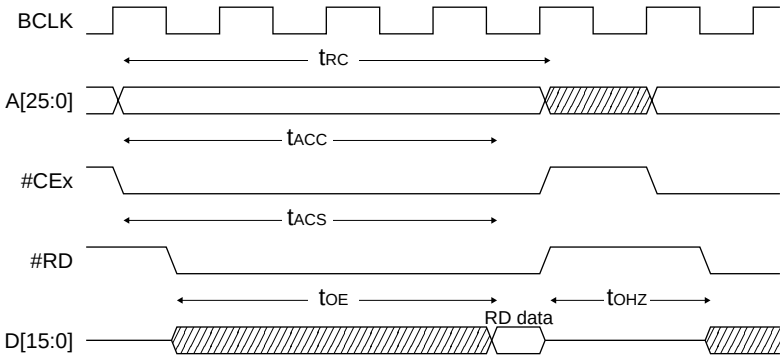
SRAM interface setup examples – 70ns

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

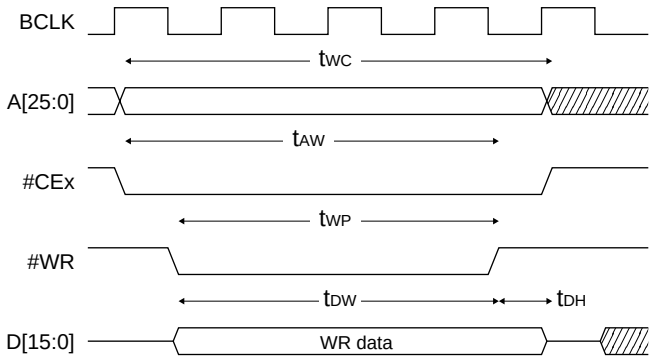
SRAM interface timing – 70ns

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

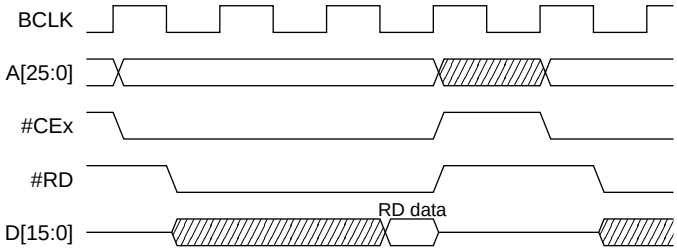
SRAM: 70ns, CPU: 33MHz, read cycle



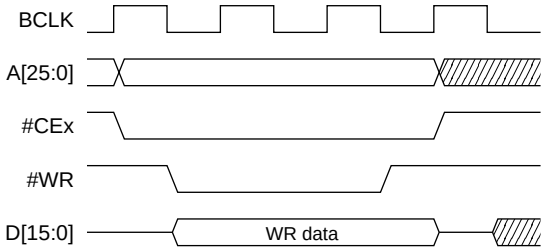
SRAM: 70ns, CPU: 33MHz, write cycle



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
<Read cycle>									
Read cycle time	t_{RC}	300	–	15	450	12	480	10	500
Address access time	t_{ACC}	–	250	15	450	12	480	10	500
#CE access time	t_{ACS}	–	250	15	450	12	480	10	500
#OE access time	t_{OE}	–	250	14.5	435	11.5	460	9.5	475
Output disable delay time	t_{OHZ}	10	150	3.5	105	3.5	140	3.5	175
<Write cycle>									
Write cycle time	t_{WC}	430	–	15	450	12	480	10	500
Address enable time	t_{AW}	400	–	14.5	435	11.5	460	9.5	475
Write pulse width	t_{WP}	400	–	14	420	11	440	9	450
Input data setup time	t_{DW}	100	–	14	420	11	440	9	450
Input data hold time *3	t_{DH}	30	–	0.5	15	0.5	20	0.5	25

*1 The C33 STD 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.

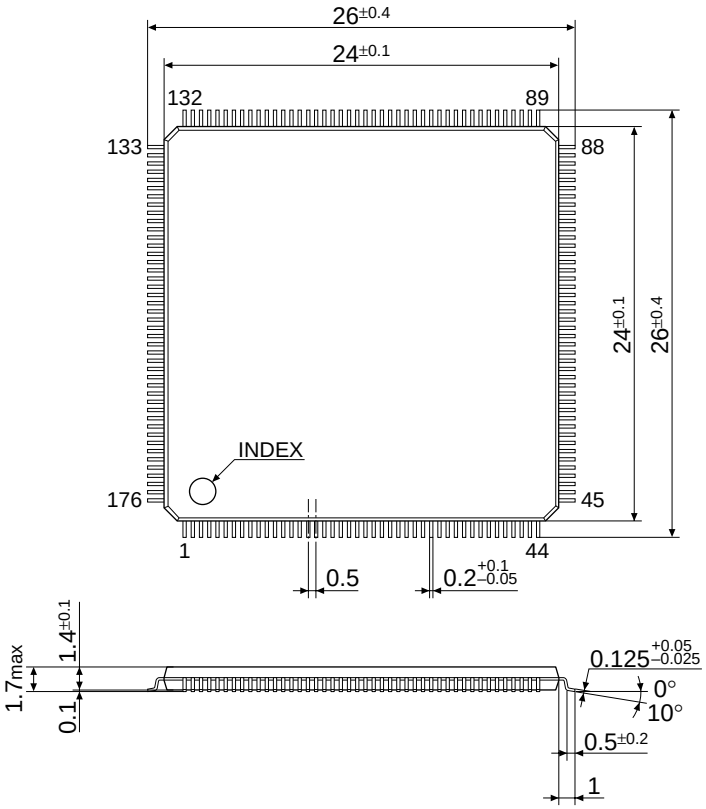
*3 If the data hold time that can be set is not sufficient for the device, secure it by connecting a bus repeater to the external data bus D[15:0] or by inserting a latch at the output side of the external system interface.

I-7 Package

Plastic Package

QFP21-176pin

(Unit: mm)



Thermal Package Resistance

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

Chip temperature (Tj) = Ta + ($P_D \times \theta$) (°C)

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

Thermal resistance of the QFP21-176pin package

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

Thermal resistance (θj-a) = 33.3°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)

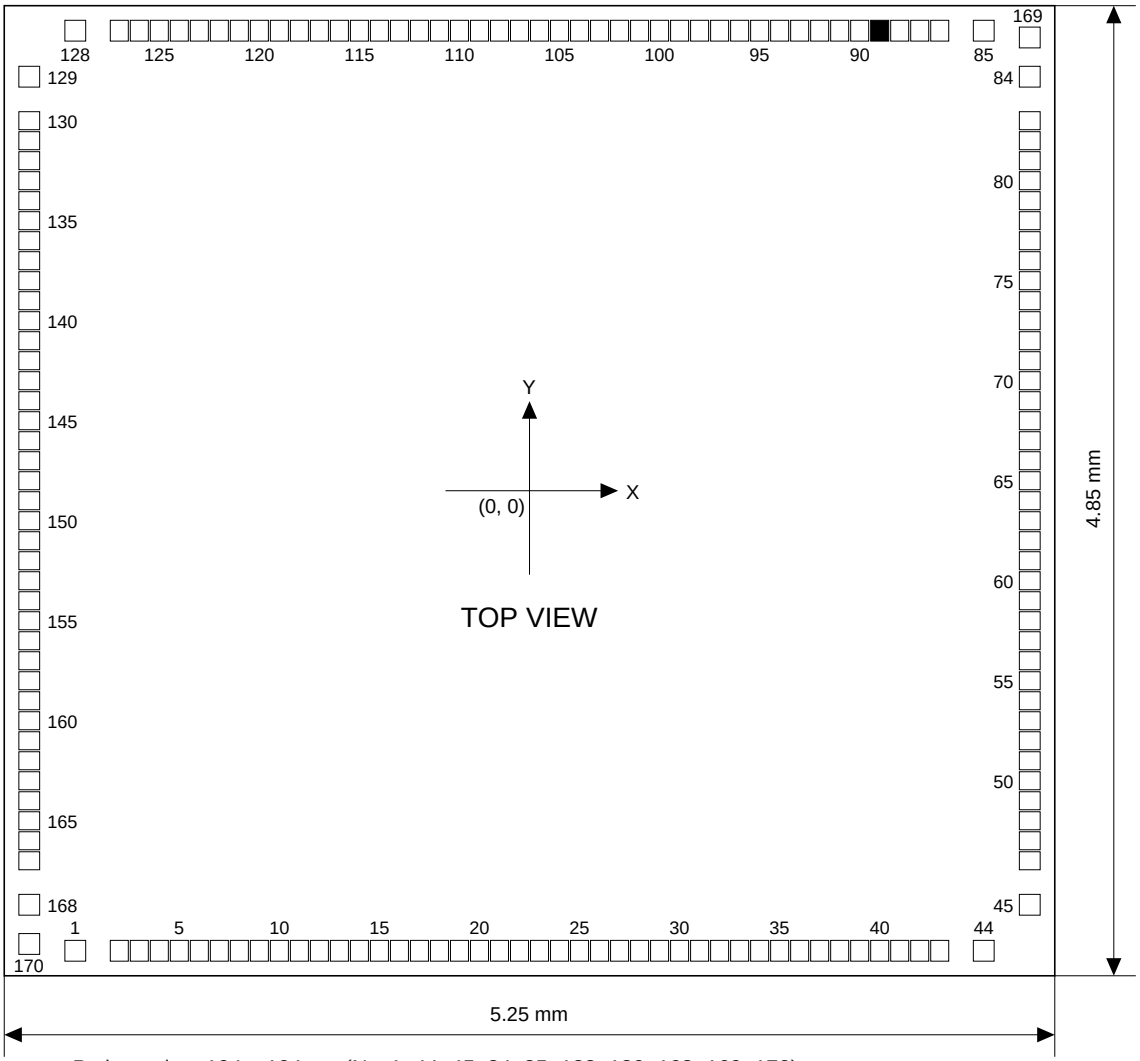
Thermal resistance = 90 to 100°C/W

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.

I-8 Pad Layout

Pad Layout Diagram



Pad opening: 104 × 104 µm (No. 1, 44, 45, 84, 85, 128, 129, 168, 169, 170)
90 × 104 µm (No. 2–43, 86–127)
104 × 90 µm (No. 46–83, 130–167)

Pad Coordinate

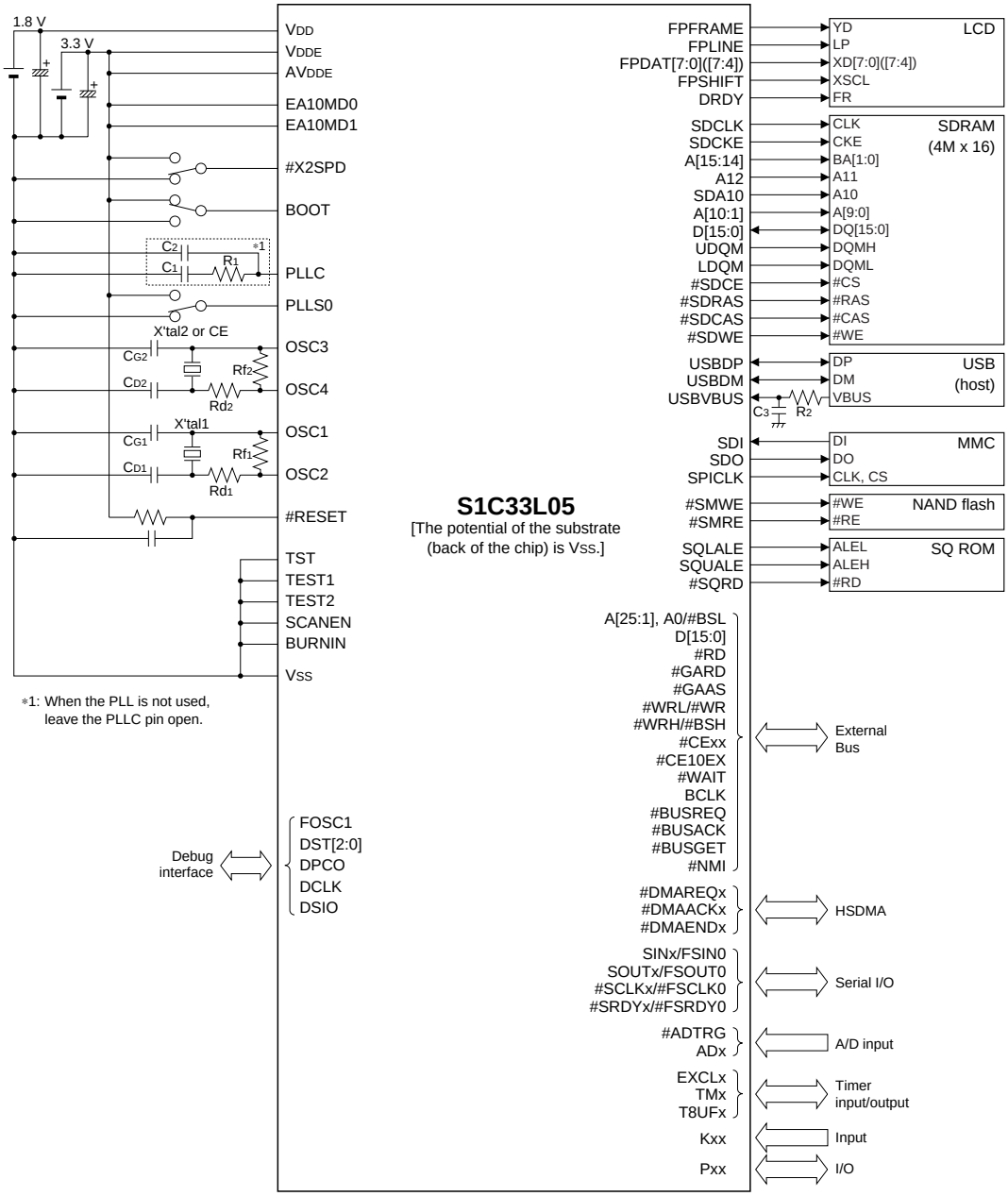
(Unit: µm)

No.	Pad name	X	Y	No.	Pad name	X	Y
1	**D9**	-2270	-2300	51	**DST1**/P11/EXCL1/T8UF1	2500	-1350
2	**D8**	-2050	-2300	52	**DST0**/P10/EXCL0/T8UF0	2500	-1250
3	**VDDE**	-1950	-2300	53	**VDD**	2500	-1150
4	**D7**	-1850	-2300	54	**#NMI**	2500	-1050
5	**D6**	-1750	-2300	55	**#RESET**	2500	-950
6	**D5**	-1650	-2300	56	**VSS**	2500	-850
7	**D4**	-1550	-2300	57	**K60**/AD0	2500	-750
8	**D3**	-1450	-2300	58	**K61**/AD1	2500	-650
9	**D2**	-1350	-2300	59	**K62**/AD2	2500	-550
10	**D1**	-1250	-2300	60	**K63**/AD3	2500	-450
11	**D0**	-1150	-2300	61	**K64**/AD4	2500	-350
12	**VSS**	-1050	-2300	62	**TEST0**	2500	-250
13	**P30**/#WAIT/#CE4&5/PA2	-950	-2300	63	**AVDDE**	2500	-150
14	**PD0**/#SQRD	-850	-2300	64	**K53**/#DMAREQ2	2500	-50
15	**PD1**/SQLALE	-750	-2300	65	**K52**/#ADTRG	2500	50
16	**PD2**/SQUALE	-650	-2300	66	**K51**/#DMAREQ1	2500	150
17	**PD3**	-550	-2300	67	**K50**/#DMAREQ0	2500	250
18	**PD4**	-450	-2300	68	**VSS**	2500	350
19	**PD5**	-350	-2300	69	**OSC1**	2500	450
20	**PD6**	-250	-2300	70	**OSC2**	2500	550
21	**PD7**	-150	-2300	71	**VDDE**	2500	650
22	**VDDE**	-50	-2300	72	**BURNIN**	2500	750
23	**P22**/TM0	50	-2300	73	**SCANEN**	2500	850
24	**P23**/TM1	150	-2300	74	**TEST1**	2500	950
25	**P24**/TM2/#SRDY2	250	-2300	75	**VDD**	2500	1050
26	**P25**/TM3/#SCLK2	350	-2300	76	**OSC3**	2500	1150
27	**P26**/TM4/SOUT2	450	-2300	77	**OSC4**	2500	1250
28	**P27**/TM5/SIN2	550	-2300	78	**VSS**	2500	1350
29	**VSS**	650	-2300	79	**P07**/#SRDY1/#DMAEND3	2500	1450
30	**PB7**/FPDAT7	750	-2300	80	**P06**/#SCLK1/#DMAACK3	2500	1550
31	**PB6**/FPDAT6	850	-2300	81	**P05**/#SOUT1/#DMAEND2	2500	1650
32	**PB5**/FPDAT5	950	-2300	82	**P04**/#SIN1/#DMAACK2	2500	1750
33	**PB4**/FPDAT4	1050	-2300	83	**P03**/#SRDY0/#FSRDY0	2500	1850
34	**VDD**	1150	-2300	84	**P02**/#SCLK0/#FSCLK0	2500	2070
35	**PB3**/FPDAT3	1250	-2300	85	**P01**/SOUT0/FSOUT0	2270	2300
36	**PB2**/FPDAT2	1350	-2300	86	**P00**/SIN0/FSIN0	2050	2300
37	**PB1**/FPDAT1	1450	-2300	87	**USBDP**	1950	2300
38	**PB0**/FPDAT0	1550	-2300	88	**USBDM**	1850	2300
39	**PC3**/DRDY	1650	-2300	89	N.C.	1750	2300
40	**PC2**/FPSHIFT	1750	-2300	90	**USBVBUS**	1650	2300
41	**PC1**/FPLINE	1850	-2300	91	**VDDE**	1550	2300
42	**PC0**/FPFRAME	1950	-2300	92	**P31**/#BUSGET/#GARD/SDO	1450	2300
43	**VSS**	2050	-2300	93	**P32**/#DMAACK0/#SRDY3/SPICLK	1350	2300
44	**P16**/EXCL5/#DMAEND1/SOUT3	2270	-2300	94	**VSS**	1250	2300
45	**P15**/EXCL4/#DMAEND0/#SCLK3	2500	-2070	95	**P33**/#DMAACK1/SIN3/SDI	1150	2300
46	**DSIO**	2500	-1850	96	**P34**/#BUSREQ/#CE6/#SMWE	1050	2300
47	**VDDE**	2500	-1750	97	**P35**/#BUSACK/#SMRE	950	2300
48	**DCLK**/P14/FOSC1	2500	-1650	98	**VDD**	850	2300
49	**DPCO**/P13/EXCL3/T8UF3	2500	-1550	99	**#X2SPD**	750	2300
50	**DST2**/P12/EXCL2/T8UF2	2500	-1450	100	**EA10MD0**	650	2300

No.	Pad name	X	Y	No.	Pad name	X	Y
101	**EA10MD1**	550	2300	136	**A16**	-2500	1250
102	**VDDE**	450	2300	137	**VDDE**	-2500	1150
103	**PLLC**	350	2300	138	**A15**	-2500	1050
104	**VSS**	250	2300	139	**A14**	-2500	950
105	**PLLS0**	150	2300	140	**A13**	-2500	850
106	**TST**	50	2300	141	**A12**	-2500	750
107	**BOOT**	-50	2300	142	**VSS**	-2500	650
108	**#CE4**/#CE11/#CE11&12/P50	-150	2300	143	**A11**	-2500	550
109	**VSS**	-250	2300	144	**A10**	-2500	450
110	**#CE5**/#CE15/#CE15&16/P51	-350	2300	145	**VDD**	-2500	350
111	**#CE6**/#CE7&8/P52	-450	2300	146	**A9**	-2500	250
112	#**CE7**/#RAS0/#CE13/#RAS2/P53/#SDCE	-550	2300	147	**A8**	-2500	150
113	**#CE8**/#RAS1/#CE14/#RAS3/P54	-650	2300	148	**A7**	-2500	50
114	**#CE9**/#CE17/#CE17&18/P55	-750	2300	149	**A6**	-2500	-50
115	**#CE10EX**/#CE9&10EX	-850	2300	150	**A5**	-2500	-150
116	**P61**/SDA10	-950	2300	151	**VSS**	-2500	-250
117	**P62**/LDQM	-1050	2300	152	**A4**	-2500	-350
118	**P63**/UDQM	-1150	2300	153	**A3**	-2500	-450
119	**P21**/#DWE/#GAAS/#SDWE	-1250	2300	154	**VDDE**	-2500	-550
120	**#LCAS**/PA0/#SDCAS	-1350	2300	155	**A2**	-2500	-650
121	**VDD**	-1450	2300	156	**A1**	-2500	-750
122	**#HCAS**/PA1/#SDRAS	-1550	2300	157	**A0**/#BSL	-2500	-850
123	**P20**/#DRD/SDCKE	-1650	2300	158	**#WRH**/#BSH	-2500	-950
124	**VDDE**	-1750	2300	159	**#WRL**/#WR/#WE	-2500	-1050
125	**BCLK**/P60/FOSC1/SDCLK	-1850	2300	160	**#RD**	-2500	-1150
126	**A25**/P40	-1950	2300	161	**VSS**	-2500	-1250
127	**VSS**	-2050	2300	162	**D15**	-2500	-1350
128	**A24**/P41	-2270	2300	163	**D14**	-2500	-1450
129	**A23**/P42	-2500	2070	164	**D13**	-2500	-1550
130	**A22**/P43	-2500	1850	165	**D12**	-2500	-1650
131	**A21**/P44	-2500	1750	166	**D11**	-2500	-1750
132	**A20**/P45	-2500	1650	167	**VDD**	-2500	-1850
133	**A19**/P46	-2500	1550	168	**D10**	-2500	-2070
134	**A18**/P47	-2500	1450	169	(Monitor pad for testing)	2500	2266
135	**A17**	-2500	1350	170	(Monitor pad for testing)	-2500	-2266

THIS PAGE IS BLANK.

I-9 Basic External Wiring Diagram



X'tal1	Crystal resonator	32.768 kHz
C_{G1}	Gate capacitor	12 pF
C_{D1}	Drain capacitor	12 pF
R_{f1}	Feedback resistor	10 MΩ
R_{d1}	Drain resistor	0 Ω
R_1	Resistor	4.7 kΩ
C_1	Capacitor	100 pF
C_2	Capacitor	3 pF
R_2	Resistor	10 Ω
C_3	Capacitor	1 µF

X'tal2 or CE	Resonator	Crystal	Ceramic (CSTCW48M0X11∗∗∗)
C_{G2}	Gate capacitor	3 pF	(6 pF) ∗1, ∗2
C_{D2}	Drain capacitor	4 pF	(6 pF) ∗1, ∗2
R_{f2}	Feedback resistor	1 MΩ	22 kΩ ∗1
R_{d2}	Drain resistor	0 Ω	47 Ω ∗1

Note: ∗1 Oscillation characteristics vary depending on conditions (components used, board pattern, etc.). The values in the above table are shown only for reference and not guaranteed. In particular, ceramic oscillation is extremely sensitive to influence of external components and printed-circuit boards. Before using a ceramic resonator, please be sure to contact Murata Manufacturing Co., Ltd. for further information on conditions of use for ceramic resonators. Furthermore, this chip supports only 48-MHz ceramic resonators. Do not use ceramic resonators with any other frequency.

∗2 Capacitance built into the ceramic resonator

THIS PAGE IS BLANK.

I-10 Precautions on Mounting

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

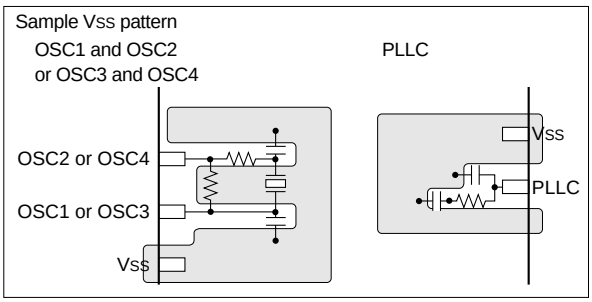
Oscillation Circuit

- 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 V_{SS}.
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) 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.

(5) 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

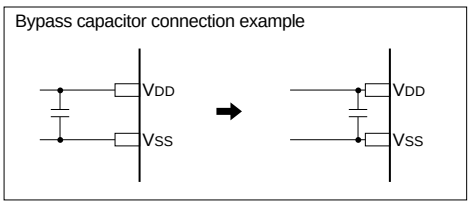
- The power-on reset signal which is input to the #RESET pin changes depending on conditions (power rise time, components used, board pattern, etc.). Decide the time constant of the capacitor and resistor after enough tests have been completed with the application product.
- In order to prevent any occurrences 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.

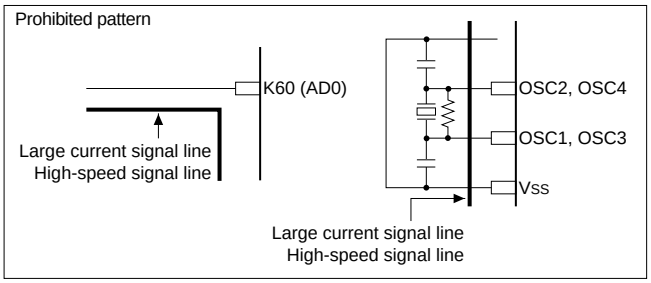


A/D Converter

- 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 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.

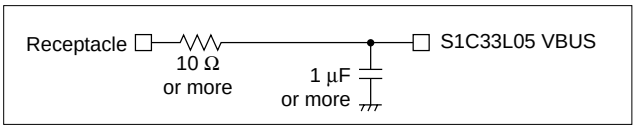


USB

The I/O block of the USB Function Controller incorporated in this chip has the following features:
The DP and DM pins can be connected directly to the USB connector.
The VBUS level is detected by means of a 2/3 resistive division internally in the chip, thus allowing for direct input of a 5 V-level signal.
The receiver does not enter a floating state even when the USB cable is disconnected from the USB connector. When the USB cable is disconnected, the VBUS pin is tied to VSS, so that leakage current will be the only source that drains power in the USB I/O block. (This applies only to the USBVDD power supply current, and does not include the VDD power supply current for internal logic.)

Precautions on VBUS

Be sure to not apply 6 V (max.) or more to the VBUS pin as the IC may be destroyed.
It is especially necessary to suppress overshoot on the input voltage and to prevent the host power source becoming unstable when the USB cable is plugged into the connector. To do this, connect a 1 µF or more capacitor near the USB connector for decoupling the VBUS signal. Choose a ceramic capacitor for decoupling.



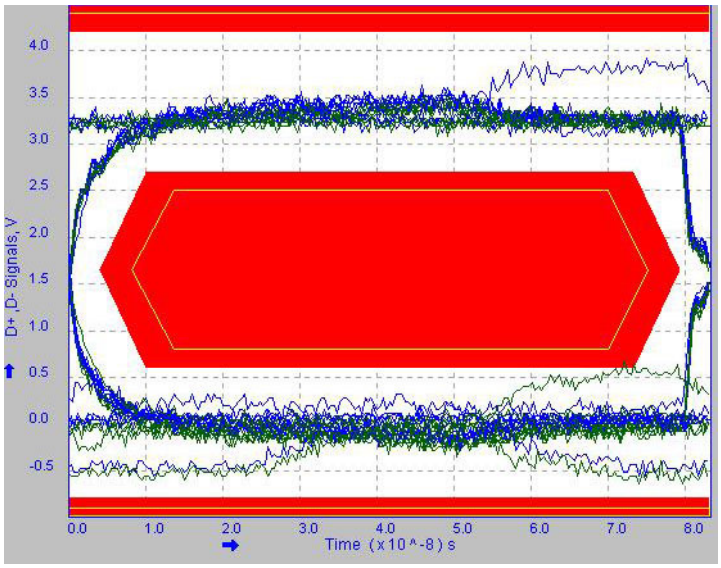
In addition to the above, verify the VBUS state completely on the actual circuit board using an oscilloscope or other device. Overshoot and other symptoms are more likely to occur when using a long USB cable and connecting it to the host side connector.

Precautions on DP and DM

When designing a printed circuit board, observe the following precautions to ensure that both DP and DM signals are properly routed:

- To prevent signal skew and to stabilize differential impedance, the DP and DM signal lines must be routed in parallel and in the same length, with the pins and connector connected in the shortest distance possible. Crossed wiring of these signals should be avoided as much as possible.
- The periphery of these signal lines must be enclosed by a GND pattern, and with the GND pattern also created for the internal layer immediately below that. In particular, the routing of high-speed digital signal lines parallel to or across these signal lines should be avoided as much as possible.

We recommend that you verify the EYE pattern on the actual machine. The diagram below shows an example measurement performed by the DMT33L05 that incorporates the S1C33L05.



Sample EYE Diagram

Noise-Induced Erratic Operations

If erratic IC operations appear to be attributable to noise, consider the following five points.

(1) TST pin (ICEMD pin in earlier products)
If this pin is exposed to high-level noise, the entire IC enters test mode or a high-impedance state and becomes inoperable. In such cases, the IC will not be restored, even when the pin is returned to a low level. Therefore, always make sure the TST pin is connected to GND on the circuit board. Although the IC contains internal pulldown resistors, it is susceptible to noise because these resistors are high impedance (approximately 50 to 100 kΩ).

(2) DSIO pin
Exposure of this pin to low-level noise causes the IC to enter debug mode. In debug mode, the clock is output from the DCLK pin and the DST2 pin is high, indicating that the IC is in debug mode.
In product versions, it is recommended that the DSIO pin be pulled high by connecting it directly to VDD or through a resistor of 10 kΩ or less.
Although the IC contains internal pull-up resistor, it is susceptible to noise because these resistors are high impedance (approximately 50 to 100 kΩ).
For details, refer to the "S1C33 Family Application Note for Standard Core".

(3) #RESET pin
Low-level noise on this pin resets the IC. However, the IC may not always be reset normally, depending on the input waveform.
Due to circuit design, this situation tends to occur when the reset input is in the high state, with high impedance. For details, refer to the "S1C33 Family Application Note for Standard Core".

(4) #NMI pin
Low-level noise on this pin causes an NMI interrupt. Due to the circuit design, this situation tends to occur when the #NMI pin is in the high state, with high impedance. Lower the impedance of #NMI when it is held high, or incorporate corrective measures into the software to protect against erratic operations.

(5) VDD, VSS, and VDDE power supplies
If noise lower than the rated voltage enters one of these power-supply lines, the IC may operate erratically. Take corrective measures in board design; for example, by using solid patterns for power supply lines, adding decoupling capacitors to eliminate noise, or incorporating surge/noise counteracting devices into the power supply lines.

To confirm the above, use an oscilloscope capable of observing higher-frequency waveforms of 200 MHz. The generation of fast noise may not be observed with a low-frequency oscilloscope.

If potential noise-induced erratic operations are detected through waveform observations using an oscilloscope, connect the suspected pin to the GND or power supply with low impedance (1 kΩ or less) and check once again. If erratic operations are no longer detected or occur at reduced frequency, or if different symptoms of erratic operations are observed, said pin may with reasonably certainty be considered to be the source of the erratic operations.

The TST (ICEMD), DSIO, #RESET, and #NMI input circuits described above are designed to detect the edges of the input signal, so that even spike noise may result in erratic operations. Among the digital signal circuits, these pins are most susceptible to noise.
In the design of the circuit board, take the following two points into consideration to protect the signal from noise.

(A) The most important measure is to lower the signal-driving impedance, as described in each item above. Connect pins to the power supply or GND, with impedance of 1 kΩ or less, preferably 0 Ω. In addition, limit the length of the connected signal lines to approximately 5 cm.

(B) Parallel routing of said signal lines with other digital lines on the board is undesirable, since the noise generated when the signal changes from high to low or vice versa may adversely affect signals. The signal may be subject to the most noise when signal lines are laid between multiple signal lines whose states change simultaneously. Take corrective measures by shortening the parallel distance (to several cm) or separating signal lines (2 mm or more).

Reference

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

Other

The 0.18 µm 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 µm 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.

S1C33L05 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.

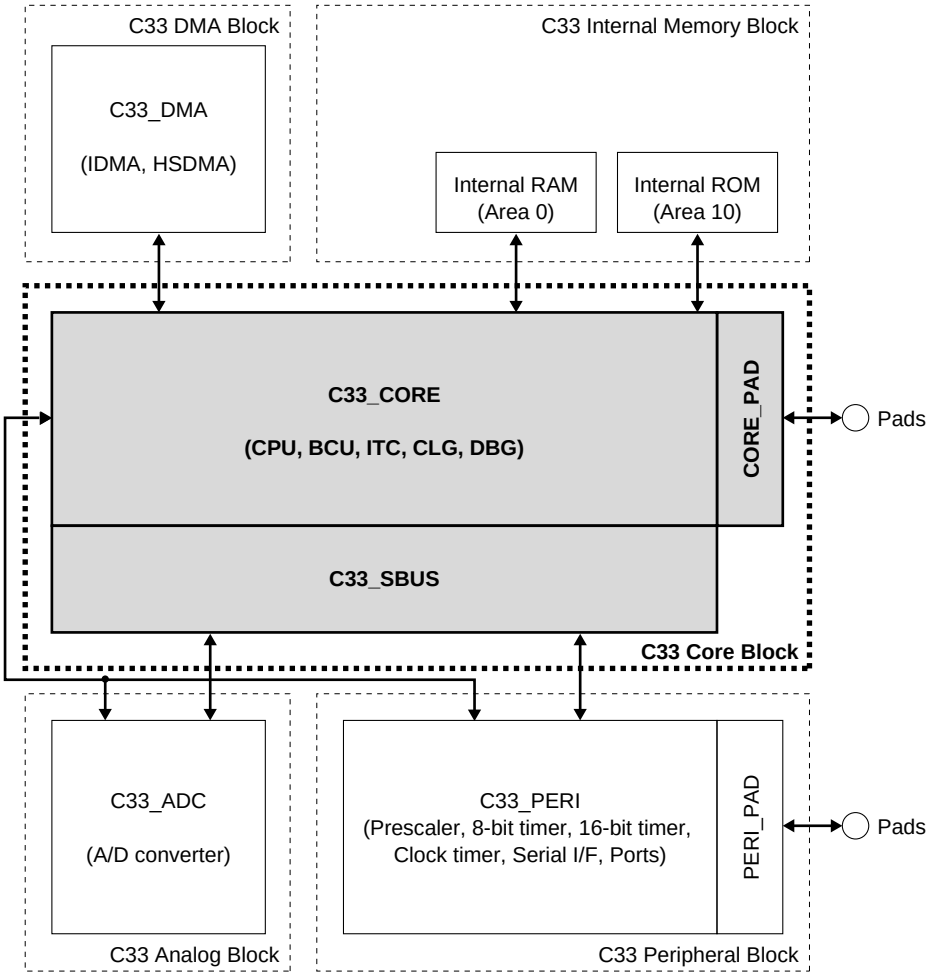


Figure II.1.1 Core Block

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.

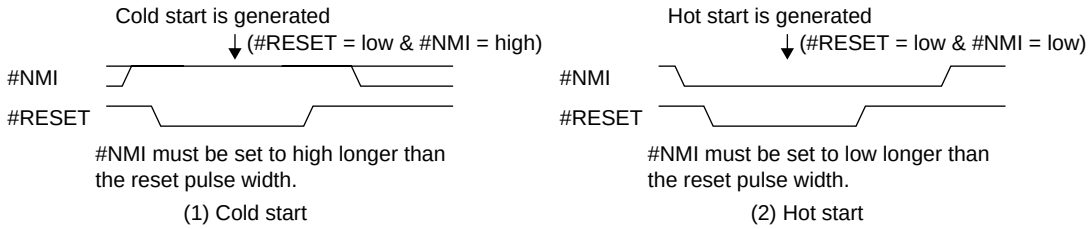


Figure II.3.1 Setup of #RESET and #NMI Pins

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.

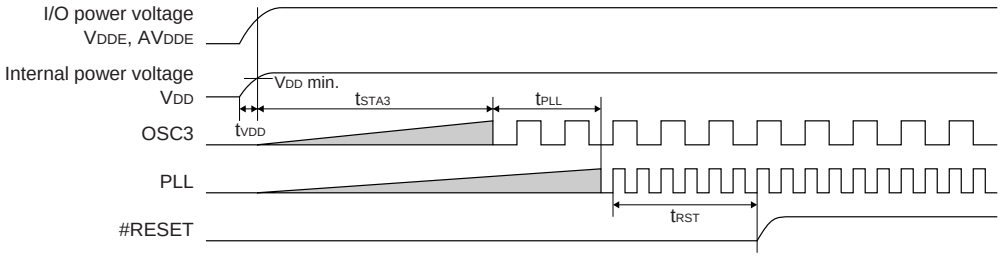


Figure II.3.2 Power-On Sequence

(1) t_{VDD}: Elapsed time until the power supply stabilizes after power-on
Supply power in the following sequence (or simultaneously).
Power-on: Internal core power voltage (V_{DD} level voltages) → I/O power voltage (V_{DDE}, AV_{DDE}, etc.) → 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.

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) *
#CEx	#CEx

* 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 9 SRAM type Burst ROM type 8 or 16 bits	0x0BFFFFF 0x0800000	External memory (4MB)
Area 8 SRAM type 8 or 16 bits	0x07FFFFF 0x0600000	External memory (2MB)
Area 7 SRAM type 8 or 16 bits	0x05FFFFF 0x0400000	External memory (2MB)
Area 6 SRAM type	0x03FFFFF 0x0380000 0x037FFFF 0x0300000	External I/O (16-bit device) External I/O (8-bit device)
Area 5 SRAM type 8 or 16 bits	0x02FFFFF 0x0200000	External memory (1MB)
Area 4 SRAM type 8 or 16 bits	0x01FFFFF 0x0100000	External memory (1MB)
Area 3 16 bits Fixed at 1 cycle	0x00FFFFF 0x0080000	(Reserved)
Area 2 16 bits Fixed at 3 cycles	0x007FFFF 0x0060000	(Reserved) For CPU core or debug mode
Area 1 8, 16 bits 2 or 4 cycles	0x005FFFF 0x0050000 0x004FFFF 0x0040000 0x003FFFF 0x0030000	(Mirror of internal I/O) Internal I/O (Mirror of internal I/O)
Area 0 32 bits Fixed at 1 cycle	0x002FFFF 0x0000000	Internal RAM

Area	Address	
Area 18 SRAM type 8 or 16 bits	0xFFFFFFF 0xC000000	External memory (64MB)
Area 17 SRAM type 8 or 16 bits	0xBFFFFFF 0x8000000	External memory (64MB)
Area 16 SRAM type 8 or 16 bits	0x7FFFFFF 0x6000000	External memory (32MB)
Area 15 SRAM type 8 or 16 bits	0x5FFFFFF 0x4000000	External memory (32MB)
Area 14 SRAM type 8 or 16 bits	0x3FFFFFF 0x3000000	External memory (16MB)
Area 13 SRAM type 8 or 16 bits	0x2FFFFFF 0x2000000	External memory (16MB)
Area 12 SRAM type 8 or 16 bits	0x1FFFFFF 0x1800000	External memory (8MB)
Area 11 SRAM type 8 or 16 bits	0x17FFFFF 0x1000000	External memory (8MB)
Area 10 SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFF 0x0C00000	External memory (4MB)

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 10 (#CE10) SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFF 0x0C00000	External memory 6 (4MB)
Area 9 (#CE9) SRAM type Burst ROM type 8 or 16 bits	0x0BFFFFF 0x0800000	External memory 5 (4MB)
Area 8 (#CE8/#RAS1) SRAM type 8 or 16 bits	0x07FFFFF 0x0600000	External memory 4 (2MB)
Area 7 (#CE7/#RAS0) SRAM type 8 or 16 bits	0x05FFFFF 0x0400000	External memory 3 (2MB)
Area 6 (#CE6) SRAM type	0x03FFFFF 0x0380000 0x037FFFF 0x0300000	External I/O (16-bit device) External I/O (8-bit device)
Area 5 (#CE5) SRAM type 8 or 16 bits	0x02FFFFF 0x0200000	External memory 2 (1MB)
Area 4 (#CE4) SRAM type 8 or 16 bits	0x01FFFFF 0x0100000	External memory 1 (1MB)

CEFUNC = "00"

Area	Address	
Area 17 (#CE17) SRAM type 8 or 16 bits	0xBFFFFFF 0x8000000	External memory 6 (64MB)
Area 15 (#CE15) SRAM type 8 or 16 bits	0x5FFFFFF 0x4000000	External memory 5 (32MB)
Area 14 (#CE14/#RAS3) SRAM type 8 or 16 bits	0x3FFFFFF 0x3000000	External memory 4 (16MB)
Area 13 (#CE13/#RAS2) SRAM type 8 or 16 bits	0x2FFFFFF 0x2000000	External memory 3 (16MB)
Area 11 (#CE11) SRAM type 8 or 16 bits	0x17FFFFF 0x1000000	External memory 2 (8MB)
Area 10 (#CE10) SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFF 0x0C00000	External memory 1 (4MB)
Area 6 (#CE6) SRAM type	0x03FFFFF 0x0380000 0x037FFFF 0x0300000	External I/O (16-bit device) External I/O (8-bit device)

CEFUNC = "01"

Area	Address	
Area 17–18 (#CE17+18) SRAM type 8 or 16 bits	0xFFFFFFF 0x8000000	External memory 7 (128MB)
Areas 15–16 (#CE15+16) SRAM type 8 or 16 bits	0x7FFFFFF 0x4000000	External memory 6 (64MB)
Area 14 (#CE14/#RAS3) SRAM type 8 or 16 bits	0x3FFFFFF 0x3000000	External memory 5 (16MB)
Area 13 (#CE13/#RAS2) SRAM type 8 or 16 bits	0x2FFFFFF 0x2000000	External memory 4 (16MB)
Areas 11–12 (#CE11+12) SRAM type 8 or 16 bits	0x1FFFFFF 0x1000000	External memory 3 (16MB)
Areas 9–10 (#CE9+10EX) SRAM type Burst ROM type 8 or 16 bits	0x0FFFFFF 0x0800000	External memory 2 (8MB)
Areas 7–8 (#CE7+8) SRAM type 8 or 16 bits	0x07FFFFF 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

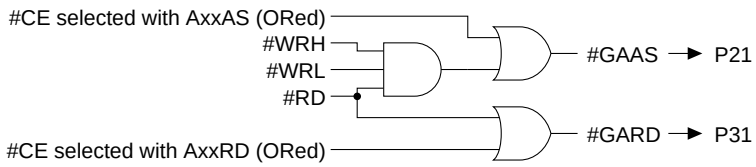


Figure II.4.3 #GAAS and #GARD Signals

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.

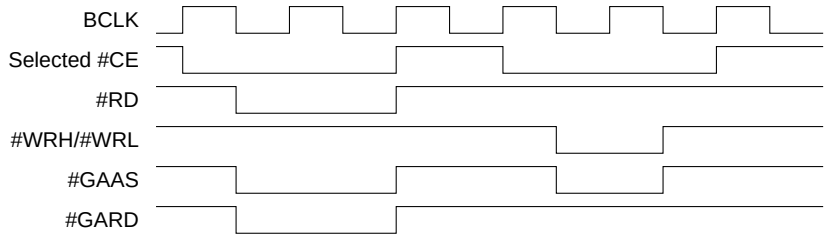


Figure II.4.4 Output Timing of #GAAS and #GARD Signals

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 a boot mode other than external 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.

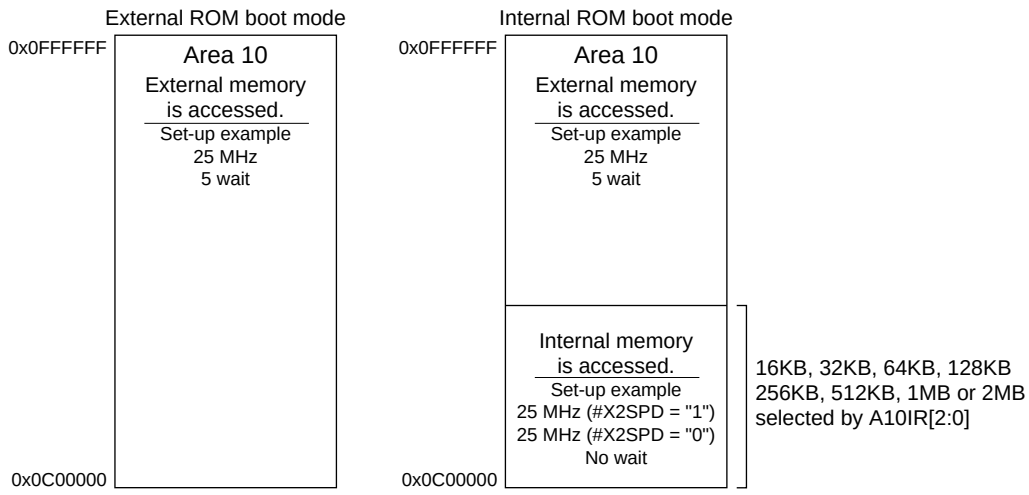


Figure II.4.5 Area 10 Memory Map

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 (0x380000–0x3FFFFF) 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 #CEx 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 #CEx 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.

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

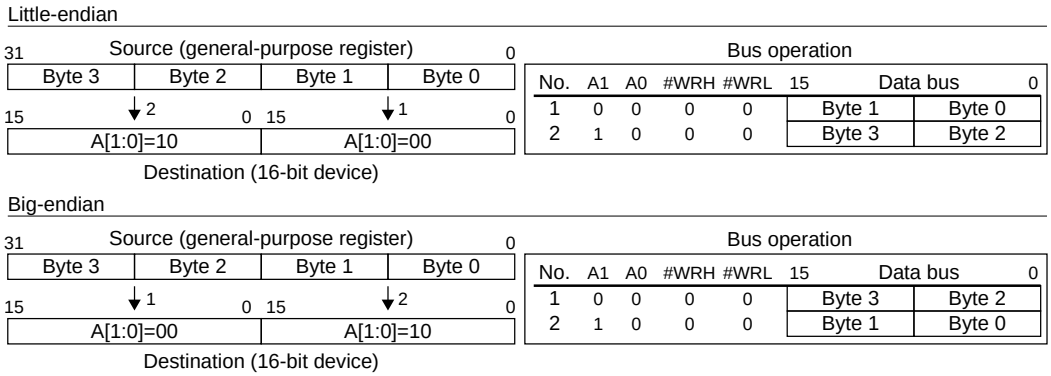


Figure II.4.6 Word Data Writing to a 16-bit Device

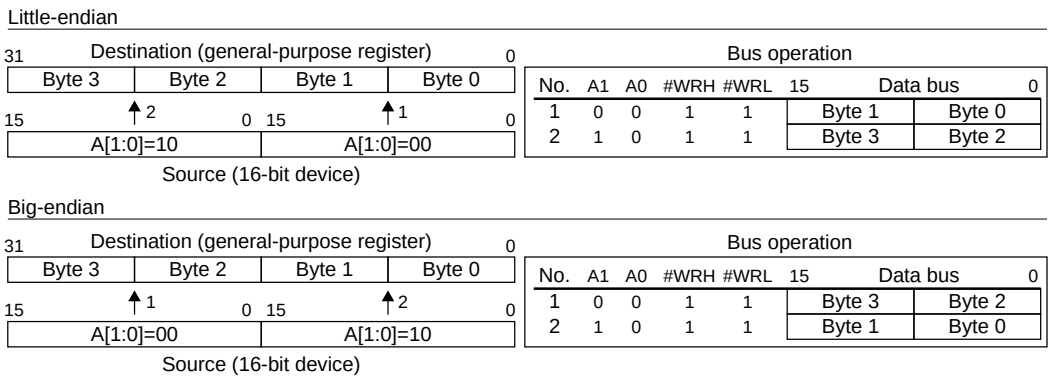


Figure II.4.7 Word Data Reading from a 16-bit Device

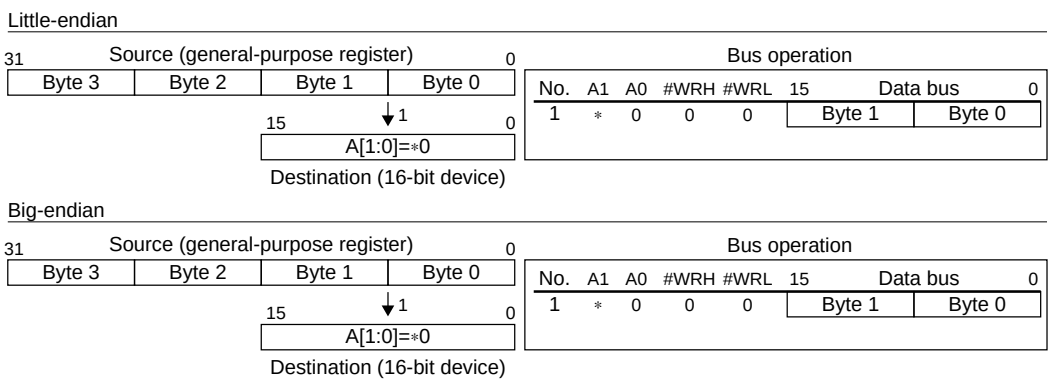


Figure II.4.8 Half-word Data Writing to a 16-bit Device

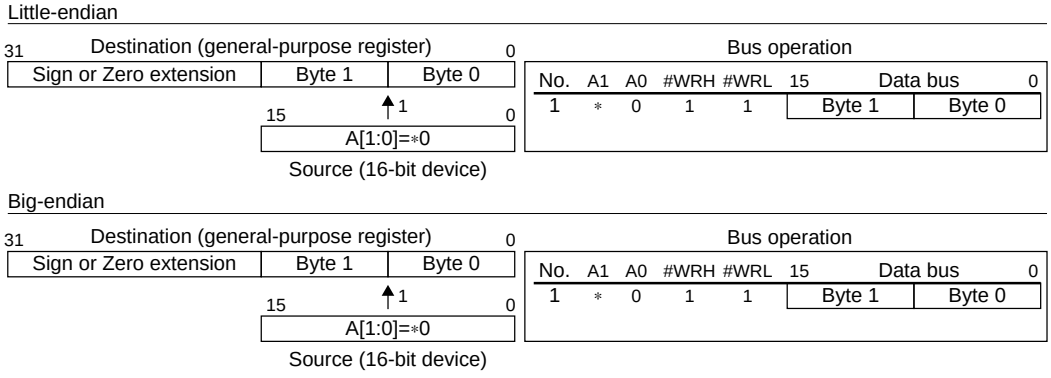
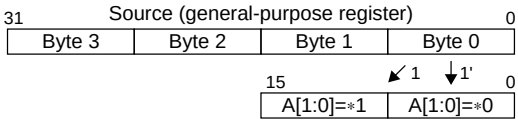


Figure II.4.9 Half-word Data Reading from a 16-bit Device

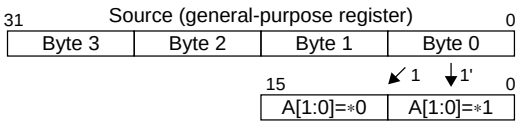
Little-endian



Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus 0	
1	∗	1	0	1	Byte 0	Data retained
1'	∗	0	1	0	Data retained	Byte 0

Big-endian

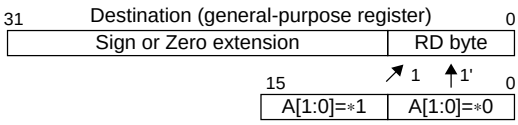


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus 0	
1	∗	0	0	1	Byte 0	Data retained
1'	∗	1	1	0	Data retained	Byte 0

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

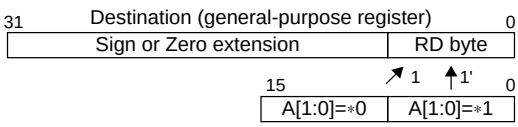
Little-endian



Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus 0	
1	∗	1	1	1	RD byte	Ignored
1'	∗	0	1	1	Ignored	RD byte

Big-endian

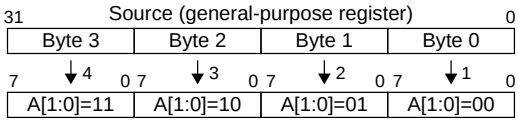


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus 0	
1	∗	0	1	1	RD byte	Ignored
1'	∗	1	1	1	Ignored	RD byte

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

Little-endian

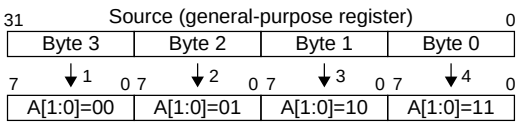


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus 0	
1	0	0	X	0	Data retained	Byte 0
2	0	1	X	0	Data retained	Byte 1
3	1	0	X	0	Data retained	Byte 2
4	1	1	X	0	Data retained	Byte 3

(X: Not connected/Unused)

Big-endian

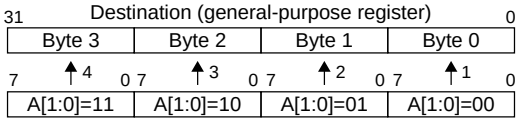


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus 0	
1	0	0	0	1	Byte 3	Data retained
2	0	1	0	1	Byte 2	Data retained
3	1	0	0	1	Byte 1	Data retained
4	1	1	0	1	Byte 0	Data retained

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

Little-endian

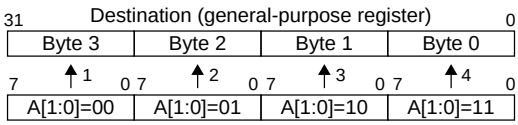


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus 0	
1	0	0	X	1	Ignored	Byte 0
2	0	1	X	1	Ignored	Byte 1
3	1	0	X	1	Ignored	Byte 2
4	1	1	X	1	Ignored	Byte 3

(X: Not connected/Unused)

Big-endian

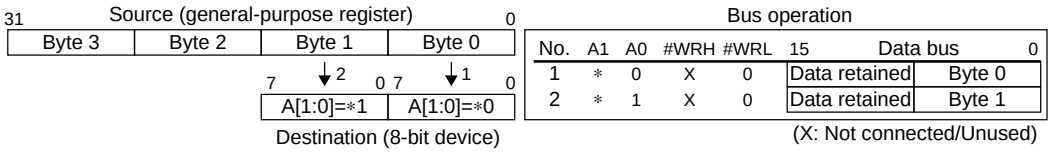


Bus operation

No.	A1	A0	#WRH	#WRL	15 Data bus 0	
1	0	0	1	1	Byte 3	Ignored
2	0	1	1	1	Byte 2	Ignored
3	1	0	1	1	Byte 1	Ignored
4	1	1	1	1	Byte 0	Ignored

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

Little-endian



Big-endian

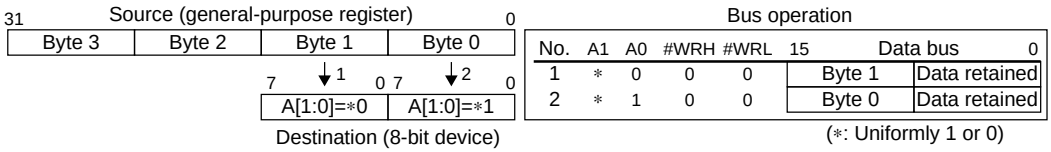
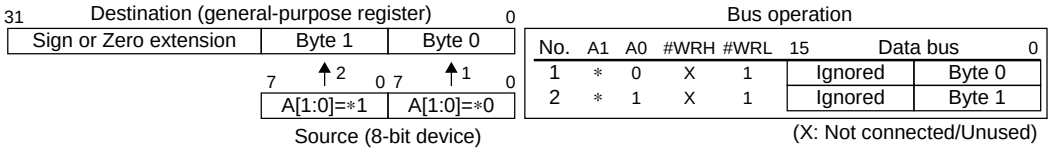


Figure II.4.14 Half-word Data Writing to an 8-bit Device

Little-endian



Big-endian

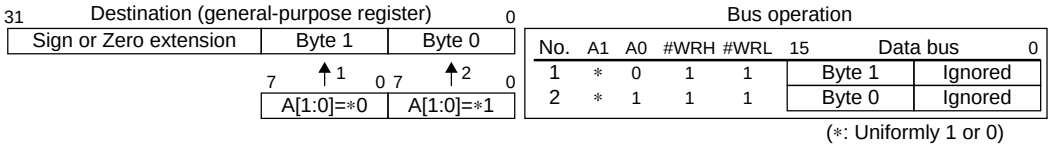
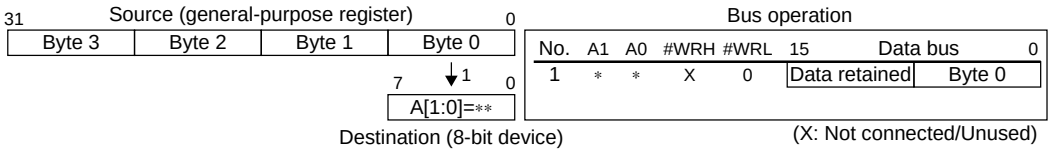


Figure II.4.15 Half-word Data Reading from an 8-bit Device

Little-endian



Big-endian

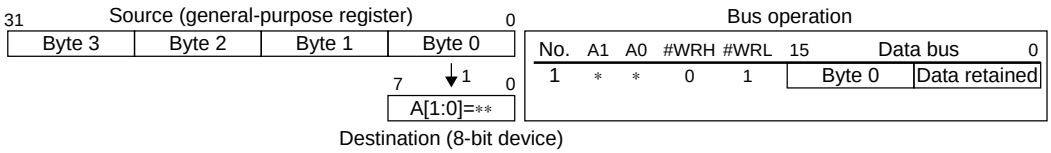
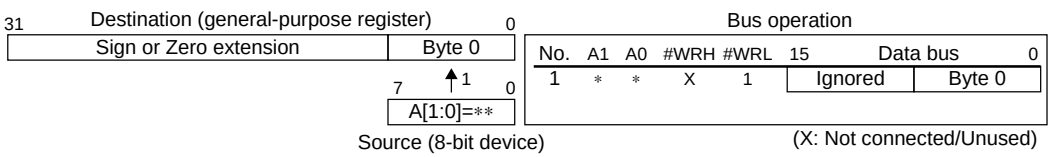


Figure II.4.16 Byte Data Writing to an 8-bit Device

Little-endian



Big-endian

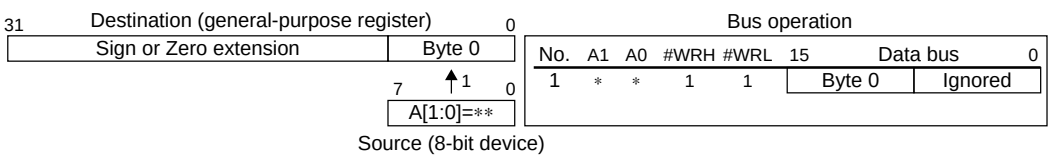


Figure II.4.17 Byte Data Reading from an 8-bit Device

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.

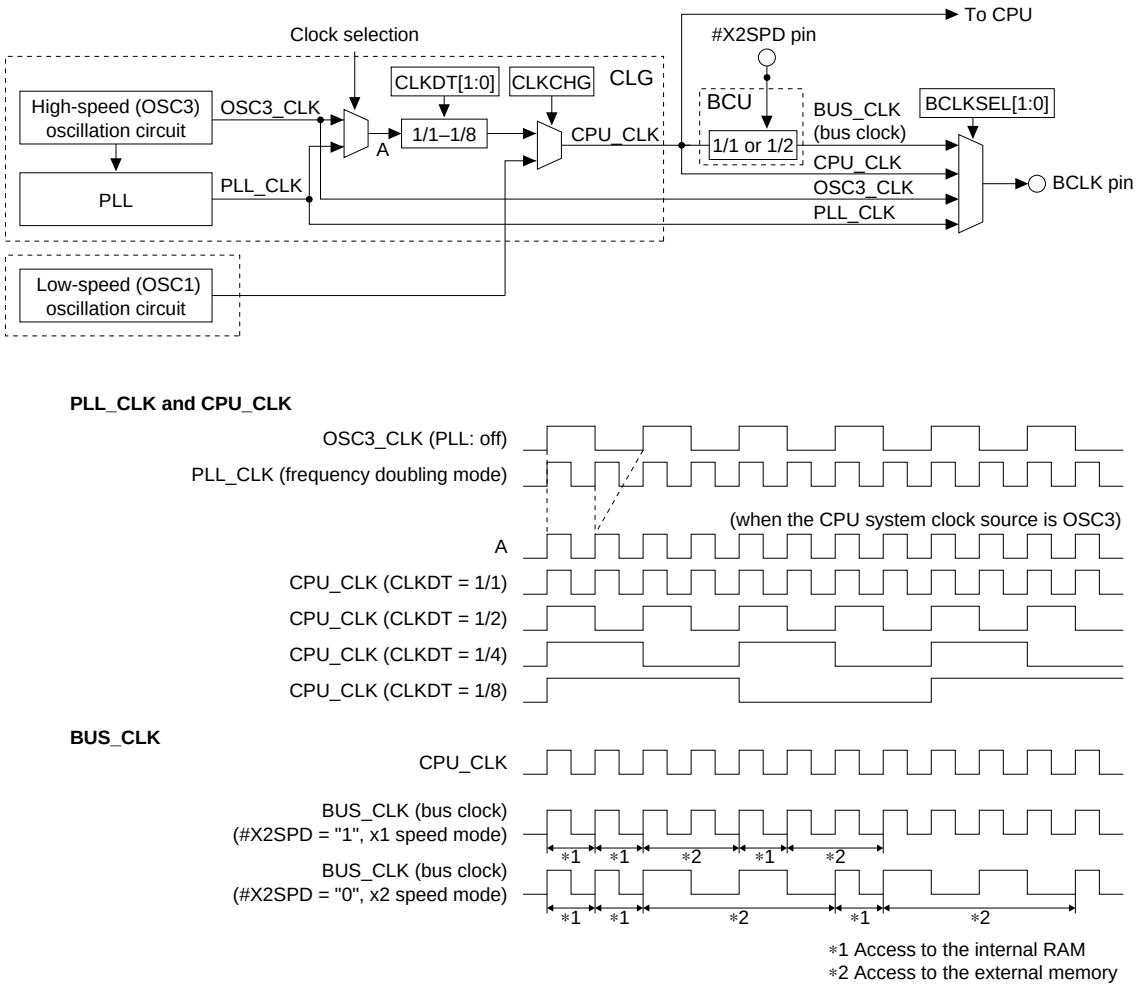


Figure II.4.18 Clock System

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

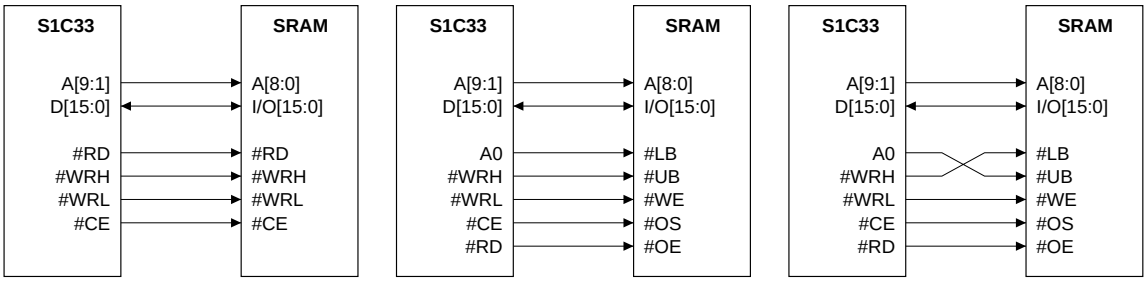
The bus clock is also output from the BCLK pin to an external device. The BCLK output clock can be selected from among four types using BCLKSEL[1:0] (D[1:0]) / BCLK select register (0x4813A).

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.



(1) A0 system (little endian/big endian) (2) #BSL system (little endian) (3) #BSL system (big endian)

Figure II.4.19 Sample DRAM Connection

SRAM Read Cycles

Basic read cycle with no wait specified

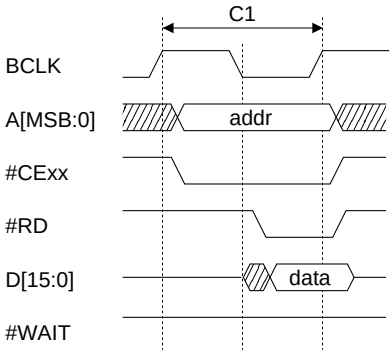


Figure II.4.20 Read Cycle with No Wait Specified

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

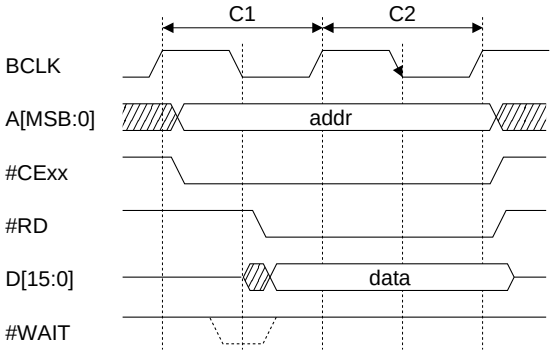


Figure II.4.21 Read Cycle with Wait (specified using AxxWT[2:0])

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].

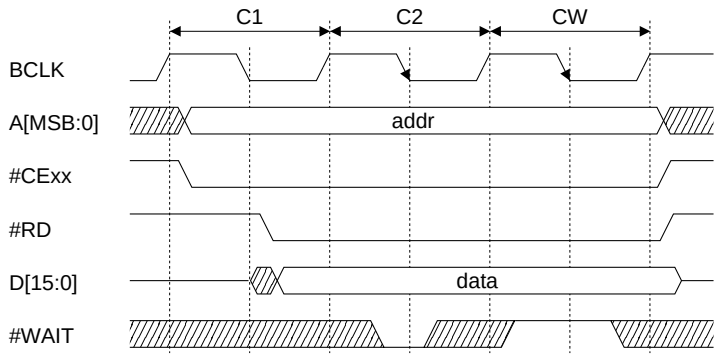


Figure II.4.22 Read Cycle with #WAIT Control

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.

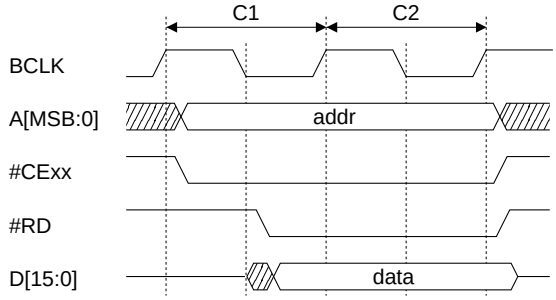
In both Figures II.4.21 and II.4.22, the #WAIT status sampled at the falling edge of BCLK during C2 cycle is effective because 1-wait cycle (C1) is inserted using AxxWT[2:0]. The #WAIT status at the falling edge of BCLK during C1 cycle does not affect wait cycle insertion.
In the case of Figure II.4.21, 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 (#CExx) 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 #CExx and A[MSB:0] signals after the #RD signal rises by setting the AxxRH 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 (AxxRH = "0", number of wait cycles AxxWT[2:0] = "0x1")



When read-hold cycle is inserted (AxxRH = "1", number of wait cycles AxxWT[2:0] = "0x1")

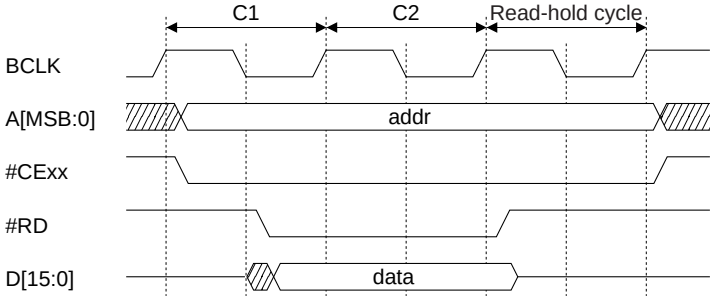
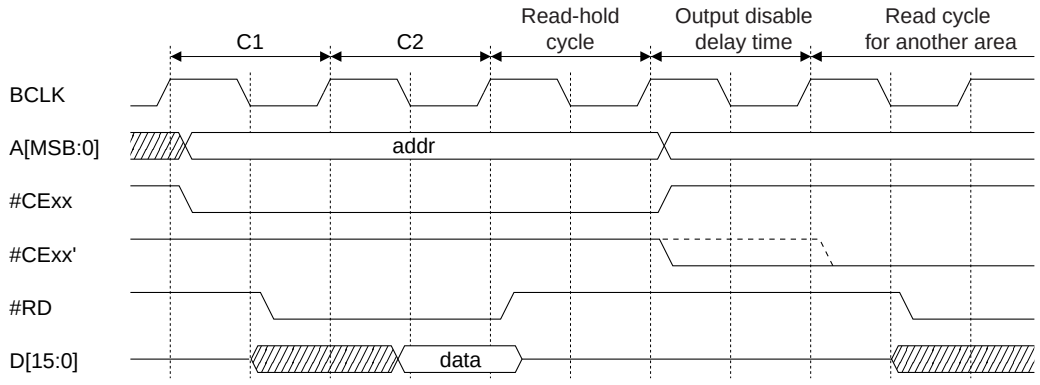


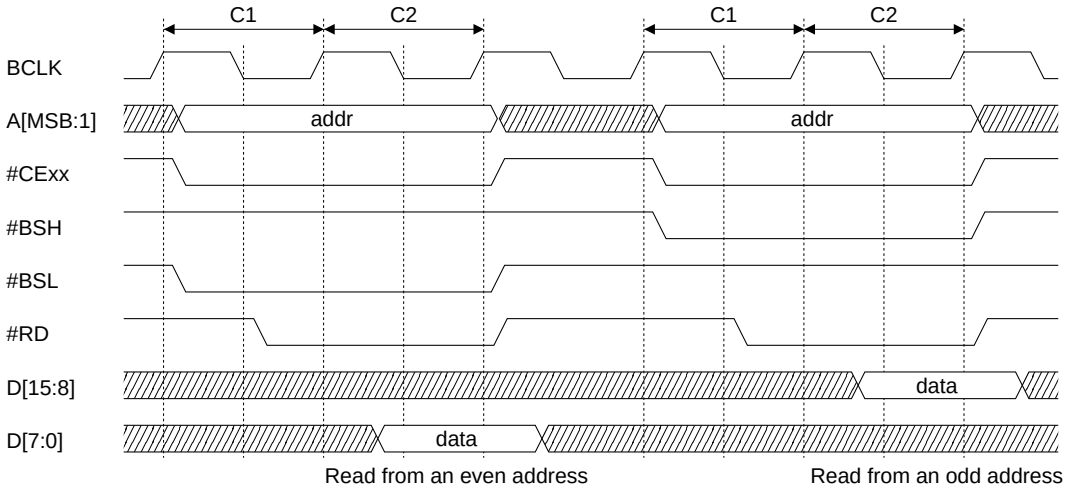
Figure II.4.23 Inserting a Read-Hold Cycle



(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)

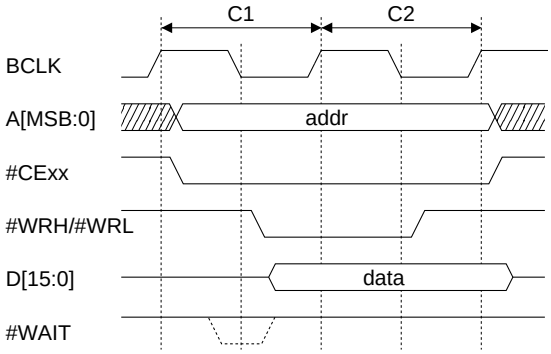


Figure II.4.26 Basic Write Cycle without a Wait Specification (A0 method)

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

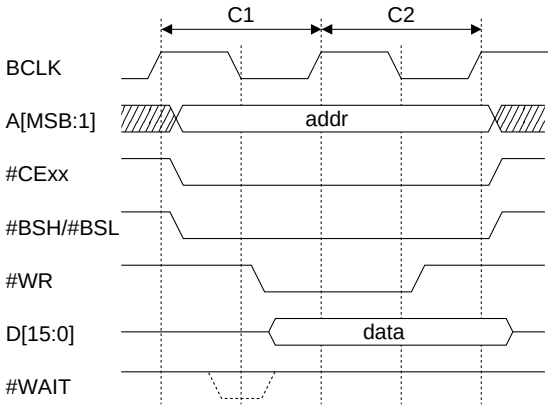


Figure II.4.27 Basic Write Cycle without a Wait Specification (BSL method)

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

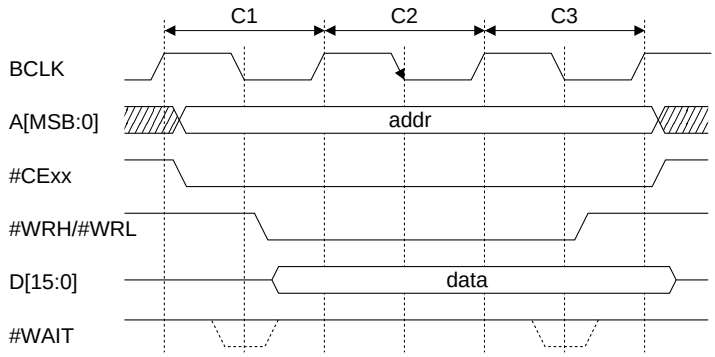


Figure II.4.28 Write Cycle in which 1 Wait Cycle is Added (specified using AxxWT[2:0])

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].

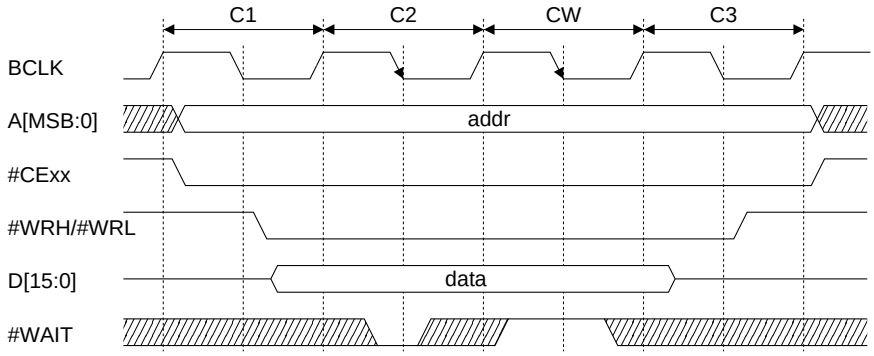


Figure II.4.29 Write Cycle with #WAIT Control

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.

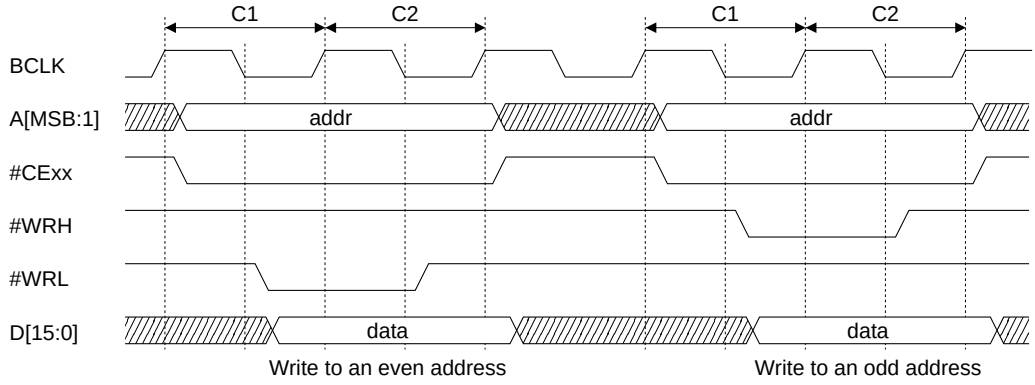
In both Figures II.4.28 and II.4.29, the #WAIT status sampled at the falling edge of BCLK during C2 cycle is effective because 2 wait cycles (C2 and C3) are inserted using AxxWT[2:0]. The #WAIT status at the falling edge of BCLK during C1 and C3 cycles do not affect wait cycle insertion.

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

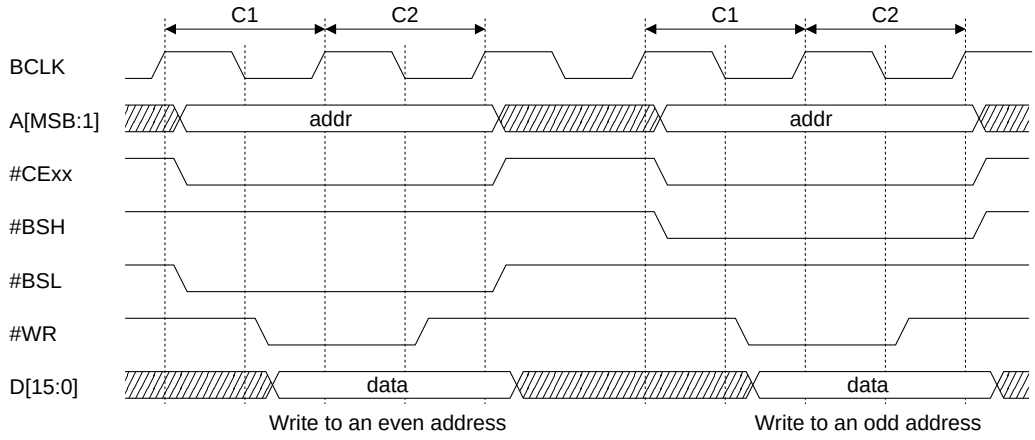
1. A0 method (little endian)



(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

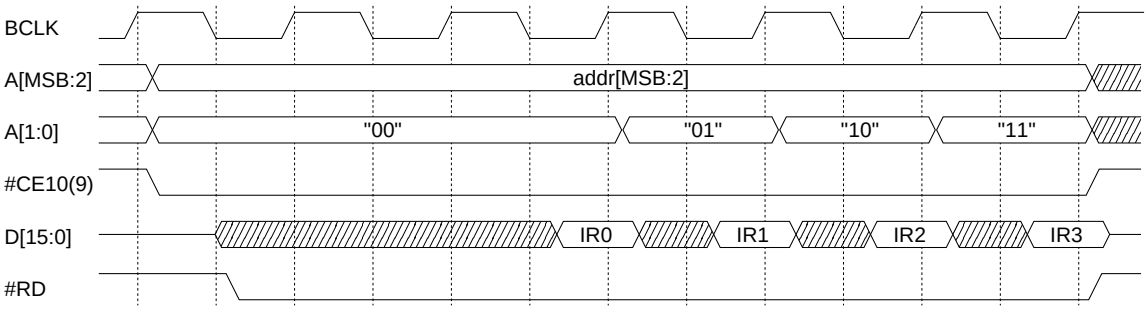


Figure II.4.32 Burst Read Cycle

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, #CExx
   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.

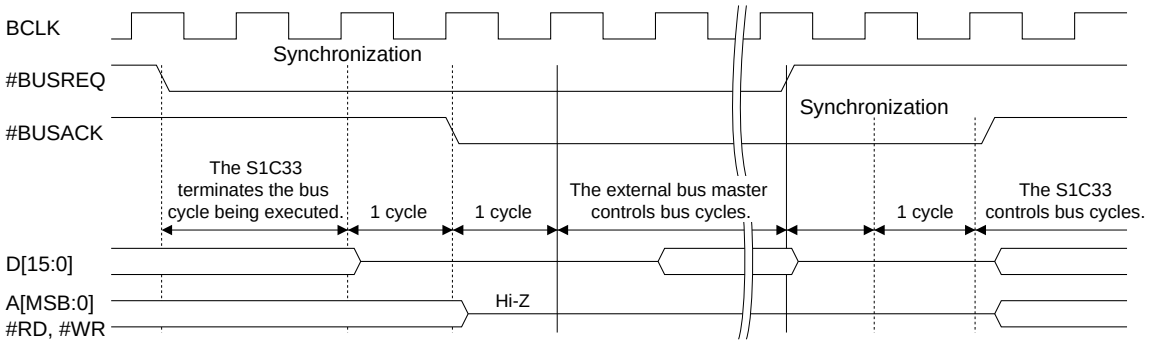


Figure II.4.33 External Bus Release Timing

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]: 11 = 3.5; 10 = 2.5; 01 = 1.5; 00 = 0.5	1	R/W	
		DC	**A18DF0**			1		
		DB	**–**	reserved	–	–	–	0 when being read.
		DA	**A18WT2**	Areas 18–17 wait control	A18WT[2:0]: 111 = 7; 110 = 6; 101 = 5; 100 = 4; 011 = 3; 010 = 2; 001 = 1; 000 = 0	1	R/W	
		D9	**A18WT1**			1		
		D8	**A18WT0**			1		
		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]: 11 = 3.5; 10 = 2.5; 01 = 1.5; 00 = 0.5	1	R/W	
		D4	**A16DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A16WT2**	Areas 16–15 wait control	A16WT[2:0]: 111 = 7; 110 = 6; 101 = 5; 100 = 4; 011 = 3; 010 = 2; 001 = 1; 000 = 0	1	R/W	
		D1	**A16WT1**			1		
		D0	**A16WT0**			1		
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]: 11 = 3.5; 10 = 2.5; 01 = 1.5; 00 = 0.5	1	R/W	
		D4	**A14DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A14WT2**	Areas 14–13 wait control	A14WT[2:0]: 111 = 7; 110 = 6; 101 = 5; 100 = 4; 011 = 3; 010 = 2; 001 = 1; 000 = 0	1	R/W	
		D1	**A14WT1**			1		
		D0	**A14WT0**			1		

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]: 11 = 3.5; 10 = 2.5; 01 = 1.5; 00 = 0.5	1	R/W	
		D4	**A12DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A12WT2**	Areas 12–11 wait control	A12WT[2:0]: 111 = 7; 110 = 6; 101 = 5; 100 = 4; 011 = 3; 010 = 2; 001 = 1; 000 = 0	1	R/W	
		D1	**A12WT1**			1		
		D0	**A12WT0**			1		
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]: 111 = 2MB; 110 = 1MB; 101 = 512KB; 100 = 256KB; 011 = 128KB; 010 = 64KB; 001 = 32KB; 000 = 16KB	1	R/W	
		DD	**A10IR1**			1		
		DC	**A10IR0**			1		
		DB	**–**	reserved	–	–	–	0 when being read.
		DA	**A10BW1**	Areas 10–9 burst ROM burst read cycle wait control	A10BW[1:0]: 11 = 3; 10 = 2; 01 = 1; 00 = 0	0	R/W	
		D9	**A10BW0**			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]: 11 = 3.5; 10 = 2.5; 01 = 1.5; 00 = 0.5	1	R/W	
		D4	**A10DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A10WT2**	Areas 10–9 wait control	A10WT[2:0]: 111 = 7; 110 = 6; 101 = 5; 100 = 4; 011 = 3; 010 = 2; 001 = 1; 000 = 0	1	R/W	
		D1	**A10WT1**			1		
		D0	**A10WT0**			1		
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]: 11 = 3.5; 10 = 2.5; 01 = 1.5; 00 = 0.5	1	R/W	
		D4	**A8DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A8WT2**	Areas 8–7 wait control	A8WT[2:0]: 111 = 7; 110 = 6; 101 = 5; 100 = 4; 011 = 3; 010 = 2; 001 = 1; 000 = 0	1	R/W	
		D1	**A8WT1**			1		
		D0	**A8WT0**			1		

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	1		
		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 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	1		
		D8	**A6WT0**			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 0 0 0.5	1		
		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 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	1		
		D0	**A5WT0**			1		
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	0		
		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	RRA[1:0] Number of cycles	0	R/W	
		D5	**RRA0**	selection	1 1 5 1 0 4 0 1 3 0 0 2	0		
		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 D9	**CEFUNC1** **CEFUNC0**	#CE pin function selection	CEFUNC[1:0]: 1 x → #CE7/8..#CE17/18; 0 1 → #CE6..#CE17; 0 0 → #CE4..#CE10	0 0	R/W	
		D8	**CRAS**	Successive RAS mode setup	1: Successive, 0: Normal	0	R/W	
		D7 D6	**RPRC1** **RPRC0**	DRAM RAS precharge cycles selection	RPRC[1:0]: 1 1 → 4; 1 0 → 3; 0 1 → 2; 0 0 → 1	0 0	R/W	
		D5	**–**	reserved	–	–	–	0 when being read.
		D4 D3	**CASC1** **CASC0**	DRAM CAS cycles selection	CASC[1:0]: 1 1 → 4; 1 0 → 3; 0 1 → 2; 0 0 → 1	0 0	R/W	
		D2	**–**	reserved	–	–	–	0 when being read.
		D1 D0	**RASC1** **RASC0**	DRAM RAS cycles selection	RASC[1:0]: 1 1 → 4; 1 0 → 3; 0 1 → 2; 0 0 → 1	0 0	R/W	
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 D0	**BCLKSEL1** **BCLKSEL0**	BCLK output clock selection	BCLKSEL[1:0]: 1 1 → PLL_CLK; 1 0 → OSC3_CLK; 0 1 → BUS_CLK; 0 0 → CPU_CLK	0 0	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
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					0	R/W	
		D5	**A14RH**	Areas 14–13 read hold cycle					0	R/W	
		D4	**A12RH**	Areas 12–11 read hold cycle					0	R/W	
		D3	**A10RH**	Areas 10–9 read hold cycle					0	R/W	
		D2	**A8RH**	Areas 8–7 read hold cycle					0	R/W	
		D1	**A6RH**	Area 6 read hold cycle					0	R/W	
		D0	**A5RH**	Areas 5–4 read hold cycle					0	R/W	
Bus speed setting register	**004813E (B)**	D7	**A18BS**	Areas 18–17 bus speed selection	1	×1 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					0	R/W	
		D5	**A14BS**	Areas 14–13 bus speed selection					0	R/W	
		D4	**A12BS**	Areas 12–11 bus speed selection					0	R/W	
		D3	**A10BS**	Areas 10– 9 bus speed selection					0	R/W	
		D2	**A8BS**	Areas 8–7 bus speed selection					0	R/W	
		D1	**A6BS**	Area 6 bus speed selection					0	R/W	
		D0	**A5BS**	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 0x3FFFFF) 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.

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 ("1").

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.

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–BCLKSEL0: 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	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)

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.

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".

(3) Table II.4.18 gives the bus condition setup list depending on the bus speed mode.

Table II.4.18 Bus Condition Setup List

	Bus condition	48 MHz @x1 speed	48 MHz @x2 speed				32 MHz @x1 speed				24 MHz @x1 speed			
Bus Arbiter function	–	**Off**	**Off**	**Off**	**On**	**On**	**Off**	**Off**	**On**	**On**	**Off**	**Off**	**On**	**On**
USB function	–	**Off**	**Off**	**On**	**Off**	**On**	**Off**	**On**	**Off**	**On**	**Off**	**On**	**Off**	**On**
IVRAM (Internal Area 6)	Wait-cycle	NA	0	0	0	0	1	1	1	1	1	1	1	1
	Output-disable		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Read-hold		0	1	0	1	0	1	0	1	0	1	0	1
SDRAM initialization (Internal Area 6)	Wait-cycle	NA	0	0	0	0	2	2	2	2	2	2	2	2
	Output-disable		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Read-hold		0	1	0	1	0	1	0	1	0	1	0	1
Area 6 IO except SDRAM initialization (Internal Area 6)	Wait-cycle	NA	0	0	0	0	0	0	0	0	0	0	0	0
	Output-disable		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Read-hold		0	1	0	1	0	1	0	1	0	1	0	1
USB IO (Internal Area 11)	Wait-cycle	NA	NA		NA		NA		NA		NA		NA	
	Output-disable													
	Read-hold			1		1		1		1		1		1
USB DMA (Internal Area 4)	Wait-cycle	NA	NA		NA		NA		NA		NA		NA	
	Output-disable													
	Read-hold			1		1		1		1		1		1
SDRAM (External)	Wait-cycle	2	0	0	0 (Note 3)	0 (Note 3)	1	1	1	1	1	1	1	1
	Output-disable	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1.5	1.5	0.5	0.5	1.5	1.5
	Read-hold	0	0	0	0	0	0	0	0	0	0	0	0	0
SRAM/ROM (External)	Wait-cycle	Any	Any	Any	1	1	Any	Any	1	1	Any	Any	1	1
	Output-disable	Any	Any	Any	0.5	0.5	Any	Any	0.5	0.5	Any	Any	0.5	0.5
	Read-hold	Any	Any	Any	0	0	Any	Any	1	1	Any	Any	1	1
NAND flash (External)	Wait-cycle	NA	2	2	2	2	NA	NA	NA	NA	2	2	2	2
	Output-disable		0.5	0.5	0.5	0.5					0.5	0.5	0.5	0.5
	Read-hold		0	0	0	0					0	0	1	1
SQ ROM (External)	Wait-cycle	NA	5	5	5	5	6	6	6	6	4	4	4	4
	Output-disable		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Read-hold		1	1	1	1	1	1	1	1	1	1	1	1

Note 1: Besides the peripheral blocks and memories, the bus arbiter and USB functions affect the minimum access time.

Note 2: There are three setup conditions for Area 6 when using the bus in x1 speed mode; (1) 2-wait cycles for SDRAM initialization, (2) 1-wait cycle for IVRAM, and (3) 0-wait cycles for other I/O devices. To achieve high-performance, set the Area 6 wait-cycle condition to 2-wait cycles before initializing SDRAM and change it to 1-wait cycle after SDRAM initialization has completed.

Note 3: In this condition, the SDRAMC must be operated in low performance mode (SDHP [D3/0x3A0210] = "0").

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 53 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

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-minuet, 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	
83–88	reserved	–	–	
89(Base+164)	LCD controller	LCD controller interrupt	43	
90(Base+168)	USB function controller	USB function controller interrupt	44	
91(Base+16C)	SPI	SPI interrupt	45	
92–111	reserved	–	–	
112(Base+1C0)	Serial interface with FIFO	Receive error	–	
113(Base+1C4)	Ch.0	Receive buffer full	56	↓
114(Base+1C8)		Transmit buffer empty	57	Low

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.

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".

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.

Control of Maskable Interrupts

Structure of the Interrupt Controller

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

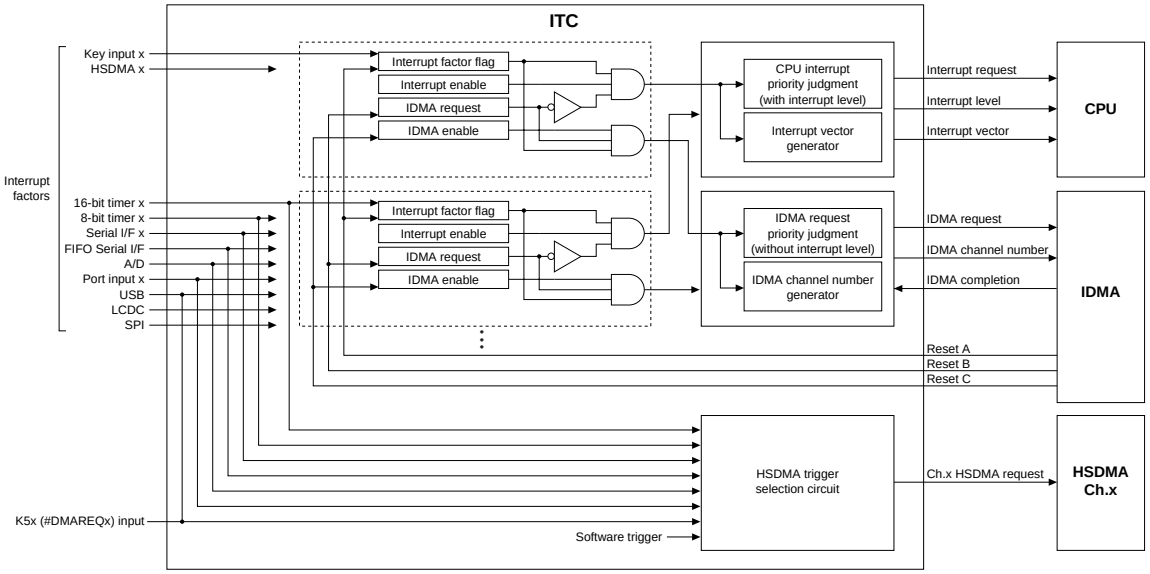


Figure II.5.1 Configuration of Interrupt Controller

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 < interrupt priority register: the interrupt request is accepted
IE bit = "1" & IL ≥ 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 (DINTEM = "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".

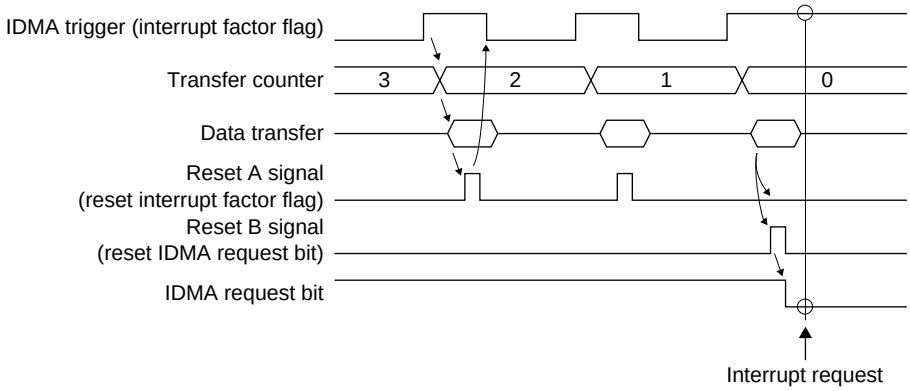


Figure II.5.2 Sequence when DINTEN = "1"

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".

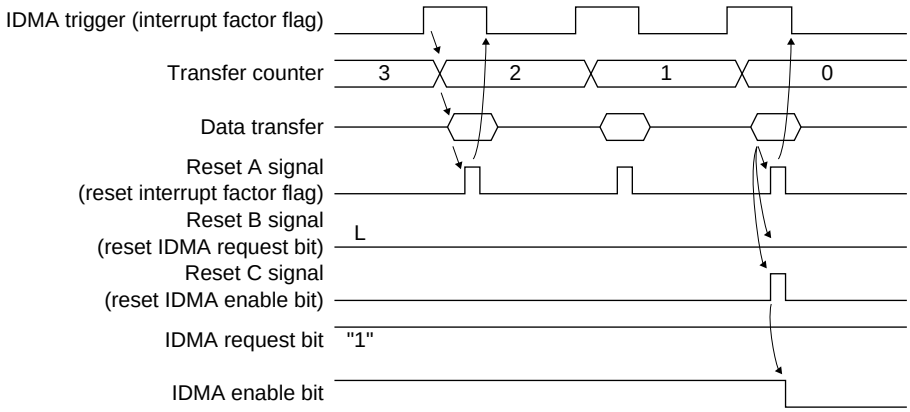


Figure II.5.3 Sequence when DINTEN = "0"

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	0 to 7	X	R/W	
		D5	**PHSD1L1**	interrupt level		X		
		D4	**PHSD1L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PHSD0L2**	High-speed DMA Ch.0	0 to 7	X	R/W	
		D1	**PHSD0L1**	interrupt level		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	0 to 7	X	R/W	
		D5	**PHSD3L1**	interrupt level		X		
		D4	**PHSD3L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PHSD2L2**	High-speed DMA Ch.2	0 to 7	X	R/W	
		D1	**PHSD2L1**	interrupt level		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		

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**			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		
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	0 to 7	X	R/W	
		D5	**PSIO01**	interrupt level		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	0 to 7	X	R/W	
		D1	**PSIO11**	interrupt level		X		
		D0	**PSIO10**			X		
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**			X		
		D0	**PCTM0**			X		
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**			X		
		D4	**PP5L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PP4L2**	Port input 4 interrupt level	0 to 7	X	R/W	
		D1	**PP4L1**			X		
		D0	**PP4L0**			X		
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**			X		
		D4	**PP7L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PP6L2**	Port input 6 interrupt level	0 to 7	X	R/W	
		D1	**PP6L1**			X		
		D0	**PP6L0**			X		
Serial I/F Ch.2/3 interrupt priority register	**004026E (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PSIO32**	Serial interface Ch.3	0 to 7	X	R/W	
		D5	**PSIO31**	interrupt level		X		
		D4	**PSIO30**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PSIO22**	Serial interface Ch.2	0 to 7	X	R/W	
		D1	**PSIO21**	interrupt level		X		
		D0	**PSIO20**			X		
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)	D7–4	–	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	
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	–				–	–	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	–				–	–	0 when being read.

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	
		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.
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	**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	–				–	–	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	
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	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	

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	0	Software trigger			0	R/W	
		D6	HSD1S2	trigger set-up	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	0	Software trigger			0	R/W	
		D2	HSD0S2	trigger set-up	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					

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
High-speed DMA Ch.2/3 trigger set-up register	**0040299 (B)**	D7 D6 D5 D4	**HSD3S3** **HSD3S2** **HSD3S1** **HSD3S0**	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 0 0 0	R/W	
		D3 D2 D1 D0	**HSD2S3** **HSD2S2** **HSD2S1** **HSD2S0**	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 0 0 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	
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 D5 D4	**–** **–** **–**	Interrupt level of extended function (reserved)	0 to 7	X X X	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2 D1 D0	**–** **–** **–**	Interrupt level of extended function (reserved)	0 to 7	X X X	R/W	
Interrupt priority register for functions extended	**00402A1 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6 D5 D4	**–** **–** **–**	Interrupt level of extended function (reserved)	0 to 7	X X X	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2 D1 D0	**–** **–** **–**	Interrupt level of extended function (reserved)	0 to 7	X X X	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC interrupt priority register	**00402A2 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PLCDCI2**	LCD controller interrupt level	0 to 7	X	R/W	
		D5	**PLCDCI1**			X		
		D4	**PLCDCI0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**–**	Interrupt level of extended	0 to 7	X	R/W	
		D1	**–**	function (reserved)		X		
		D0	**–**			X		
USB, SPI interrupt priority register	**00402A3 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PSPII2**	SPI interrupt level	0 to 7	X	R/W	
		D5	**PSPII1**			X		
		D4	**PSPII0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PUSBI2**	USB interrupt level	0 to 7	X	R/W	
		D1	**PUSBI1**			X		
		D0	**PUSBI0**			X		
Interrupt priority register for functions extended	**00402A4 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**–**	Interrupt level of extended	0 to 7	X	R/W	
		D5	**–**	function (reserved)		X		
		D4	**–**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**–**	Interrupt level of extended	0 to 7	X	R/W	
		D1	**–**	function (reserved)		X		
		D0	**–**			X		
Interrupt priority register for functions extended	**00402A5 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**–**	Interrupt level of extended	0 to 7	X	R/W	
		D5	**–**	function (reserved)		X		
		D4	**–**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**–**	Interrupt level of extended	0 to 7	X	R/W	
		D1	**–**	function (reserved)		X		
		D0	**–**			X		
LCDC, USB, SPI interrupt enable register	**00402A6 (B)**	D7	**ESPII**	SPI interrupts	1 Enabled 0 Disabled	0	R/W	
		D6	**EUSBI**	USB interrupts		0	R/W	
		D5	**ELCDCI**	LCDC interrupts		0	R/W	
		D4	**–**	Extended interrupt (reserved)		0	R/W	Do not write 1.
		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	
LCDC, USB, SPI interrupt factor flag register	**00402A9 (B)**	D7	**FSPII**	SPI interrupts	1 Factor is generated 0 No factor is generated	X	R/W	
		D6	**FUSBI**	USB interrupts		X	R/W	
		D5	**FLCDCI**	LCDC interrupts		X	R/W	
		D4	**–**	Extended interrupt (reserved)		X	R/W	Do not write 1.
		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	

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	
LCDC, USB, SPI IDMA request register	**00402AC (B)**	D7	**RSPII**	SPI interrupts	1	IDMA request	0	Interrupt request	0	R/W	
		D6	**RUSBI**	USB interrupts					0	R/W	
		D5	**RLCDCI**	LCDC interrupts					0	R/W	
		D4	**–**	Extended interrupt (reserved)					0	R/W	Do not write 1.
		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	
LCDC, USB, SPI IDMA enable register	**00402AE (B)**	D7	**DESPII**	SPI interrupts	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D6	**DEUSBI**	USB interrupts					0	R/W	
		D5	**DELCDCI**	LCDC interrupts					0	R/W	
		D4	**–**	Extended interrupt (reserved)					0	R/W	Do not write 1.
		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.

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.
		D8	TTBR08			0		Writing 1 not allowed.
		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.
		DE	TTBR32			0		Writing 1 not allowed.
		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).

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 not 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.

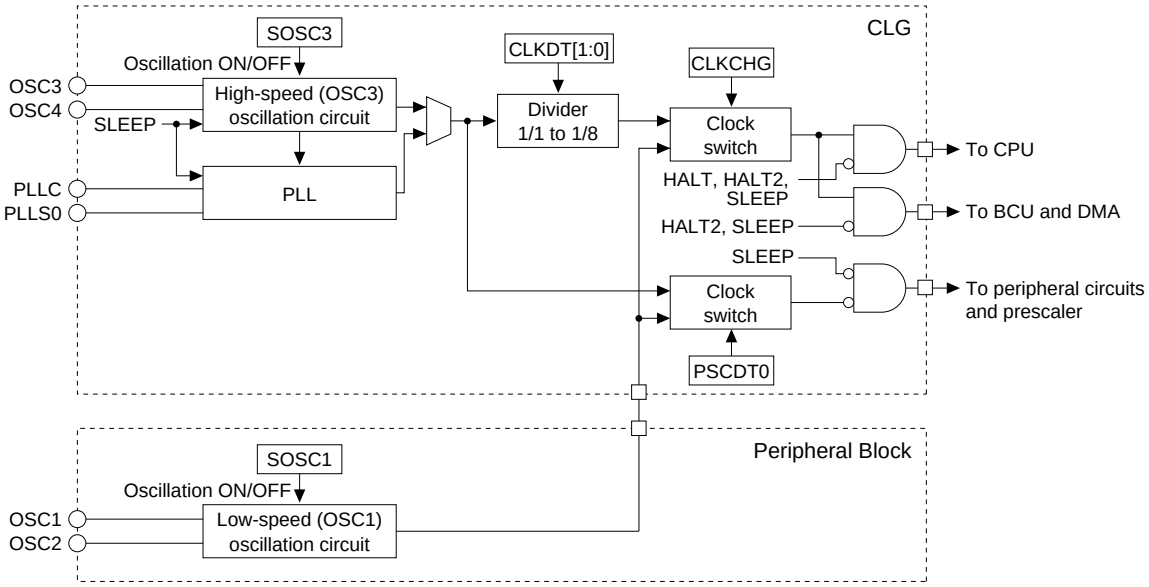


Figure II.6.1 Configuration of Clock Generator

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

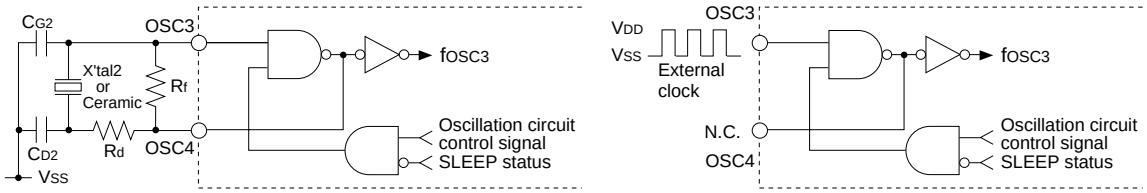
Table II.6.1 lists the I/O pins of the clock generator.

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	–	Capacitor connecting pin for PLL
PLLS0	I	PLL set-up pins 0: PLL not used, 1: PLL enabled

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.
When using the USB function, the OSC3 oscillation frequency must be 48 MHz. Since the USB function controller inputs the OSC3 clock, the multiplied clock output by the PLL cannot be used as the USB clock.



(1) Crystal (ceramic) oscillation circuit (2) External clock input

Figure II.6.2 High-Speed (OSC3) Oscillation Circuit

When using a crystal (ceramic) oscillation for this circuit, connect a crystal (X'tal2) or ceramic (Ceramic) resonator, a feedback resistor (R_f), two capacitors (C_{G2}, C_{D2}), and a drain resistor (R_d), if necessary, to the OSC3 and OSC4 pins and V_{SS} 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 48 MHz. When an external clock is input, a clock within the range from 2 MHz to 48 MHz can be used.

For details on oscillation characteristics and the external clock input characteristics, refer to "Electrical Characteristics".

Note: Oscillation characteristics vary depending on conditions (components used, board pattern, etc.). In particular, ceramic oscillation is extremely sensitive to influence of external components and printed-circuit boards. Before using a ceramic resonator, please be sure to contact Murata Manufacturing Co., Ltd. for further information on conditions of use for ceramic resonators. Furthermore, this chip supports only 48-MHz ceramic resonators. Do not use ceramic resonators with any other frequency.

PLL

The PLL inputs the OSC3 clock and multiply its frequency. The PLL mode should be set using the PLLS0 pin.

Table II.6.2 Setting the PLLS0 Pin

PLLS0	Mode	fin	fout
1	PLL enabled	10 to 24 MHz	10 to 24 MHz (x1) or 20 to 48 MHz (x2) *
0	PLL not used	–	Not used

* Refer to Section III.17.

Note: In the default setting of the S1C33L05 when the PLLS0 pin is set to "1", the PLL input clock frequency is OSC3/2 and the PLL outputs the clock without doubling it.
Refer to Section III-17, "S1C33L05 Clock System and Miscellaneous Registers".

Figure II.6.3 shows a basic external connection diagram for the PLL pins.

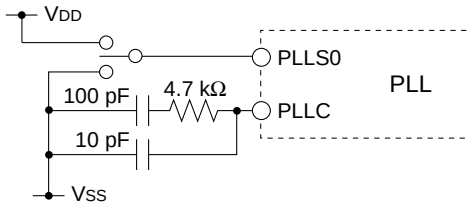


Figure II.6.3 External Connection Diagram

Note: When the PLL is not used, the OSC3 oscillation output is used as the source clock.

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 CLKDT[1:0] (D[7:6]) / Power control register (0x40180) for this switchover.

Table II.6.3 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

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 fout/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 CLKDT[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 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 (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 waitting 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	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	
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		

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).

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	• The CPU clock is stopped. (CPU stop status) • BCU clock is supplied. (BCU run status) • DMA clock is not stopped. (DMA run status) • Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop) • The high-speed oscillation circuit maintains the status before entering HALT mode. • The low-speed oscillation circuit maintains the status before entering HALT mode.	• Reset, NMI • Enabled (not masked) interrupt factors
	HALT2 mode	• The CPU clock is stopped. (CPU stop status) • BCU clock is stopped. (BCU stop status) • DMA clock is stopped. (DMA stop status) • Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop) • The high-speed oscillation circuit maintains the status before entering HALT mode. • The low-speed oscillation circuit maintains the status before entering HALT mode.	A restart is possible only in the case of: • Reset, NMI • Enabled (not masked) interrupt factors 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		• The CPU clock is stopped. (CPU stop status) • BCU clock is stopped. (BCU stop status) • Clocks for the peripheral circuits are stopped. • The high-speed oscillation circuit is stopped. • The low-speed oscillation circuit maintains the status before entering SLEEP mode.	• Reset, NMI • Enabled (not masked) input port interrupt factors • Clock timer interrupt when the low-speed oscillation circuit is being operated

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.

(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)	Operating
System clock	–	OSC1	OSC1	OSC3	OSC3	OSC3
OSC3 oscillation circuit	OFF	OFF	OFF	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 can 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).

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.

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 S1C33L05 comes with extended peripheral circuits, chip ID and pin control registers, serial interface with FIFO, MMC (SPI mode) interface, NAND flash interface, sequential ROM interface and extended I/O ports.

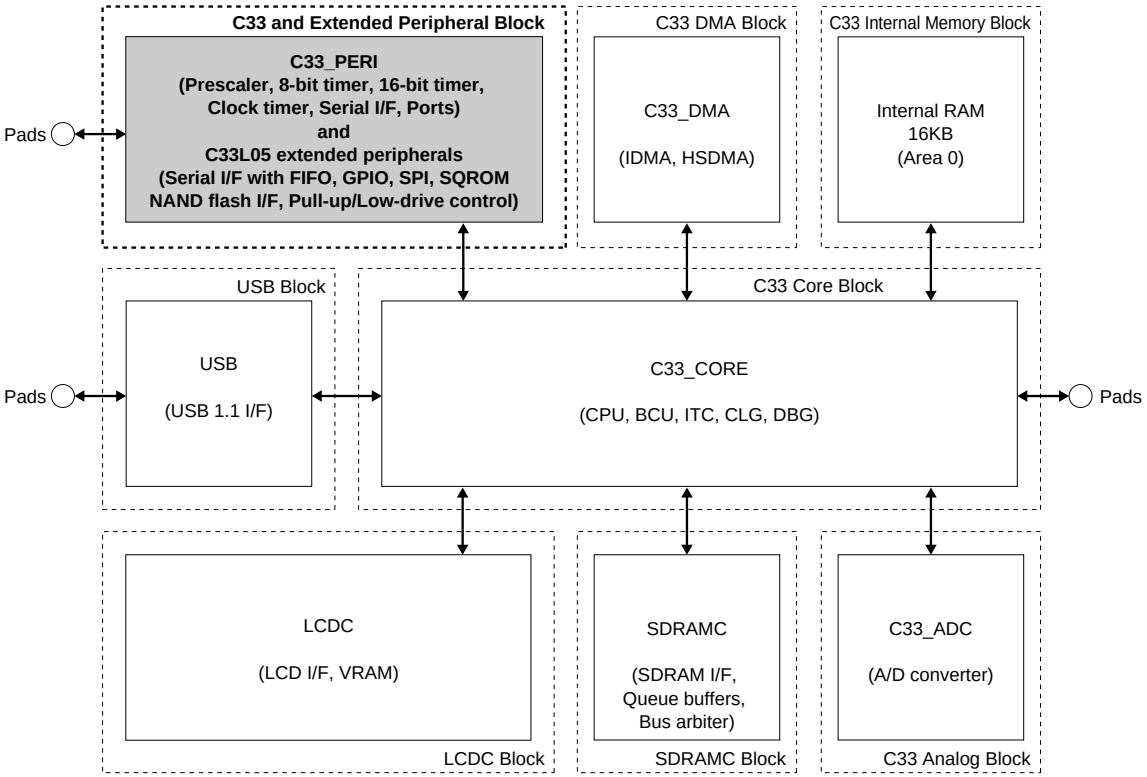


Figure III.1.1 Peripheral Block

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 S1C33L05 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 S1C33L05 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 S1C33L05 is 0x15.

Model ID register (0x300002)

This register provides an 8-bit ID code that indicates the model.
The model ID of the S1C33L05 is 0x05.

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 S1C33L05 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 (PA0), #HCAS (PA1), BCLK (P60)
	PUPCE (D2)	#CE10EX, #CE[9:4] (P5[5:0])
	PUPAD (D1)	A[25:18] (P4[0:7]), A[17:0]
	PUPRW (D0)	#WRH, #WRL, #RD
Input port pull-up control register (0x300F02)	PUPK6H (D2)	K64, BOOT
	PUPK6L (D1)	K6[3:0]
	PUPK5 (D0)	K5[3:0]
I/O port pull-up control register (0x300F04)	PUPP6 (D6)	P6[3:1]
	PUPP3 (D3)	P3[5:0] (PA2)
	PUPP2 (D2)	P2[7:0]
	PUPP1 (D1)	P1[6:0]
	PUPP0 (D0)	P0[7:0]
PB, PC port pull-up control register (0x300F06)	PUPPBH (D4)	PB[7:4]
	PUPPBL (D3)	PB[3:0]
	PUPPCL (D2)	PC[3:0]
PD port pull-up control register (0x300F08)	PUPPDH (D1)	PD[7:4]
	PUPPDL (D0)	PD[3:0]

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".

Note: 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.

Driving Bus Signals Low

The S1C33L05 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, #HCAS, BCLK
	LDRVCE (D2)	#CE10EX, #CE[9:4]
	LDRVAD (D1)	A[25:0]
	LDRVRW (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 0x300F08. 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)
   The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.

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		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	0x15	0	R	
		D6	**MID6**	0x03: S1C333xx		0		
		D5	**MID5**	0x15: S1C33Lxx		0		
		D4	**MID4**			1		
		D3	**MID3**			0		
		D2	**MID2**			1		
		D1	**MID1**			0		
		D0	**MID0**			1		
Model ID register	**0300002 (B)**	D7	**NAME7**	Model ID	0x05	0	R	
		D6	**NAME6**	0x04: S1C33L04		0		
		D5	**NAME5**	0x05: S1C33L05		0		
		D4	**NAME4**	0x11: S1C33L11		0		
		D3	**NAME3**			0		
		D2	**NAME2**			1		
		D1	**NAME1**			0		
		D0	**NAME0**			1		
Version register	**0300003 (B)**	D7	**VER3**	Version code	0x00	0	R	
		D6	**VER2**	0x00: version 1.0		0		
		D5	**VER1**			0		
		D4	**VER0**			0		
		D3–0	**–**	reserved	–	–	–	0 when being read.
Bus signal pull-up control register	**0300F00 (B)**	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**PUPCAS**	#HCAS, #LCAS, BCLK (PA1, PA0, P60) pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D2	**PUPCE**	#CE10EX–#CE4 (P55–P50) pull-up		1	R/W	
		D1	**PUPAD**	A25–A0 (P40–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**	#HCAS, #LCAS, BCLK low drive		0	R/W	
		D2	**LDRVCE**	#CE10EX–#CE4 low drive		0	R/W	
		D1	**LDRVAD**	A25–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**	K64, BOOT pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D1	**PUPK6L**	K63–K60 pull-up		1	R/W	
		D0	**PUPK5**	K53–K50 pull-up		1	R/W	
I/O port pull-up control register	**0300F04 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PUPP6**	P63–P61 pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D5–4	**–**	reserved	–	–	–	0 when being read.
		D3	**PUPP3**	P35–P30 (PA2) 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	
PB, PC port pull-up control register	**0300F06 (B)**	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**PUPPBH**	PB7–PB4 pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D3	**PUPPBL**	PB3–PB0 pull-up		1	R/W	
		D2	**PUPPCL**	PC3–PC0 pull-up		1	R/W	
		D1–0	**–**	reserved	–	–	–	0 when being read.
PD port pull-up control register	**0300F08 (B)**	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PUPPDH**	PD7–PD4 pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D0	**PUPPDL**	PD3–PD0 pull-up		1	R/W	

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 (0x15 = S1C33Lxx 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 (0x05 = S1C33L05) 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: #CE10EX–#CE4 pull-up control (D2) / <Bus signal pull-up control register (0x300F00)
PUPAD: A25–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 (PA0), #HCAS (PA1), BCLK (P60)
PUPCE #CE10EX, #CE[9:4] (P5[5:0])
PUPAD A[25:18] (P4[0:7]), 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: K64, BOOT 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: K53–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).

PUPP6: P63–P61 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).

PUPPBH: PB7–PB4 pull-up control (D4) / PB, PC port pull-up control register (0x300F06)
PUPPBL: PB3–PB0 pull-up control (D3) / PB, PC port pull-up control register (0x300F06)
PUPPCL: PC3–PC0 pull-up control (D2) / PB, PC port pull-up control register (0x300F06)
PUPPDH: PD7–PD4 pull-up control (D1) / PD port pull-up control register (0x300F08)
PUPPDL: PD3–PD0 pull-up control (D0) / PD port pull-up control register (0x300F08)

Controls the pull-up resistors at the GPIO 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).

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: #CE10EX–#CE4 low drive control (D2) / Bus signal low-drive control register (0x300F01)
LDRVAD: A25–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, #HCAS, BCLK
LDRVCE	#CE10EX, #CE[9:4]
LDRVAD	A[25:0]
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.

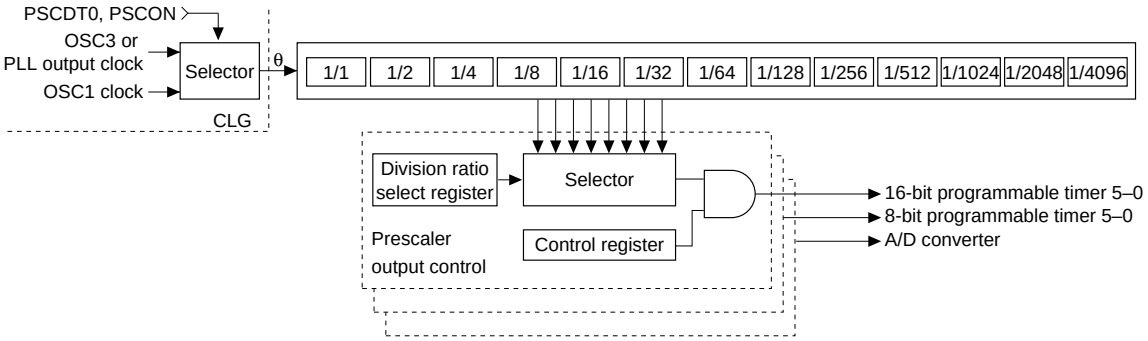


Figure III.3.1 Configuration of Prescaler and Clock Control Circuit

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	(D3/0x4014D)
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	θ/4096	θ/1024	θ/256	θ/64	θ/16	θ/4	θ/2	θ/1
*2	θ/256	θ/128	θ/64	θ/32	θ/16	θ/8	θ/4	θ/2
*3	θ/4096	θ/2048	θ/1024	θ/512	θ/256	θ/128	θ/64	θ/32
*4	θ/4096	θ/2048	θ/64	θ/32	θ/16	θ/8	θ/4	θ/2

(θ = 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 (θ/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 θ/1 0 Divided clk.	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D0	P8TPCK4	8-bit timer 4 clock selection	1 θ/1 0 Divided clk.	0	R/W	
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 θ/256 1 1 0 θ/128 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	0	R/W	θ: selected by Prescaler clock select register (0x40181) 8-bit timer 5 can generate the clock for the serial I/F Ch.3.
		D5	P8TS51			0	R/W	
		D4	P8TS50			0	R/W	
		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 θ/4096 1 1 0 θ/2048 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	0	R/W	θ: selected by Prescaler clock select register (0x40181) 8-bit timer 4 can generate the clock for the serial I/F Ch.2.
		D1	P8TS41			0	R/W	
		D0	P8TS40			0	R/W	
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 θ/1 0 Divided clk.	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D2	P8TPCK2	8-bit timer 2 clock selection	1 θ/1 0 Divided clk.	0	R/W	
		D1	P8TPCK1	8-bit timer 1 clock selection	1 θ/1 0 Divided clk.	0	R/W	
		D0	P8TPCK0	8-bit timer 0 clock selection	1 θ/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 1 1 1 θ/4096 1 1 0 θ/1024 1 0 1 θ/256 1 0 0 θ/64 0 1 1 θ/16 0 1 0 θ/4 0 0 1 θ/2 0 0 0 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181) 16-bit timer 0 can be used as a watchdog timer.
		D1	P16TS01			0		
		D0	P16TS00			0		
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 1 1 1 θ/4096 1 1 0 θ/1024 1 0 1 θ/256 1 0 0 θ/64 0 1 1 θ/16 0 1 0 θ/4 0 0 1 θ/2 0 0 0 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P16TS11			0		
		D0	P16TS10			0		
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 1 1 1 θ/4096 1 1 0 θ/1024 1 0 1 θ/256 1 0 0 θ/64 0 1 1 θ/16 0 1 0 θ/4 0 0 1 θ/2 0 0 0 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	P16TS21			0		
		D0	P16TS20			0		

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]: 1 1 1 θ/4096; 1 1 0 θ/1024; 1 0 1 θ/256; 1 0 0 θ/64; 0 1 1 θ/16; 0 1 0 θ/4; 0 0 1 θ/2; 0 0 0 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**P16TS31**			0		
		D0	**P16TS30**			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 clock division ratio selection	P16TS4[2:0]: 1 1 1 θ/4096; 1 1 0 θ/1024; 1 0 1 θ/256; 1 0 0 θ/64; 0 1 1 θ/16; 0 1 0 θ/4; 0 0 1 θ/2; 0 0 0 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**P16TS41**			0		
		D0	**P16TS40**			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 clock division ratio selection	P16TS5[2:0]: 1 1 1 θ/4096; 1 1 0 θ/1024; 1 0 1 θ/256; 1 0 0 θ/64; 0 1 1 θ/16; 0 1 0 θ/4; 0 0 1 θ/2; 0 0 0 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**P16TS51**			0		
		D0	**P16TS50**			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 clock division ratio selection	P8TS1[2:0]: 1 1 1 θ/4096; 1 1 0 θ/2048; 1 0 1 θ/1024; 1 0 0 θ/512; 0 1 1 θ/256; 0 1 0 θ/128; 0 0 1 θ/64; 0 0 0 θ/32	0	R/W	θ: selected by Prescaler clock select register (0x40181) 8-bit timer 1 can generate the OSC3 oscillation-stabilize waiting period.
		D5	**P8TS11**			0		
		D4	**P8TS10**			0		
		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]: 1 1 1 θ/256; 1 1 0 θ/128; 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	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**P8TS01**			0		
		D0	**P8TS00**			0		
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]: 1 1 1 θ/256; 1 1 0 θ/128; 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	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**			0		
		D4	**P8TS30**			0		
		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]: 1 1 1 θ/4096; 1 1 0 θ/2048; 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	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**			0		
		D0	**P8TS20**			0		

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 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			
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) 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		

III

PSC

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).

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 (θ/1)
Write "0": Divided clock
Read: Valid

When "1" is written to P8TPCKx, the prescaler input clock (θ/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.

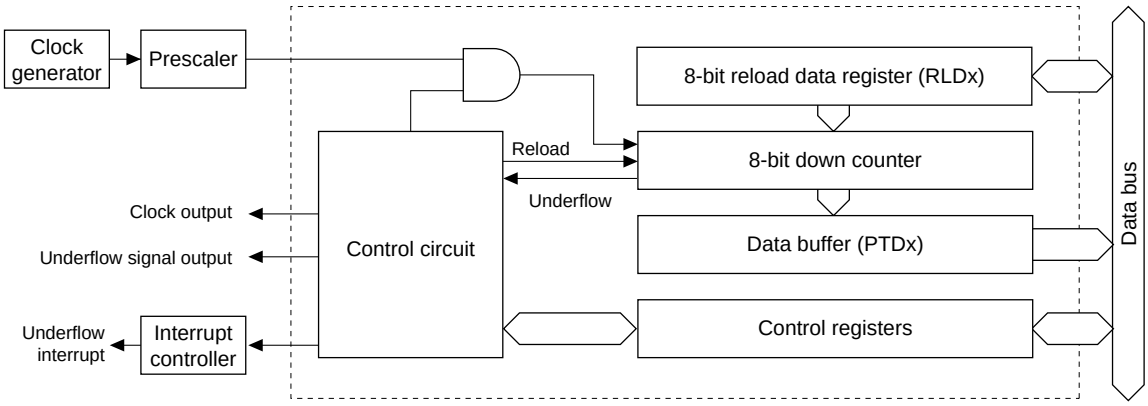


Figure III.4.1 Structure of 8-Bit Programmable Timer

Each timer consists of an 8-bit presentable 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 it is 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 RLDx 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 PSETx. 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 underflown 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 PSETx 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:

$$\text{Under flow cycle} = \frac{\text{RLDx} + 1}{f_{PSCIN} \times \text{pdr}} \quad [\text{sec.}]$$

f_{PSCIN}: Prescaler input clock frequency [Hz]
pdr: Prescaler division ratio set by P8TSx
RLDx: Set value of the RLDx register (0 to 255)

Timer RUN/STOP control

Each timer has a PTRUNx 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 PTRUNx. Writing "0" to PTRUNx 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 (PTRUNx) and the timer preset bit (PSETx) are set to "1" at the same time, the timer starts counting after presetting the reload register value into the counter.

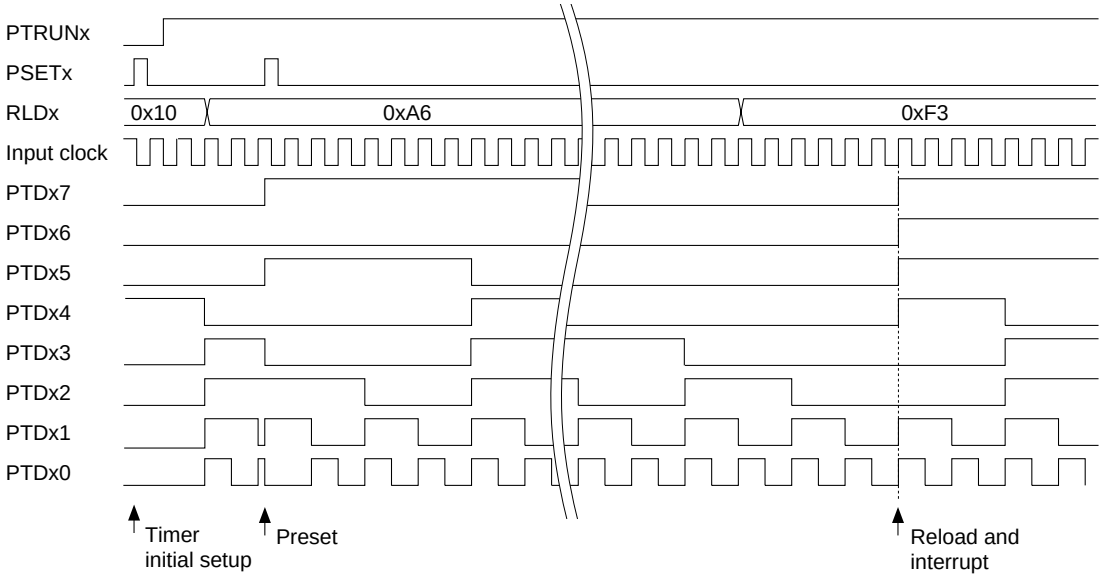


Figure III.4.2 Basic Operation Timing of Counter

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.

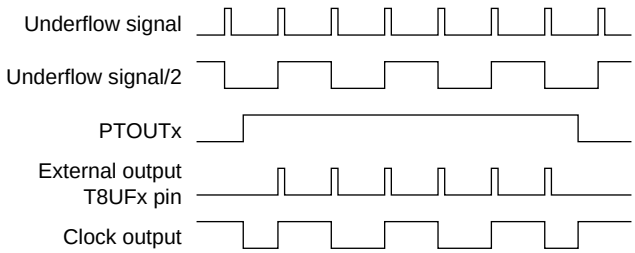


Figure III.4.3 8-Bit Programmable Timer Output Waveform

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)".

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)".

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		

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	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		
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	0 to 255				X	R/W	
		D6	**RLD46**	RLD47 = MSB					X		
		D5	**RLD45**	RLD40 = LSB					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	0 to 255				X	R	
		D6	**PTD46**	PTD47 = MSB					X		
		D5	**PTD45**	PTD40 = LSB					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	0 to 255				X	R/W	
		D6	**RLD56**	RLD57 = MSB					X		
		D5	**RLD55**	RLD50 = LSB					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	0 to 255				X	R	
		D6	**PTD56**	PTD57 = MSB					X		
		D5	**PTD55**	PTD50 = LSB					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	0 to 7				X	R/W	
		D5	**PSIO01**	interrupt level					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	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	
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	
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	
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	

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 port s 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".

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.

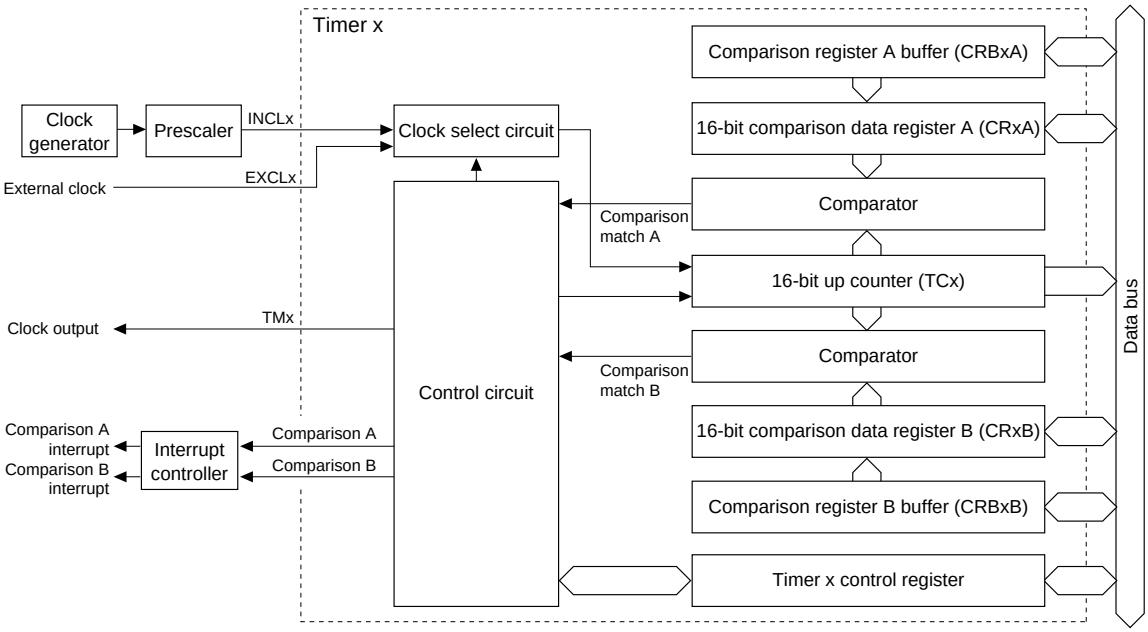


Figure III.5.1 Structure of 16-Bit Programmable Timer

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
f_{PSCIN}/4096	f_{PSCIN}/1024	f_{PSCIN}/256	f_{PSCIN}/64	f_{PSCIN}/16	f_{PSCIN}/4	f_{PSCIN}/2	f_{PSCIN}/1

f_{PSCIN}: 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.

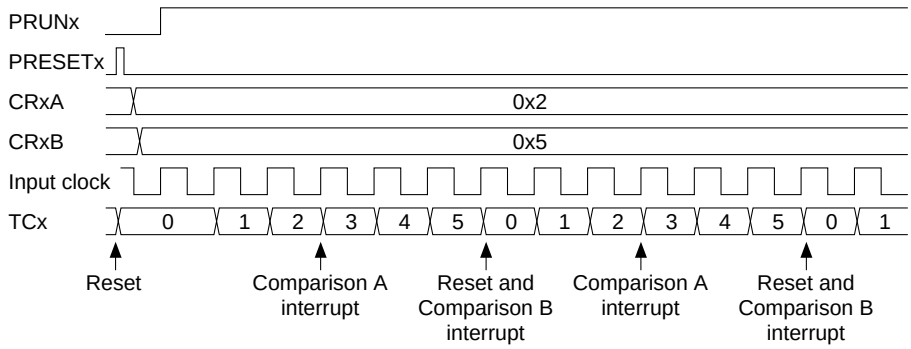


Figure III.5.2 Basic Operation Timing of Counter

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.

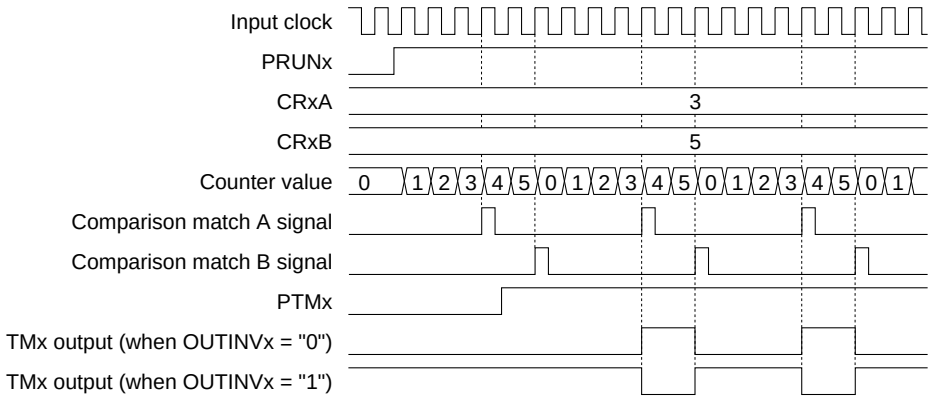


Figure III.5.3 Waveform of 16-Bit Programmable Timer Output

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

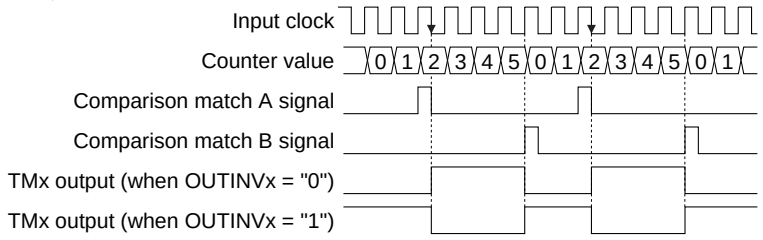


Figure III.5.4 Clock Output in Fine Mode

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 × 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 ≥ 0 and B ≥ 1. The minimum settings are A = 0 and B = 1. In this case, the timer output clock cycle is the input clock × 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

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"
	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "1001"
Timer 4 comparison B	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "1000"
	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "1000"
Timer 5 comparison A	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "1001"
	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2/3 trigger set-up register (0x40299) = "1001"
Timer 5 comparison B	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0/1 trigger set-up register (0x40298) = "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	

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**	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	
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	**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	

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	0 to 65535	X	R/W	
		DE	**CR1A14**	CR1A15 = MSB		X		
		DD	**CR1A13**	CR1A0 = LSB		X		
		DC	**CR1A12**			X		
		DB	**CR1A11**			X		
		DA	**CR1A10**			X		
		D9	**CR1A9**			X		
		D8	**CR1A8**			X		
		D7	**CR1A7**			X		
		D6	**CR1A6**			X		
		D5	**CR1A5**			X		
		D4	**CR1A4**			X		
		D3	**CR1A3**			X		
		D2	**CR1A2**			X		
		D1	**CR1A1**			X		
		D0	**CR1A0**			X		
16-bit timer 1 comparison data B set-up register	**004818A (HW)**	DF	**CR1B15**	16-bit timer 1 comparison data B	0 to 65535	X	R/W	
		DE	**CR1B14**	CR1B15 = MSB		X		
		DD	**CR1B13**	CR1B0 = LSB		X		
		DC	**CR1B12**			X		
		DB	**CR1B11**			X		
		DA	**CR1B10**			X		
		D9	**CR1B9**			X		
		D8	**CR1B8**			X		
		D7	**CR1B7**			X		
		D6	**CR1B6**			X		
		D5	**CR1B5**			X		
		D4	**CR1B4**			X		
		D3	**CR1B3**			X		
		D2	**CR1B2**			X		
		D1	**CR1B1**			X		
		D0	**CR1B0**			X		
16-bit timer 1 counter data register	**004818C (HW)**	DF	**TC115**	16-bit timer 1 counter data	0 to 65535	X	R	
		DE	**TC114**	TC115 = MSB		X		
		DD	**TC113**	TC10 = LSB		X		
		DC	**TC112**			X		
		DB	**TC111**			X		
		DA	**TC110**			X		
		D9	**TC19**			X		
		D8	**TC18**			X		
		D7	**TC17**			X		
		D6	**TC16**			X		
		D5	**TC15**			X		
		D4	**TC14**			X		
		D3	**TC13**			X		
		D2	**TC12**			X		
		D1	**TC11**			X		
		D0	**TC10**			X		
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	0 to 65535	X	R/W	
		DE	**CR2A14**	CR2A15 = MSB		X		
		DD	**CR2A13**	CR2A0 = LSB		X		
		DC	**CR2A12**			X		
		DB	**CR2A11**			X		
		DA	**CR2A10**			X		
		D9	**CR2A9**			X		
		D8	**CR2A8**			X		
		D7	**CR2A7**			X		
		D6	**CR2A6**			X		
		D5	**CR2A5**			X		
		D4	**CR2A4**			X		
		D3	**CR2A3**			X		
		D2	**CR2A2**			X		
		D1	**CR2A1**			X		
		D0	**CR2A0**			X		

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	0 to 65535	X	R/W	
		DE	**CR2B14**	CR2B15 = MSB		X		
		DD	**CR2B13**	CR2B0 = LSB		X		
		DC	**CR2B12**			X		
		DB	**CR2B11**			X		
		DA	**CR2B10**			X		
		D9	**CR2B9**			X		
		D8	**CR2B8**			X		
		D7	**CR2B7**			X		
		D6	**CR2B6**			X		
		D5	**CR2B5**			X		
		D4	**CR2B4**			X		
		D3	**CR2B3**			X		
		D2	**CR2B2**			X		
		D1	**CR2B1**			X		
		D0	**CR2B0**			X		
16-bit timer 2 counter data register	**0048194 (HW)**	DF	**TC215**	16-bit timer 2 counter data	0 to 65535	X	R	
		DE	**TC214**	TC215 = MSB		X		
		DD	**TC213**	TC20 = LSB		X		
		DC	**TC212**			X		
		DB	**TC211**			X		
		DA	**TC210**			X		
		D9	**TC29**			X		
		D8	**TC28**			X		
		D7	**TC27**			X		
		D6	**TC26**			X		
		D5	**TC25**			X		
		D4	**TC24**			X		
		D3	**TC23**			X		
		D2	**TC22**			X		
		D1	**TC21**			X		
		D0	**TC20**			X		
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	
16-bit timer 3 comparison data A set-up register	**0048198 (HW)**	DF	**CR3A15**	16-bit timer 3 comparison data A	0 to 65535	X	R/W	
		DE	**CR3A14**	CR3A15 = MSB		X		
		DD	**CR3A13**	CR3A0 = LSB		X		
		DC	**CR3A12**			X		
		DB	**CR3A11**			X		
		DA	**CR3A10**			X		
		D9	**CR3A9**			X		
		D8	**CR3A8**			X		
		D7	**CR3A7**			X		
		D6	**CR3A6**			X		
		D5	**CR3A5**			X		
		D4	**CR3A4**			X		
		D3	**CR3A3**			X		
		D2	**CR3A2**			X		
		D1	**CR3A1**			X		
		D0	**CR3A0**			X		
16-bit timer 3 comparison data B set-up register	**004819A (HW)**	DF	**CR3B15**	16-bit timer 3 comparison data B	0 to 65535	X	R/W	
		DE	**CR3B14**	CR3B15 = MSB		X		
		DD	**CR3B13**	CR3B0 = LSB		X		
		DC	**CR3B12**			X		
		DB	**CR3B11**			X		
		DA	**CR3B10**			X		
		D9	**CR3B9**			X		
		D8	**CR3B8**			X		
		D7	**CR3B7**			X		
		D6	**CR3B6**			X		
		D5	**CR3B5**			X		
		D4	**CR3B4**			X		
		D3	**CR3B3**			X		
		D2	**CR3B2**			X		
		D1	**CR3B1**			X		
		D0	**CR3B0**			X		

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					X		
		DD	**TC313**	TC30 = LSB					X		
		DC	**TC312**						X		
		DB	**TC311**						X		
		DA	**TC310**						X		
		D9	**TC39**						X		
		D8	**TC38**						X		
		D7	**TC37**						X		
		D6	**TC36**						X		
		D5	**TC35**						X		
		D4	**TC34**						X		
		D3	**TC33**						X		
		D2	**TC32**						X		
		D1	**TC31**						X		
		D0	**TC30**						X		
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					X		
		DD	**CR4A13**	CR4A0 = LSB					X		
		DC	**CR4A12**						X		
		DB	**CR4A11**						X		
		DA	**CR4A10**						X		
		D9	**CR4A9**						X		
		D8	**CR4A8**						X		
		D7	**CR4A7**						X		
		D6	**CR4A6**						X		
		D5	**CR4A5**						X		
		D4	**CR4A4**						X		
		D3	**CR4A3**						X		
		D2	**CR4A2**						X		
		D1	**CR4A1**						X		
		D0	**CR4A0**						X		
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					X		
		DD	**CR4B13**	CR4B0 = LSB					X		
		DC	**CR4B12**						X		
		DB	**CR4B11**						X		
		DA	**CR4B10**						X		
		D9	**CR4B9**						X		
		D8	**CR4B8**						X		
		D7	**CR4B7**						X		
		D6	**CR4B6**						X		
		D5	**CR4B5**						X		
		D4	**CR4B4**						X		
		D3	**CR4B3**						X		
		D2	**CR4B2**						X		
		D1	**CR4B1**						X		
		D0	**CR4B0**						X		
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					X		
		DD	**TC413**	TC40 = LSB					X		
		DC	**TC412**						X		
		DB	**TC411**						X		
		DA	**TC410**						X		
		D9	**TC49**						X		
		D8	**TC48**						X		
		D7	**TC47**						X		
		D6	**TC46**						X		
		D5	**TC45**						X		
		D4	**TC44**						X		
		D3	**TC43**						X		
		D2	**TC42**						X		
		D1	**TC41**						X		
		D0	**TC40**						X		

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	**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	
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					X		
		DD	**CR5A13**	CR5A0 = LSB					X		
		DC	**CR5A12**						X		
		DB	**CR5A11**						X		
		DA	**CR5A10**						X		
		D9	**CR5A9**						X		
		D8	**CR5A8**						X		
		D7	**CR5A7**						X		
		D6	**CR5A6**						X		
		D5	**CR5A5**						X		
		D4	**CR5A4**						X		
		D3	**CR5A3**						X		
		D2	**CR5A2**						X		
		D1	**CR5A1**						X		
		D0	**CR5A0**						X		
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					X		
		DD	**CR5B13**	CR5B0 = LSB					X		
		DC	**CR5B12**						X		
		DB	**CR5B11**						X		
		DA	**CR5B10**						X		
		D9	**CR5B9**						X		
		D8	**CR5B8**						X		
		D7	**CR5B7**						X		
		D6	**CR5B6**						X		
		D5	**CR5B5**						X		
		D4	**CR5B4**						X		
		D3	**CR5B3**						X		
		D2	**CR5B2**						X		
		D1	**CR5B1**						X		
		D0	**CR5B0**						X		
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					X		
		DD	**TC513**	TC50 = LSB					X		
		DC	**TC512**						X		
		DB	**TC511**						X		
		DA	**TC510**						X		
		D9	**TC59**						X		
		D8	**TC58**						X		
		D7	**TC57**						X		
		D6	**TC56**						X		
		D5	**TC55**						X		
		D4	**TC54**						X		
		D3	**TC53**						X		
		D2	**TC52**						X		
		D1	**TC51**						X		
		D0	**TC50**						X		
16-bit timer 5 control register	**00481AE (B)**	D7	**–**	reserved	–				0	–	0 when being read.
		D6	**SELFM5**	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	

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 port s 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.

SELFM0: Timer 0 fine mode selection (D6) / 16-bit timer 0 control register (0x48186)
SELFM1: Timer 1 fine mode selection (D6) / 16-bit timer 1 control register (0x4818E)
SELFM2: Timer 2 fine mode selection (D6) / 16-bit timer 2 control register (0x48196)
SELFM3: Timer 3 fine mode selection (D6) / 16-bit timer 3 control register (0x4819E)
SELFM4: Timer 4 fine mode selection (D6) / 16-bit timer 4 control register (0x481A6)
SELFM5: 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

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 Periheral 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.

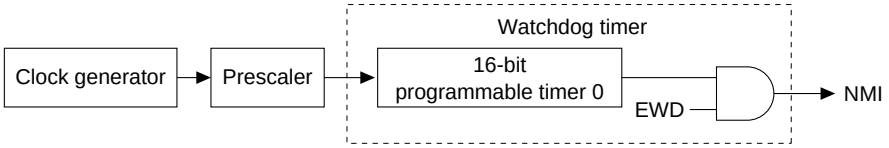


Figure III.6.1 Watchdog Timer Block Diagram

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:

$$\text{NMI generating interval} = \frac{\text{CR0B} + 1}{f_{PSCIN} \times \text{pdr}} \quad \text{[sec.]}$$

f_{PSCIN}: 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 reenable 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.

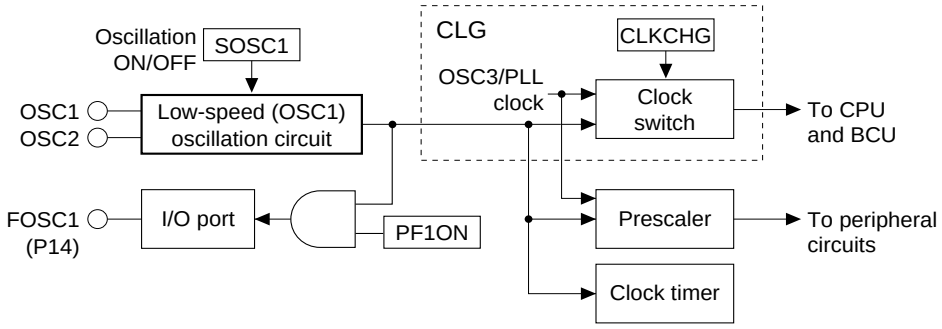


Figure III.7.1 Configuration of Clock System

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.

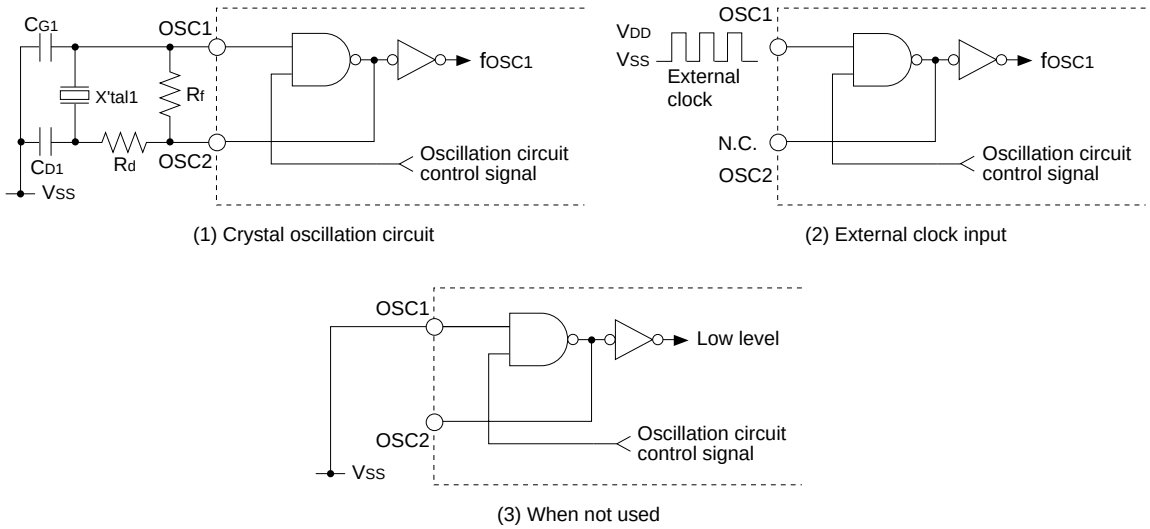


Figure III.7.2 Low-Speed (OSC1) Oscillation Circuit

When using a crystal oscillation for this circuit, connect a crystal resonator X'tal1 (32.768 kHz, Typ.), a feedback resistor (R_f), two capacitors (C_{G1}, C_{D1}), and a drain resistor (R_d), if necessary, to the OSC1 and OSC2 pins and V_{SS} 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 V_{SS} 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 / Low-speed (OSC1) oscillation clock output	CFP14 (D4) / P1 function select register (0x402D4) 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).

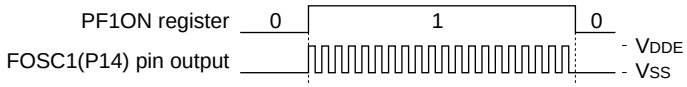


Figure III.7.3 OSC1 Clock Output

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 D6	**CLKDT1** **CLKDT0**	System clock division ratio selection	CLKDT[1:0]: 1 1 = 1/8; 1 0 = 1/4; 0 1 = 1/2; 0 0 = 1/1	0 0	R/W	
		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 D6 D5 D4 D3 D2 D1 D0	**CLGP7** **CLGP6** **CLGP5** **CLGP4** **CLGP3** **CLGP2** **CLGP1** **CLGP0**	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 0 0 0 0 0 0 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	
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	

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.

PF1ON: 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 PF1ON. 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 PF1ON is allowed only when CLGP[7:0] is set to "0b10010110".
At initial reset, PF1ON is set to "0" (Off).

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).

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	• The CPU clock is stopped. (CPU stop status) • BCU clock is supplied. (BCU run status) • DMA clock is not stopped. (DMA run status) • Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop) • The high-speed oscillation circuit maintains the status before entering HALT mode. • The low-speed oscillation circuit maintains the status before entering HALT mode.	• Reset, NMI • Enabled (not masked) interrupt factors
	HALT2 mode	• The CPU clock is stopped. (CPU stop status) • BCU clock is stopped. (BCU stop status) • DMA clock is stopped. (DMA stop status) • Clocks for the peripheral circuits maintain the status before entering HALT mode. (run or stop) • The high-speed oscillation circuit maintains the status before entering HALT mode. • The low-speed oscillation circuit maintains the status before entering HALT mode.	A restart is possible only in the case of: • Reset, NMI • Enabled (not masked) interrupt factors 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		• The CPU clock is stopped. (CPU stop status) • BCU clock is stopped. (BCU stop status) • Clocks for the peripheral circuits are stopped. • The high-speed oscillation circuit is stopped. • The low-speed oscillation circuit maintains the status before entering SLEEP mode.	• Reset, NMI • Enabled (not masked) input port interrupt factors • Clock timer interrupt when the low-speed oscillation circuit is being operated

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 fOSC1, 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, Typ.) is used.

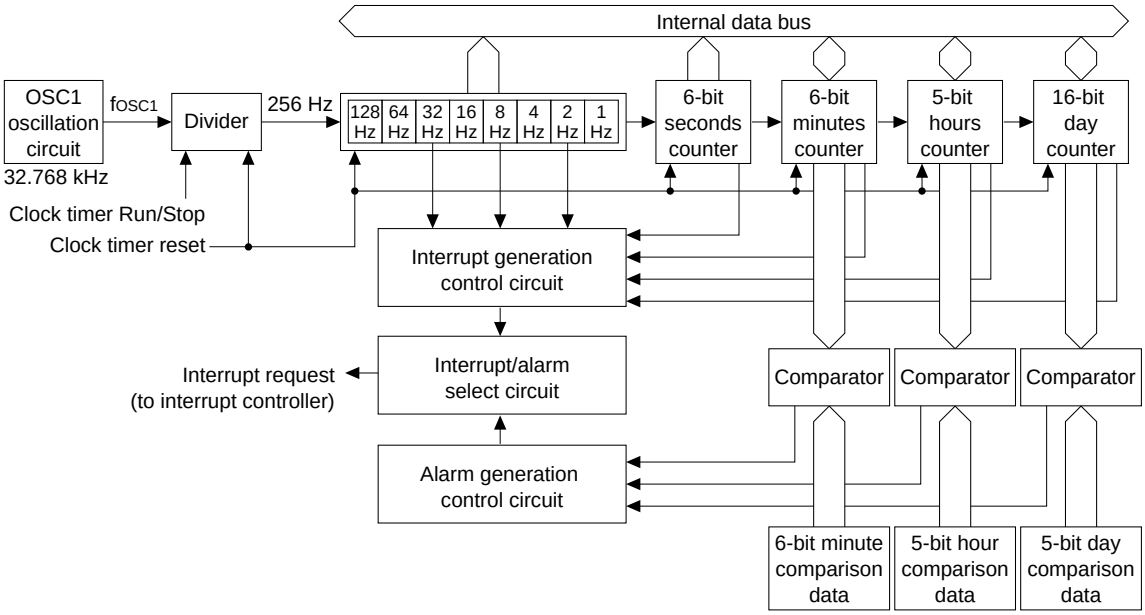


Figure III.8.1 Structure of Clock Timer

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.

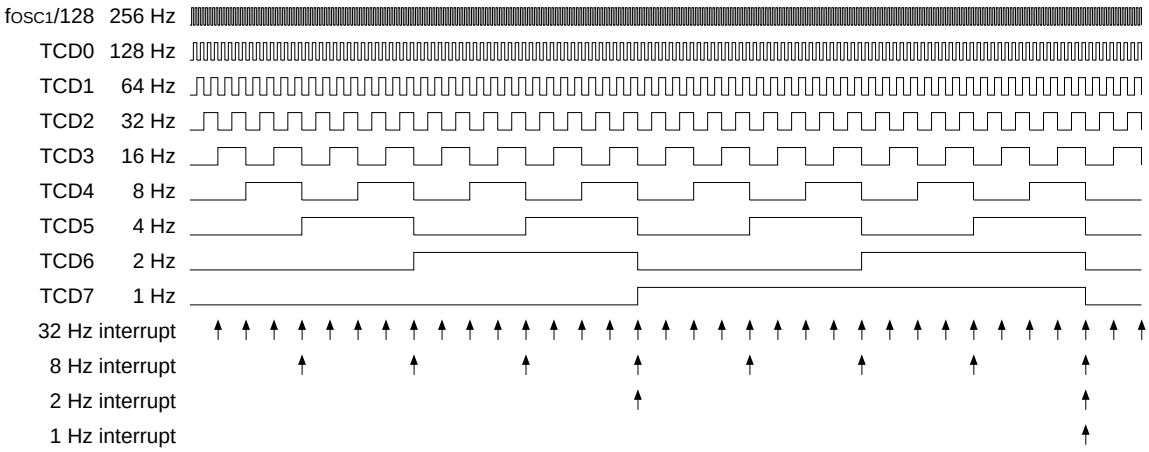


Figure III.8.2 Timing Chart of 8-Bit Binary Counter

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 actor 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 (low-order 8 bits)	0 to 65535 days (low-order 8 bits)	X	R/W	
		D6	TCND6			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 (high-order 8 bits)	0 to 65535 days (high-order 8 bits)	X	R/W	
		D6	TCND14			X		
		D5	TCND13	TCND15 = MSB		X		
		D4	TCND12			X		
		D3	TCND11			X		
		D2	TCND10			X		
		D1	TCND9			X		
		D0	TCND8			X		

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	0 to 59 minutes	X	R/W	
		D4	**TCCH4**	data	(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		
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**			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–TCD0: 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 been detected.

  Asynchronous mode
  - Data length: 7 or 8 bits, selectable
  - Receive error: Overrun, framing, or parity errors can been 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).

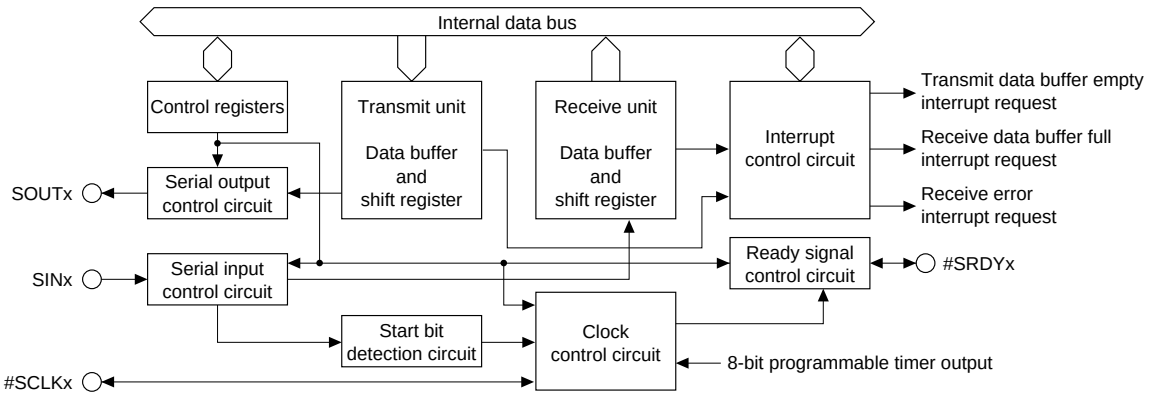


Figure III.9.1 Configuration of Serial Interface

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(**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 signal output	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 signal output	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 signal output	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 signal output	CFP07(D7)/P0 function select register(0x402D0) CFEX7(D7)/Port function extension register(0x402DF)
P27(TM5/**SIN2**)	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/**SOUT2**)	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/**#SCLK2**)	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/**#SRDY2**)	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(#DMAACK1/**SIN3**)	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(EXCL5/#DMAAND1/ **SOUT3**)	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(EXCL4/#DMAAND0/ **#SCLK3**)	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(#DMAACK0/ **#SRDY3**)	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, SSxxxx = "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.

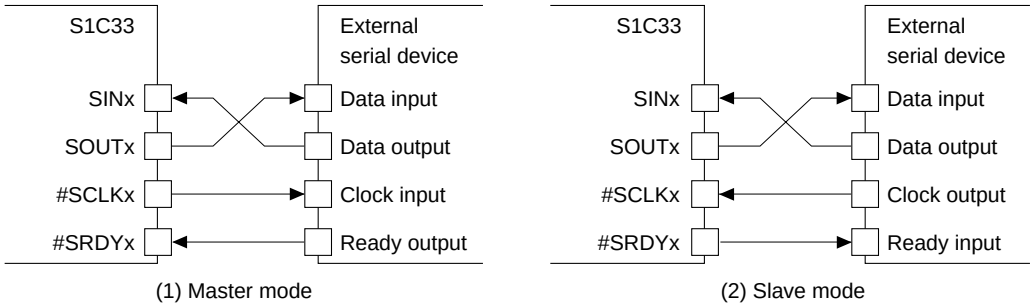


Figure III.9.2 Example of Connection in Clock-Synchronized Mode

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

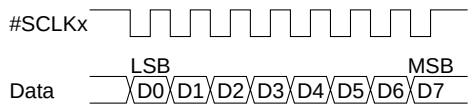


Figure III.9.3 Clock-Synchronized Transfer Data Format

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.

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.

$$RLD = \frac{f_{PSCIN} \times pdr}{2 \times bps} - 1 \qquad \text{(Eq. 1)}$$

RLD: Reload data register setup value of the 8-bit programmable timer
f_{PSCIN}: 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

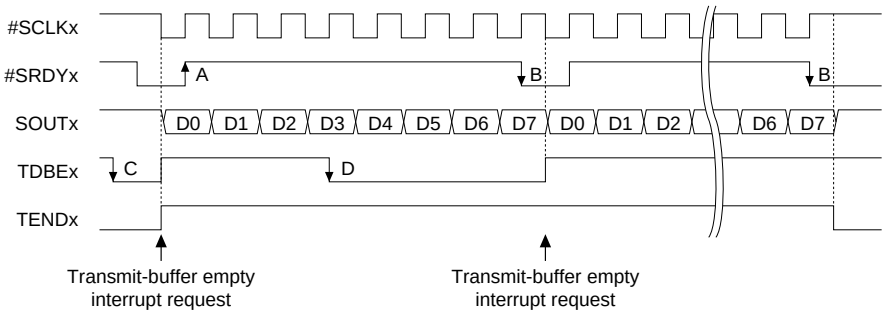
The serial interface contains a transmit shift register and a transmit data register (transmit data buffer), which are provided independently of those used for a receive operation.
Ch.0 transmit data: TXD0[7:0] (D[7:0]) / Serial I/F Ch.0 transmit data register (0x401E0)
Ch.1 transmit data: TXD1[7:0] (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.



A Slave device receives the LSB.
B Slave device receives the MSB.
C First data is written.
D Next data is written.

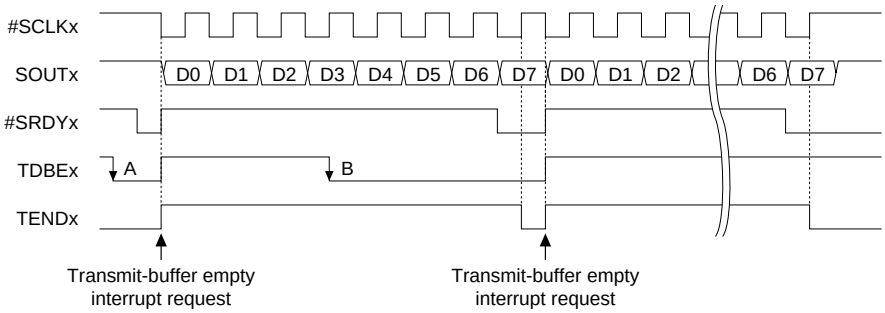
Figure III.9.4 Transmit Timing Chart in Clock-Synchronized Master Mode

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.



A First data is written.
B Next data is written.

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

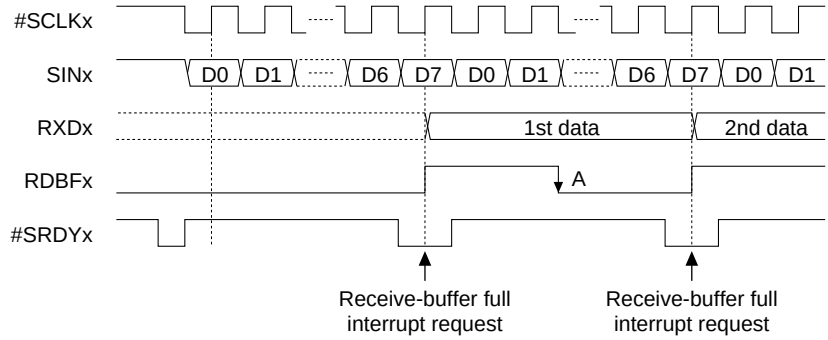
This serial interface has a receive shift register and a receive data register (receive data buffer) that are provided independently of those used for transmit operations.
Ch.0 receive data: RXD0[7:0] (D[7:0]) / Serial I/F Ch.0 receive data register (0x401E1)
Ch.1 receive data: RXD1[7:0] (D[7:0]) / Serial 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.



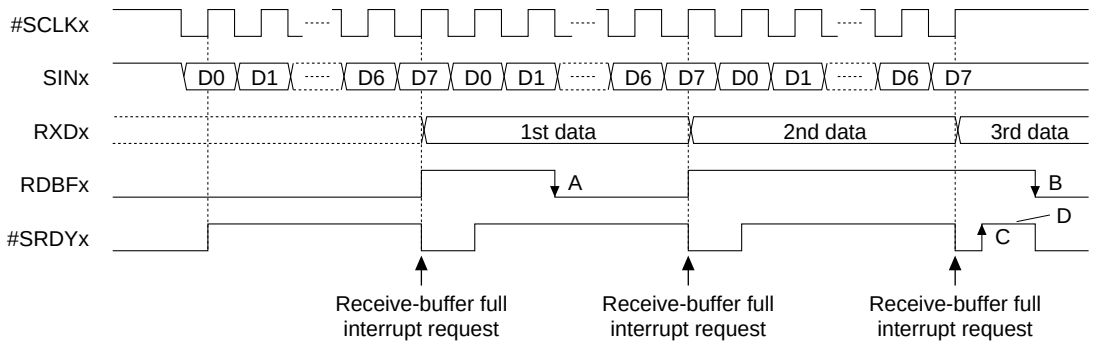
A First data is read.

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.
B 3rd data is read.
C An overrun error occurs because the receive operation has completed when RDBFx = "1".
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, RDBFx 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 RDBFx 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) #SRDYx in slave mode

When receive operations are enabled by writing "1" to RXENx, the #SRDYx 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, #SRDYx is turned to a high level; when the MSB is received, #SRDYx is returned to a low level, in preparation for the next receive operation.
If an overrun error occurs, #SRDYx is turned to a high level (unable to receive) at that point, with receive operations for the following data thus suspended. In this case, #SRDYx 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 RXENx 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.

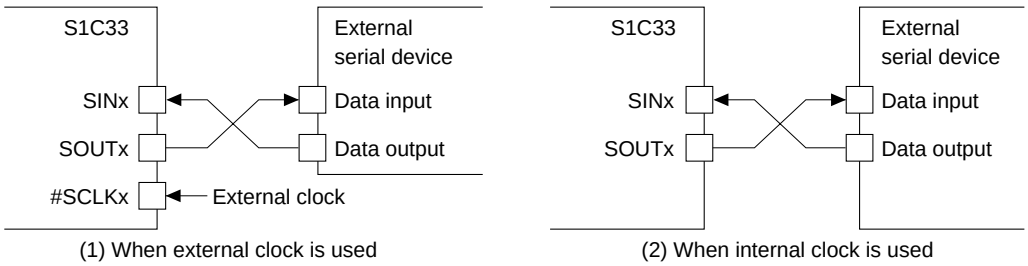


Figure III.9.8 Example of Connection in Asynchronous Mode

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

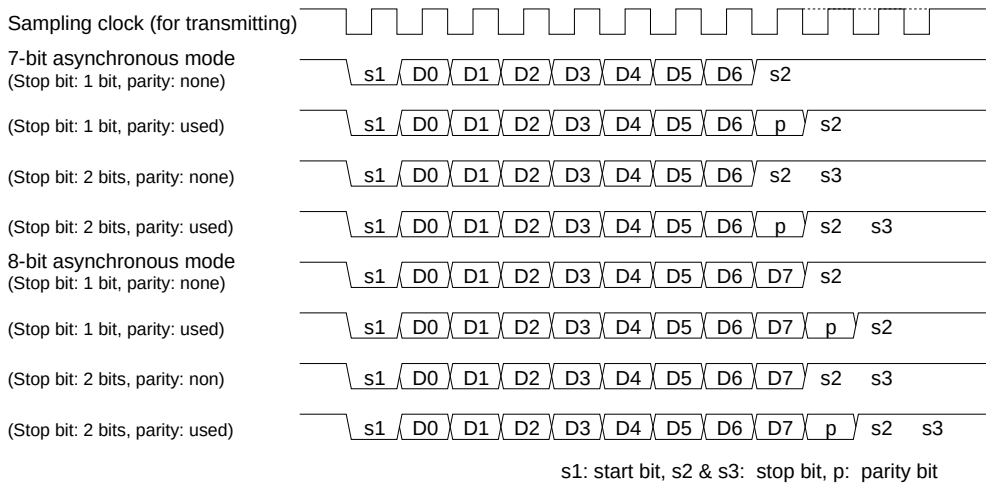


Figure III.9.9 Data Format for Asynchronous Transfer

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.

- **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.

$$RLD = \frac{f_{PSCIN} \times pdr \times sdr}{2 \times bps} - 1 \qquad \text{(Eq. 2)}$$

RLD: Set value of the 8-bit programmable timer's reload data register
f_{PSCIN}: 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	f_{PSCIN} = 20 MHz			f_{PSCIN} = 25 MHz			f_{PSCIN} = 33 MHz		
(bps)	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:

$$Error = \{\frac{f_{PSCIN} \times pdr}{(RLD + 1) \times 32 \times bps} - 1\} \times 100\ [\%]$$

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

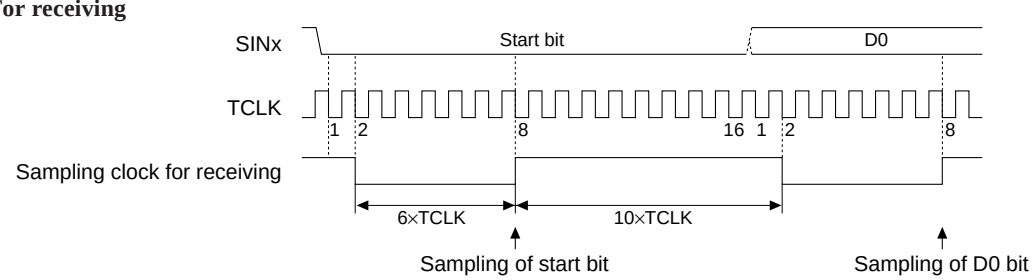


Figure III.9.10 Sampling Clock for Asynchronous Receive Operation (when 1/16 division is selected)

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

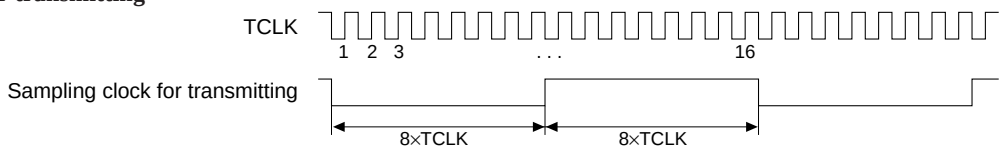


Figure III.9.11 Sampling Clock for Asynchronous Transmit Operation (when 1/16 division is selected)

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	*	2 bits	None
0	1	1	1 bit	Odd
		0	1 bit	Even
	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

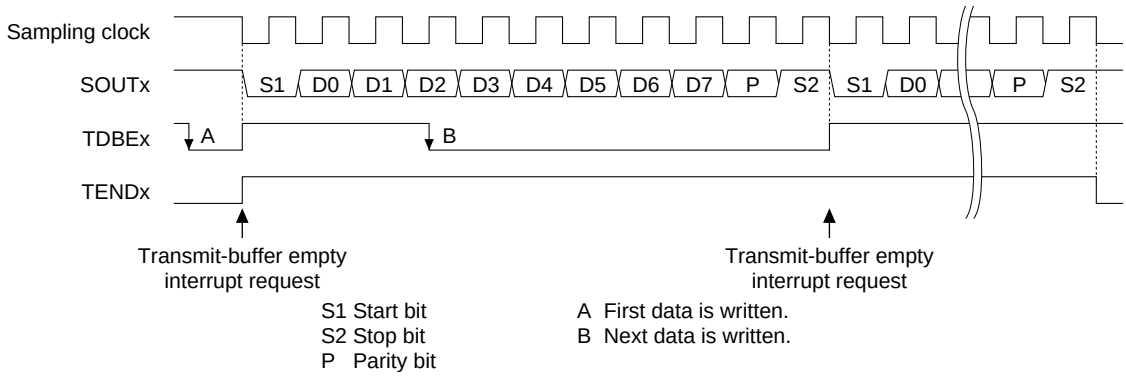


Figure III.9.12 Transmit Timing Chart in Asynchronous Mode

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 insuccession.

- **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

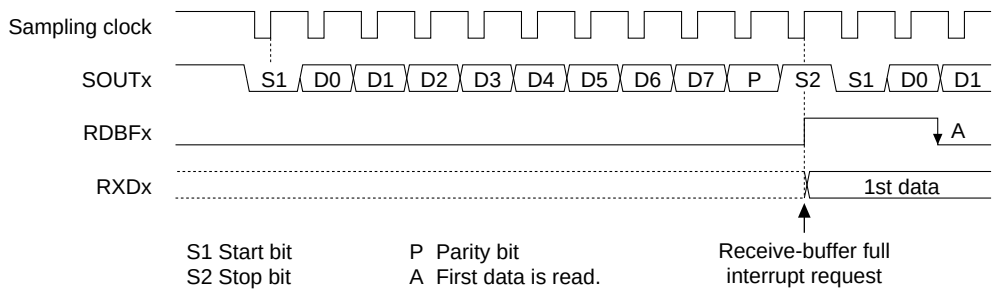


Figure III.9.13 Receive Timing Chart in Asynchronous Mode

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, RDBFx 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 RDBFx 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 EPRx 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 PMDx 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 PERx 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 FERx 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".

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.

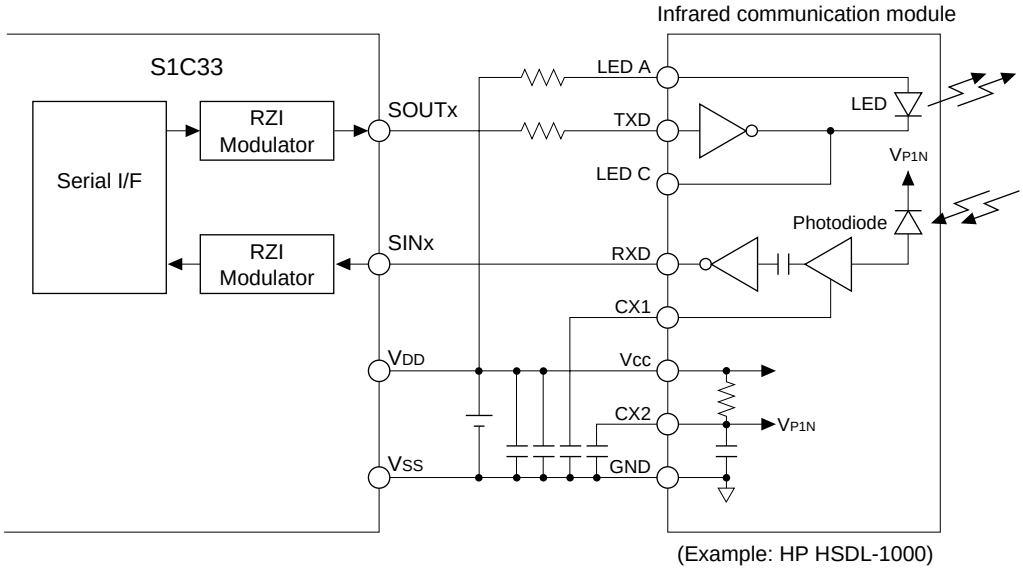


Figure III.9.14 Configuration Example of IrDA Interface

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".

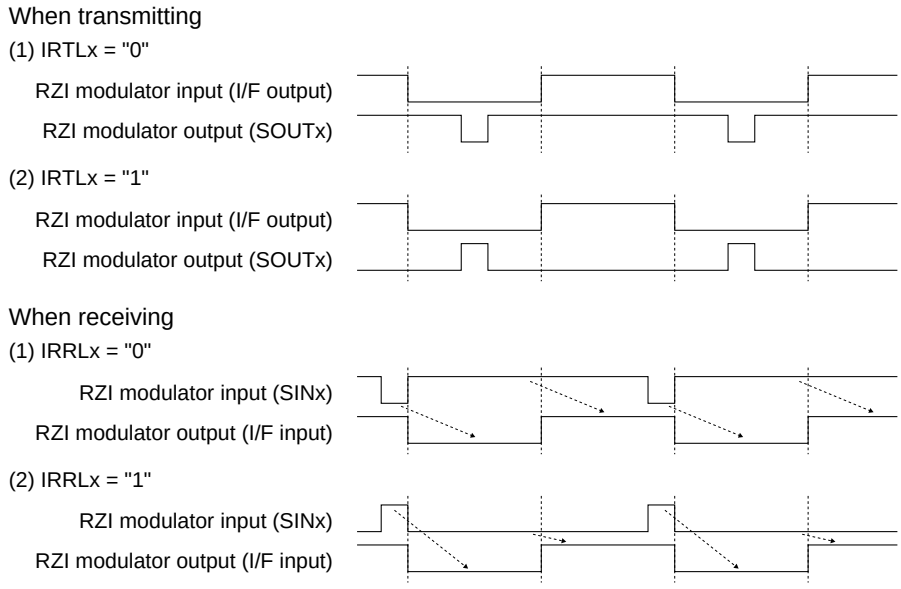


Figure III.9.15 IRRLx and IRTLx Settings

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.

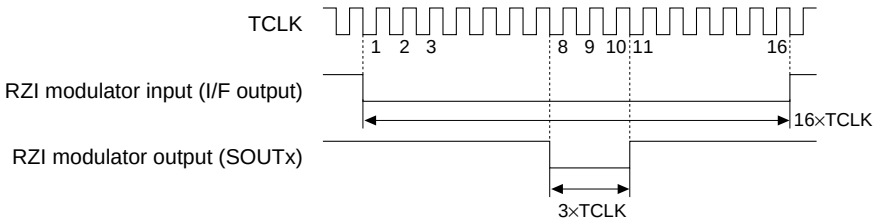


Figure III.9.16 Data Modulation by RZI Circuit

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.

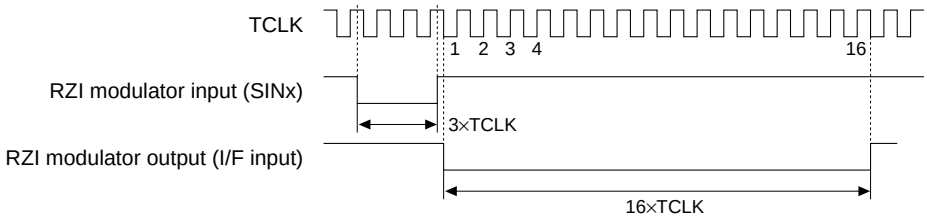


Figure III.9.17 Demodulation by RZI Circuit

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.9.17 shows the input signal as a low pulse of a 3×TCLK 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 of 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	0x0 to 0xFF(0x7F)	X	R/W	7-bit asynchronous mode does not use TXD07.
		D6	**TXD06**	TXD07(06) = MSB		X		
		D5	**TXD05**	TXD00 = LSB		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	0x0 to 0xFF(0x7F)	X	R	7-bit asynchronous mode does not use RXD07 (fixed at 0).
		D6	**RXD06**	RXD07(06) = MSB		X		
		D5	**RXD05**	RXD00 = LSB		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	**–**	–	–	–	–	0 when being read.
		D5	**TEND0**	Ch.0 transmit-completion flag	1 Transmitting 0 End	0	R	
		D4	**FER0**	Ch.0 framing 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	**SSCK0**	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	0x0 to 0xFF(0x7F)	X	R/W	7-bit asynchronous mode does not use TXD17.
		D6	**TXD16**	TXD17(16) = MSB		X		
		D5	**TXD15**	TXD10 = LSB		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	0x0 to 0xFF(0x7F)	X	R	7-bit asynchronous mode does not use RXD17 (fixed at 0).
		D6	**RXD16**	RXD17(16) = MSB		X		
		D5	**RXD15**	RXD10 = LSB		X		
		D4	**RXD14**			X		
		D3	**RXD13**			X		
		D2	**RXD12**			X		
		D1	**RXD11**			X		
		D0	**RXD10**			X		

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	**TEND1**	Ch.1 transmit-completion flag	1 Transmitting 0 End	0	R	
		D4	**FER1**	Ch.1 framing 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	
Serial I/F Ch.1 control register	**00401E8 (B)**	D7	**TXEN1**	Ch.1 transmit enable	1 Enabled 0 Disabled	0	R/W	
		D6	**RXEN1**	Ch.1 receive enable	1 Enabled 0 Disabled	0	R/W	
		D5	**EPR1**	Ch.1 parity enable	1 With parity 0 No parity	X	R/W	Valid only in
		D4	**PMD1**	Ch.1 parity mode selection	1 Odd 0 Even	X	R/W	asynchronous mode.
		D3	**STPB1**	Ch.1 stop bit selection	1 2 bits 0 1 bit	X	R/W	
		D2	**SSCK1**	Ch.1 input clock selection	1 #SCLK1 0 Internal clock	X	R/W	
		D1	**SMD11**	Ch.1 transfer mode selection	SMD1[1:0] Transfer mode	X	R/W	
		D0	**SMD10**		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	**DIVMD1**	Ch.1 async. clock division ratio	1 1/8 0 1/16	X	R/W	
		D3	**IRTL1**	Ch.1 IrDA I/F output logic inversion	1 Inverted 0 Direct	X	R/W	Valid only in
		D2	**IRRL1**	Ch.1 IrDA I/F input logic inversion	1 Inverted 0 Direct	X	R/W	asynchronous mode.
		D1	**IRMD11**	Ch.1 interface mode selection	IRMD1[1:0] I/F mode	X	R/W	
		D0	**IRMD10**		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	**TXD27**	Serial I/F Ch.2 transmit data	0x0 to 0xFF(0x7F)	X	R/W	
		D6	**TXD26**	TXD27(26) = MSB		X		
		D5	**TXD25**	TXD20 = LSB		X		
		D4	**TXD24**			X		
		D3	**TXD23**			X		
		D2	**TXD22**			X		
		D1	**TXD21**			X		
		D0	**TXD20**			X		
Serial I/F Ch.2 receive data register	**00401F1 (B)**	D7	**RXD27**	Serial I/F Ch.2 receive data	0x0 to 0xFF(0x7F)	X	R	
		D6	**RXD26**	RXD27(26) = MSB		X		
		D5	**RXD25**	RXD20 = LSB		X		
		D4	**RXD24**			X		
		D3	**RXD23**			X		
		D2	**RXD22**			X		
		D1	**RXD21**			X		
		D0	**RXD20**			X		
Serial I/F Ch.2 status register	**00401F2 (B)**	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**TEND2**	Ch.2 transmit-completion flag	1 Transmitting 0 End	0	R	
		D4	**FER2**	Ch.2 framing error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.
		D3	**PER2**	Ch.2 parity error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.
		D2	**OER2**	Ch.2 overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.
		D1	**TDBE2**	Ch.2 transmit data buffer empty	1 Empty 0 Buffer full	1	R	
		D0	**RDBF2**	Ch.2 receive data buffer full	1 Buffer full 0 Empty	0	R	
Serial I/F Ch.2 control register	**00401F3 (B)**	D7	**TXEN2**	Ch.2 transmit enable	1 Enabled 0 Disabled	0	R/W	
		D6	**RXEN2**	Ch.2 receive enable	1 Enabled 0 Disabled	0	R/W	
		D5	**EPR2**	Ch.2 parity enable	1 With parity 0 No parity	X	R/W	Valid only in
		D4	**PMD2**	Ch.2 parity mode selection	1 Odd 0 Even	X	R/W	asynchronous mode.
		D3	**STPB2**	Ch.2 stop bit selection	1 2 bits 0 1 bit	X	R/W	
		D2	**SSCK2**	Ch.2 input clock selection	1 #SCLK2 0 Internal clock	X	R/W	
		D1	**SMD21**	Ch.2 transfer mode selection	SMD2[1:0] Transfer mode	X	R/W	
		D0	**SMD20**		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	**DIVMD2**	Ch.2 async. clock division ratio	1 1/8 0 1/16	X	R/W	
		D3	**IRTL2**	Ch.2 IrDA I/F output logic inversion	1 Inverted 0 Direct	X	R/W	Valid only in
		D2	**IRRL2**	Ch.2 IrDA I/F input logic inversion	1 Inverted 0 Direct	X	R/W	asynchronous mode.
		D1	**IRMD21**	Ch.2 interface mode selection	IRMD2[1:0] I/F mode	X	R/W	
		D0	**IRMD20**		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	R	
		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	–	–	–	0 when being read.
		D5	**TEND3**	Ch.3 transmit-completion flag	1 Transmitting 0 End	0	R	
		D4	**FER3**	Ch.3 framing 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	0 to 7	X	R/W	
		D5	**PSIO01**	interrupt level		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	0 to 7	X	R/W	
		D1	**PSIO11**	interrupt level		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	0 to 7	X	R/W	
		D5	**PSIO31**	interrupt level		X		
		D4	**PSIO30**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PSIO22**	Serial interface Ch.2	0 to 7	X	R/W	
		D1	**PSIO21**	interrupt level		X		
		D0	**PSIO20**			X		

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	**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	
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 port s 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 TXENx, 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, TXDx7 (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 TXDx 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), RDBFx 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 RXDx7.
The serial data input from the SINx 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 RXDx 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

TENDx 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, TENDx maintains "1" until all data is transmitted (see Figure III.9.4 and Figure III.9.12). In clock-synchronized slave mode, TENDx goes "0" every time 1-byte data is transmitted (see Figure III.9.5).
At initial reset, TENDx 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.

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).

FSERR0, FSRX0, FSTX0: Ch.0 interrupt factor flags
(D0,D1,D2) / Serial I/F Ch.0/1 interrupt factor flag register (0x40286)
FSERR1, FSRX1, FSTX1: Ch.1 interrupt factor flags
(D3,D4,D5) / Serial I/F Ch.0/1 interrupt factor flag register (0x40286)
FSERR2, FSRX2, FSTX2: Ch.2 interrupt factor flags
(D0,D1,D2) / Serial I/F Ch.2/3 interrupt factor flag register (0x40289)
FSERR3, FSRX3, FSTX3: 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).

Programming Notes

(1) Before setting various serial-interface parameters, make sure the transmit and receive operations are disabled (TXENx = RXENx = "0").

(2) When the serial interface is transmitting or receiving data, do not set TXENx or RXENx 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, RXENx and TXENx 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.

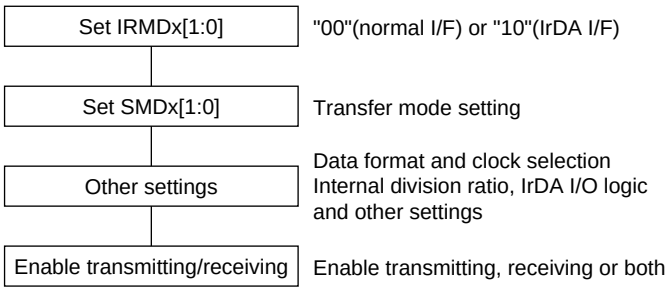


Figure III.9.18 Serial Interface Initialize Procedure

(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 IRRLx = "0"; falling edge when IRRLx = "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 S1C33L05 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 been detected.

  Asynchronous mode
  - Data length: 7 or 8 bits, selectable
  - Receive error: Overrun, framing, or parity errors can been 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.

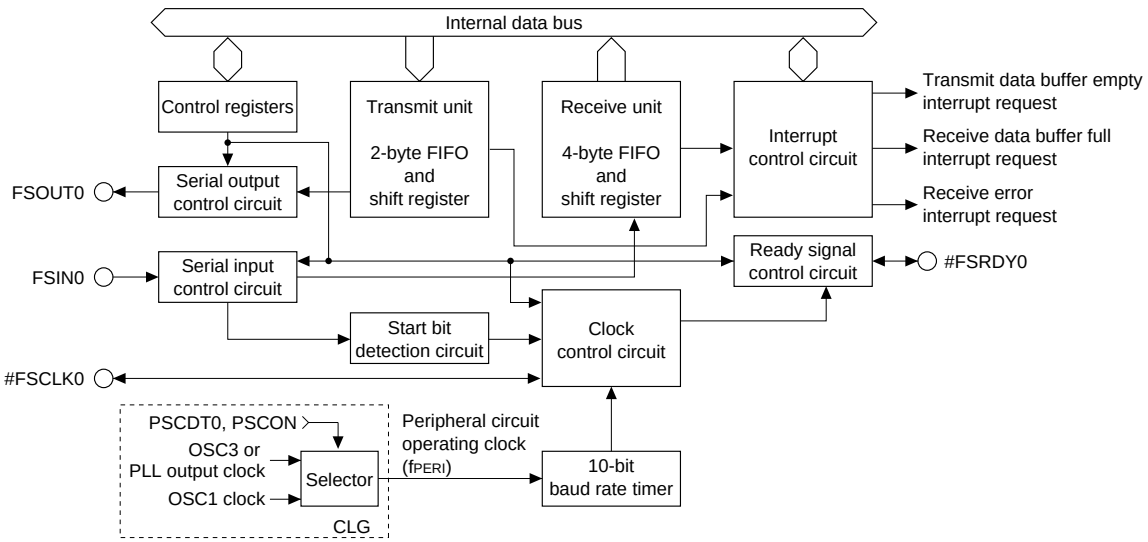


Figure III.10.1 Configuration of Serial Interface

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/**FSIN0**)	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/**FSOUT0**)	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/**#FSCLK0**)	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/**#FSRDY0**)	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)
   The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.

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.

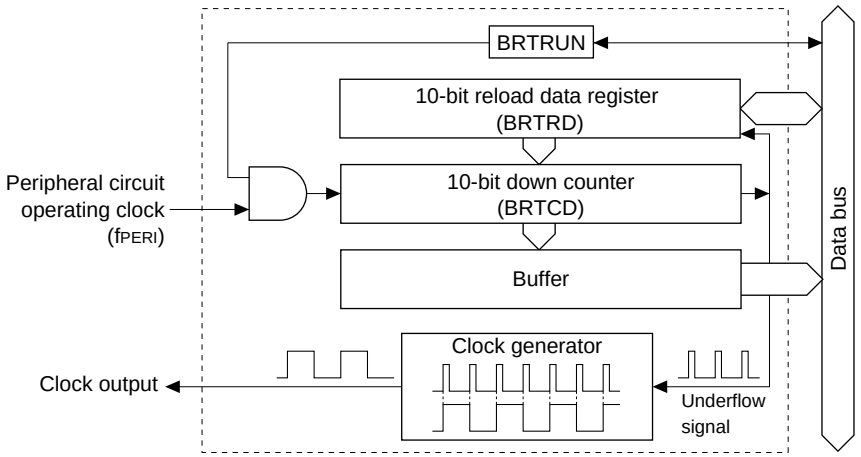


Figure III.10.2 Baud-Rate Timer

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 operading 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

$$BRTRD = \frac{f_{PERI}}{2 \times bps} - 1 \qquad \text{(Eq. 1)}$$

BRTRD: Reload data register setup value of the baud-rate timer
f_{PERI}: C33 peripheral circuit operating clock frequency (Hz)
bps: Transfer rate (bits/second)

Asynchronous mode

$$BRTRD = \frac{f_{PERI} \times DIVMD}{2 \times bps} - 1 \qquad \text{(Eq. 2)}$$

BRTRD: Reload data register setup value of the baud-rate timer
f_{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	f_{PERI} = 20 MHz		f_{PERI} = 25 MHz		f_{PERI} = 33 MHz	
(bps)	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:

$$Error = \{\frac{f_{PERI} \times DIVMD}{(BRTRD + 1) \times 2 \times bps} - 1\} \times 100\ [\%]$$

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.

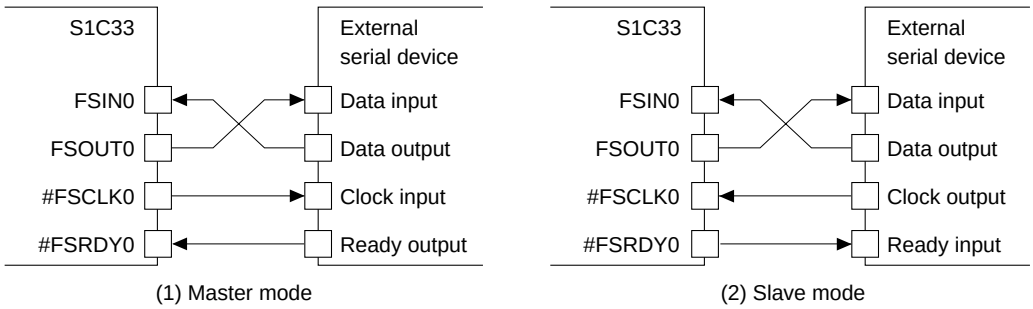


Figure III.10.3 Example of Connection in Clock-Synchronized Mode

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

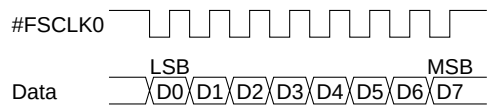


Figure III.10.4 Clock-Synchronized Transfer Data Format

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.

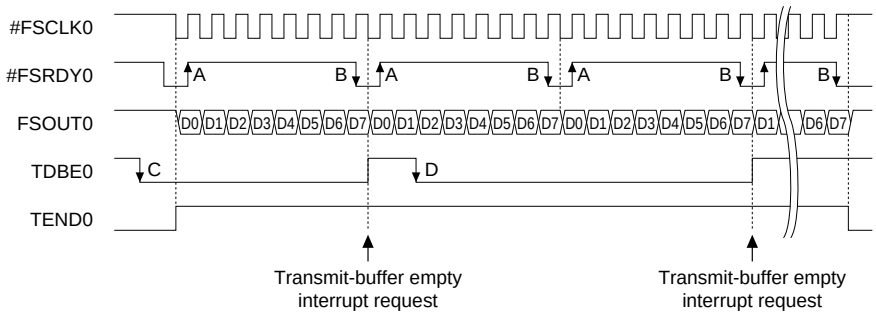


Figure III.10.5 Transmit Timing Chart in Clock-Synchronized Master Mode

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.

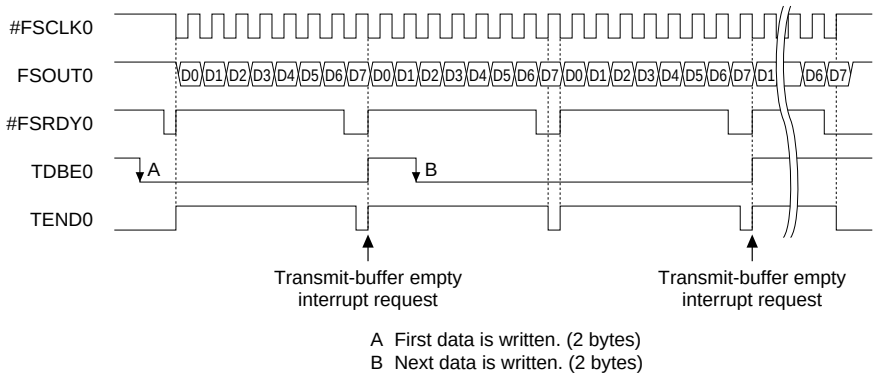


Figure III.10.6 Transmit Timing Chart in Clock-Synchronized Slave Mode

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.

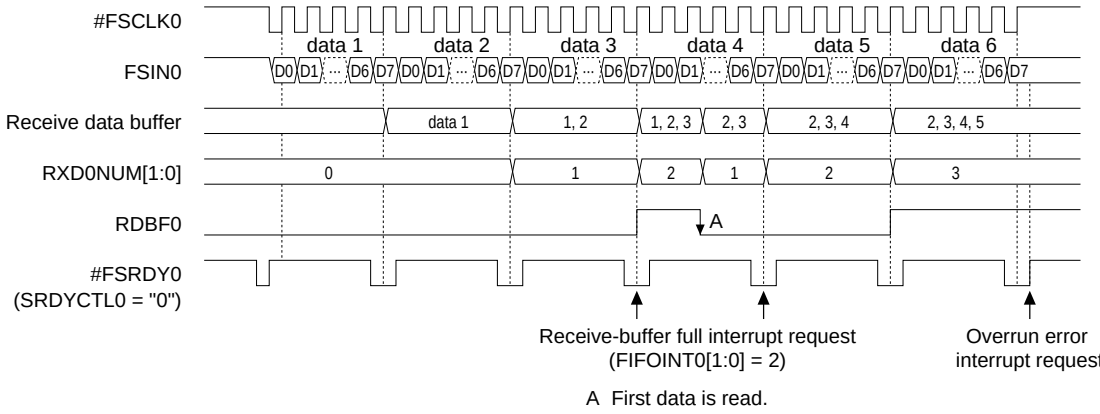
The number of data in the receive data buffer can be checked by reading RXD0NUM[1:0] (D[7:6]) / FIFO serial I/F Ch.0 status register (0x300202). When RXD0NUM[1:0] is 0, the buffer contains 0 or 1 data. When RXD0NUM[1:0] is 1–3, the buffer contains 2–4 data.
Furthermore, the RDBF0 flag (D0) / FIFO serial 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.



A First data is read.

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.

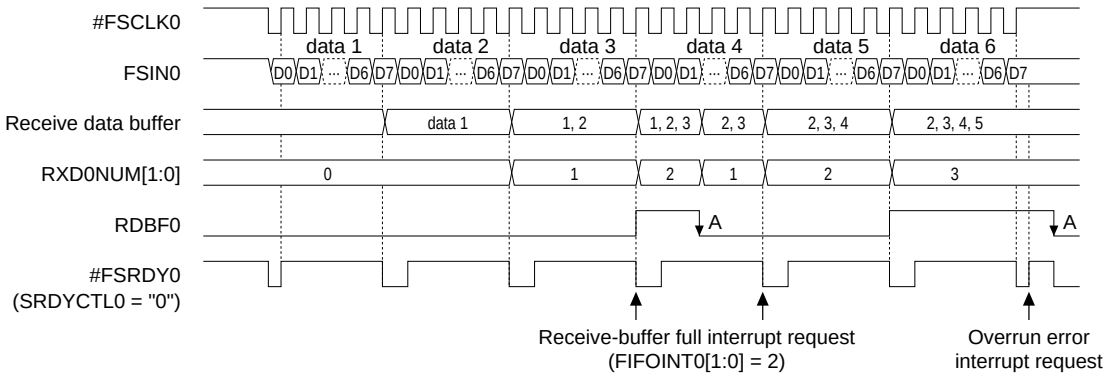


Figure III.10.8 Receive Timing Chart in Clock-Synchronized Slave Mode

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.

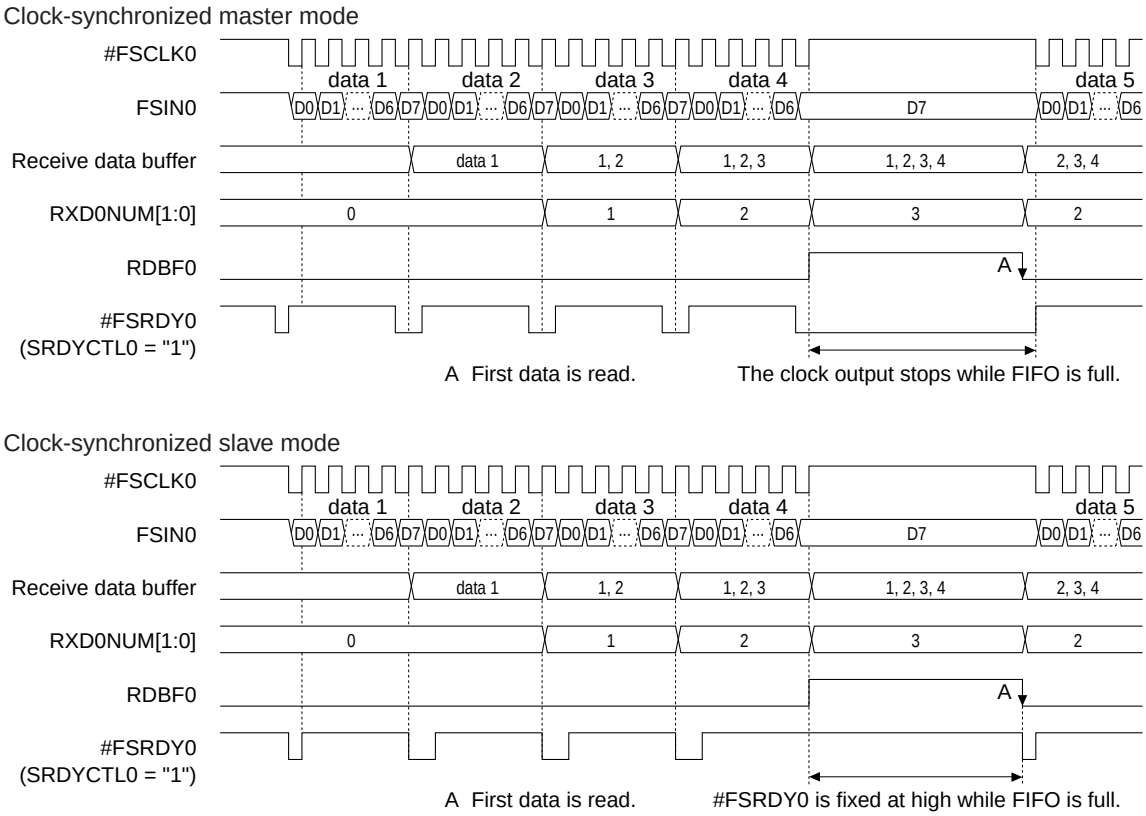


Figure III.10.9 #FSRDY0 High Mask Mode

(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.

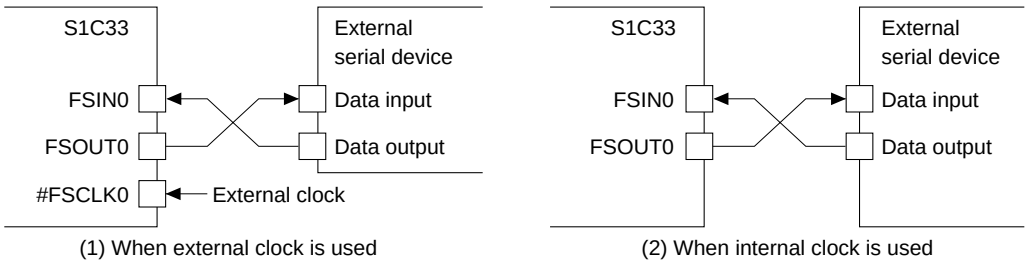


Figure III.10.10 Example of Connection in Asynchronous Mode

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

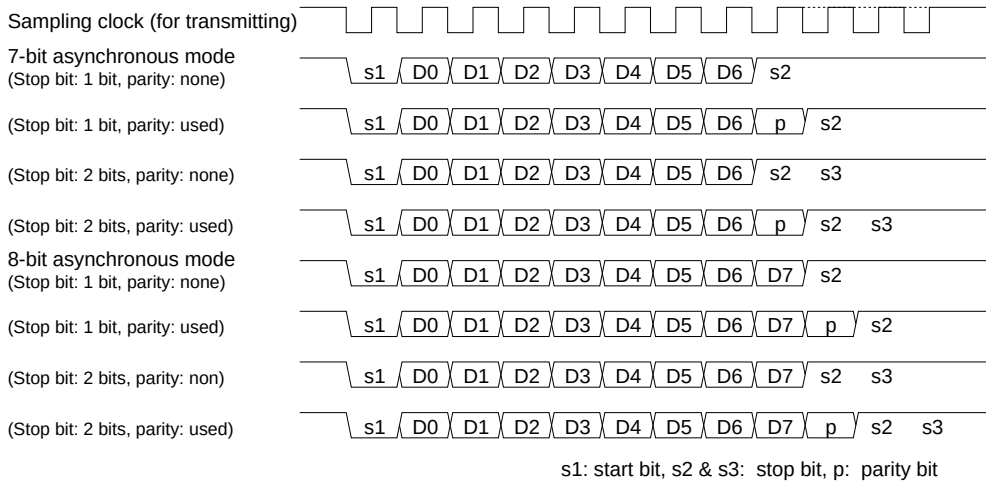


Figure III.10.11 Data Format for Asynchronous Transfer

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

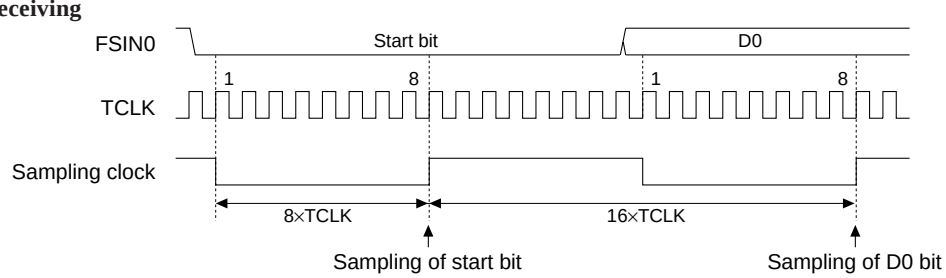


Figure III.10.12 Sampling Timing for Asynchronous Receive Operation (when 1/16 division is selected)

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×TCLK (4×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×TCLK cycles (8×TCLK cycles when 1/8 division is selected).

For transmitting

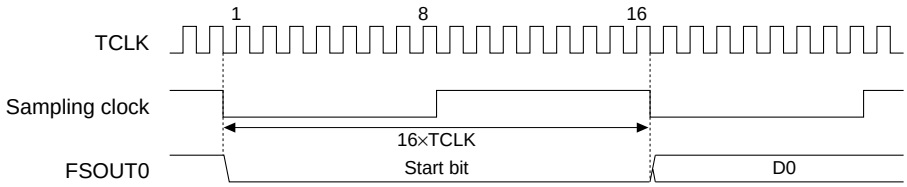


Figure III.10.13 Bit data Output During Asynchronous Transmit Operation (when 1/16 division is selected)

During transmission, each bit data is output from the FSOUT0 pin in 16×TCLK cycles (8×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

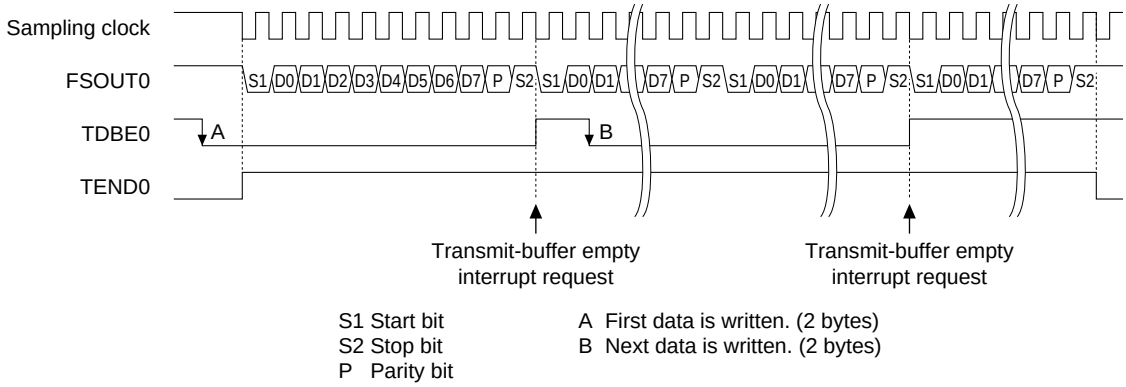


Figure III.10.14 Transmit Timing Chart in Asynchronous Mode

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

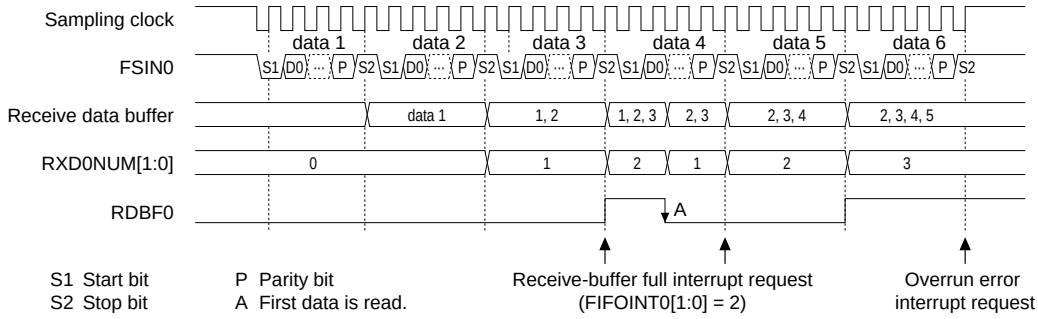


Figure III.10.15 Receive Timing Chart in Asynchronous Mode

1. The serial interface starts sampling when the start bit is input (FSIN0 = 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.

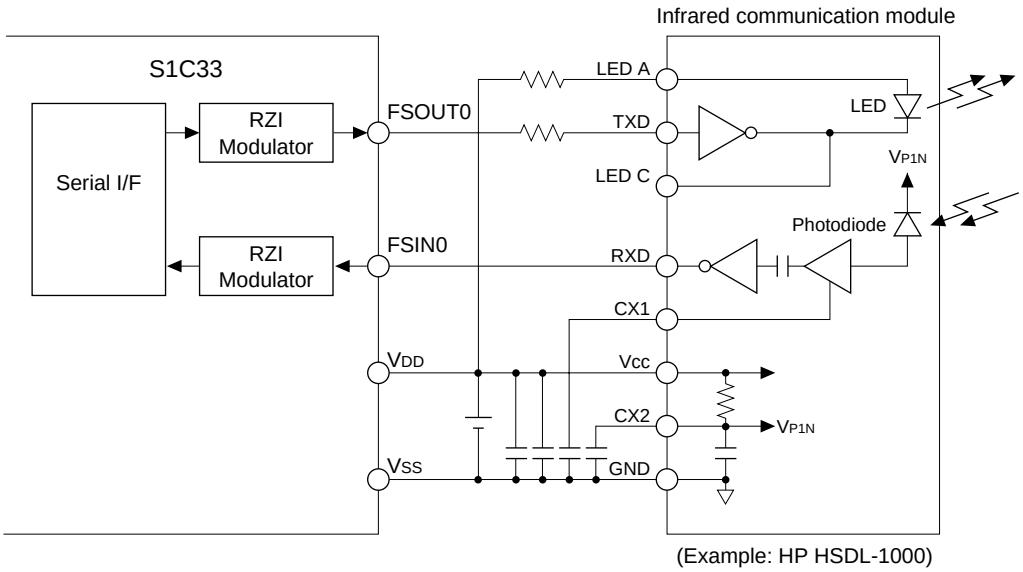


Figure III.10.16 Configuration Example of IrDA Interface

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 IRRL0 (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".

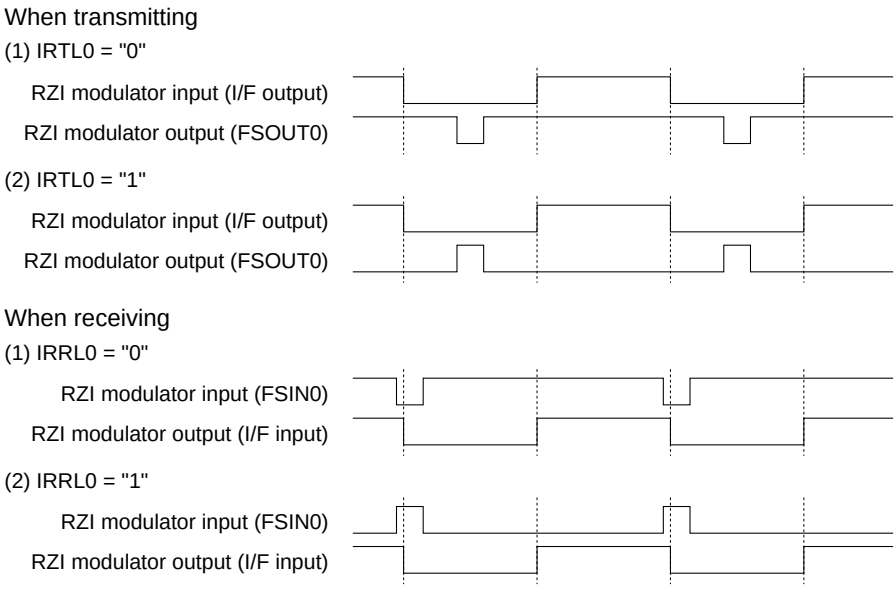


Figure III.10.17 IRRL0 and IRTL0 Settings

Note: The IRRL0 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.

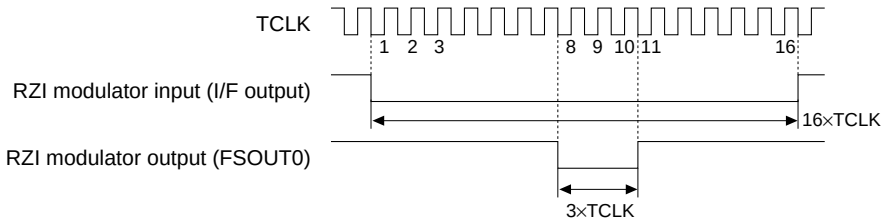


Figure III.10.18 Data Modulation by RZI Circuit

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.

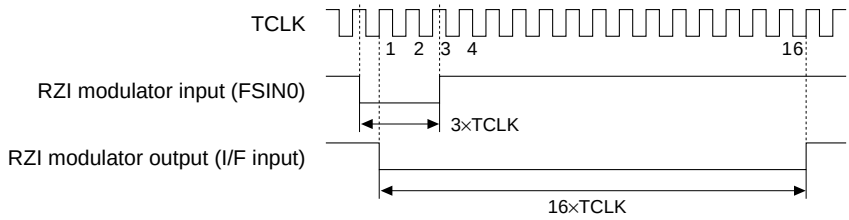


Figure III.10.19 Demodulation by RZI Circuit

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)	EFSERR0 (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	0x0 to 0xFF(0x7F)	X	R/W	7-bit asynchronous mode does not use TXD07.
		D6	**TXD06**	TXD07(06) = MSB		X		
		D5	**TXD05**	TXD00 = LSB		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	0x0 to 0xFF(0x7F)	X	R	7-bit asynchronous mode does not use RXD07 (fixed at 0).
		D6	**RXD06**	RXD07(06) = MSB		X		
		D5	**RXD05**	RXD00 = LSB		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	RXD0NUM[1:0] / Number of data: 1 1 / 4; 1 0 / 3; 0 1 / 2; 0 0 / 1 or 0	0	R/W	
		D6	**RXD0NUM0**	in FIFO		0		
		D5	**TEND0**	Ch.0 transmit-completion flag	1 Transmitting / 0 End	0	R	
		D4	**FER0**	Ch.0 framing 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: 1 1 / 8-bit asynchronous; 1 0 / 7-bit asynchronous; 0 1 / Clock sync. Slave; 0 0 / Clock sync. Master	X	R/W	
		D0	**SMD00**			X		
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: 1 1 / 4; 1 0 / 3; 0 1 / 2; 0 0 / 1	0	R/W	
		D5	**FIFOINT00**			0	R/W	
		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: 1 1 / reserved; 1 0 / IrDA 1.0; 0 1 / reserved; 0 0 / General I/F	X	R/W	
		D0	**IRMD00**			X		
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		
		D5	**BRTRD05**			0		
		D4	**BRTRD04**			0		
		D3	**BRTRD03**			0		
		D2	**BRTRD02**			0		
		D1	**BRTRD01**			0		
		D0	**BRTRD00**			0		

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	**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		
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		
		D5	**BRTCD05**			0		
		D4	**BRTCD04**			0		
		D3	**BRTCD03**			0		
		D2	**BRTCD02**			0		
		D1	**BRTCD01**			0		
		D0	**BRTCD00**			0		
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		
P00–P03 port function extension register	**0300040 (B)**	D7	**EFP031**	P03 port extended function	EFP03[1:0] = 1 * : reserved; 0 1 : #FSRDY0; 0 0 : P03/#SRDY0	0	R/W	
		D6	**EFP030**			0		
		D5	**EFP021**	P02 port extended function	EFP02[1:0] = 1 * : reserved; 0 1 : #FSCLK0; 0 0 : P02/#SCLK0	0	R/W	
		D4	**EFP020**			0		
		D3	**EFP011**	P01 port extended function	EFP01[1:0] = 1 * : reserved; 0 1 : FSOUT0; 0 0 : P01/SOUT0	0	R/W	
		D2	**EFP010**			0		
		D1	**EFP001**	P00 port extended function	EFP00[1:0] = 1 * : reserved; 0 1 : FSIN0; 0 0 : P00/SIN0	0	R/W	
		D0	**EFP000**			0		
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**	interrupt level		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.

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-purpos 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", eceive 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.

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.

IRTL0: 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 IRTL0 is set to "1", a high pulse is output when the output data = "0" (held low-level when the output data = "1"). If IRTL0 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, IRTL0 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.

EFSERR0, 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 EFSERR0, 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).

FFSERR0, 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 FFSERR0, 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, DEFFST0X0: 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.

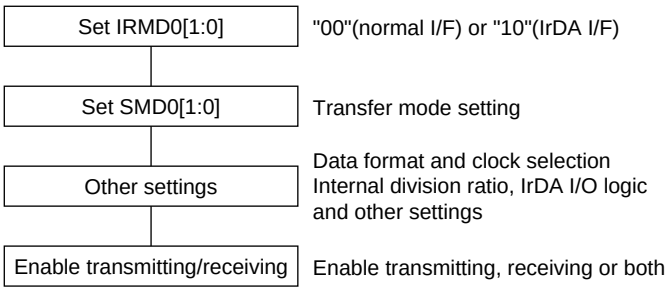


Figure III.10.20 Serial Interface Initialize Procedure

(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.

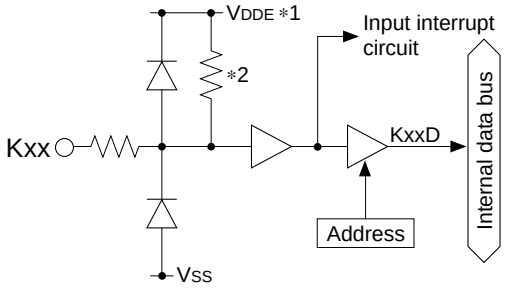
III-11 INPUT/OUTPUT PORTS

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 AVDDE 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 AVDDE. Therefore, when these ports are used as DC input ports, the high level must be AVDDE, and the low level VSS.
If there is a potential difference between AVDDE and VDDE, in particular, if the level from outside is VDDE, a current may flow in the input buffer (when AVDDE > VDDE) or between VDDE and AVDDE (when AVDDE < VDDE). Therefore, if these ports are not used, when the input level is fixed externally, it should be fixed at VSS or AVDDE.
In the model that has built-in pull-up resistors for the K60–K67 ports, the ports are pulled up to AVDDE.

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
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)
K52(#ADTRG)	I	Input port / AD converter trigger	CFK52(D2)/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)
K60(AD0)	I	Input port / AD converter input 0	CFK60(D0)/K6 function select register(0x402C3)
K61(AD1)	I	Input port / AD converter input 1	CFK61(D1)/K6 function select register(0x402C3)
K62(AD2)	I	Input port / AD converter input 2	CFK62(D2)/K6 function select register(0x402C3)
K63(AD3)	I	Input port / AD converter input 3	CFK63(D3)/K6 function select register(0x402C3)
K64(AD4)	I	Input port / AD converter input 4	CFK64(D4)/K6 function select register(0x402C3)
K65(AD5)	I	Input port / AD converter input 5	CFK65(D5)/K6 function select register(0x402C3)
K66(AD6)	I	Input port / AD converter input 6	CFK66(D6)/K6 function select register(0x402C3)
K67(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_{DDE} (power voltage for A/D converter) as their power source. Furthermore, the pull-up resistors of these ports are connected to AV_{DDE}. 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_{DDE} and low level = V_{SS}.
2) When using V_{DDE} as high level similar to other ports, V_{DDE} must be the same voltage level as AV_{DDE}. If the input V_{DDE} level is lower than the AV_{DDE} level, current flows in the input buffer, or if the input V_{DDE} level is higher than the AV_{DDE} level, current flows from the V_{DDE} power supply to the AV_{DDE} power supply.
3) To fix the input level externally when the port is not used, the input pin should be connected to V_{SS} or AV_{DDE}.

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	**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	0	K67	0	R/W	
		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	

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_{DD}) or "0" when the voltage is low (V_{SS}) 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.

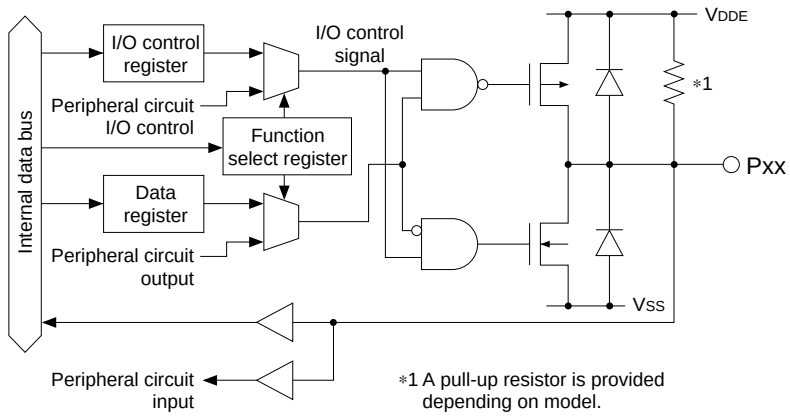


Figure III.11.2 Structure of I/O Port

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_{DDE} level) or "0" when the pin state is low (V_{SS} 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_{DDE} level); when the data is "0", the port outputs a low level (V_{SS} 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	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	
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	–				–	–	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 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	

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	
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
		D6	**IOC26**	P26 I/O control					0	R/W	indicates the values
		D5	**IOC25**	P25 I/O control					0	R/W	of the I/O control
		D4	**IOC24**	P24 I/O control					0	R/W	signals of the ports
		D3	**IOC23**	P23 I/O control					0	R/W	when it is read. (See
		D2	**IOC22**	P22 I/O control					0	R/W	detailed explanation.)
		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	–				–	–	0 when being read.
		D5	**IOC35**	P35 I/O control	1	Output	0	Input	0	R/W	This register
		D4	**IOC34**	P34 I/O control					0	R/W	indicates the values
		D3	**IOC33**	P33 I/O control					0	R/W	of the I/O control
		D2	**IOC32**	P32 I/O control					0	R/W	signals of the ports
		D1	**IOC31**	P31 I/O control					0	R/W	when it is read. (See
		D0	**IOC30**	P30 I/O control					0	R/W	detailed explanation.)
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	

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 (V_{DD} and V_{DDE} level); if the data is "0", the port pin is set low (V_{SS} 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 (V_{DD} and V_{DDE} level), "1" is read out as input data; if the pin voltage is low (V_{SS} 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.

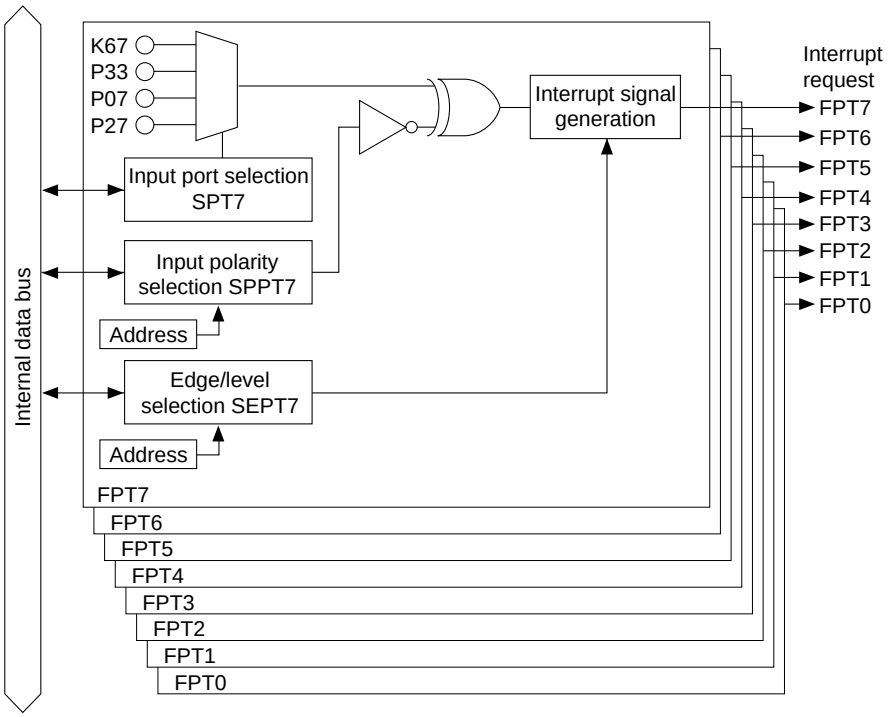


Figure III.11.3 Configuration of Port Input Interrupt Circuit

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.

Furthermore, the signal polarity can be selected using the SPPTx bit of the input polarity 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.

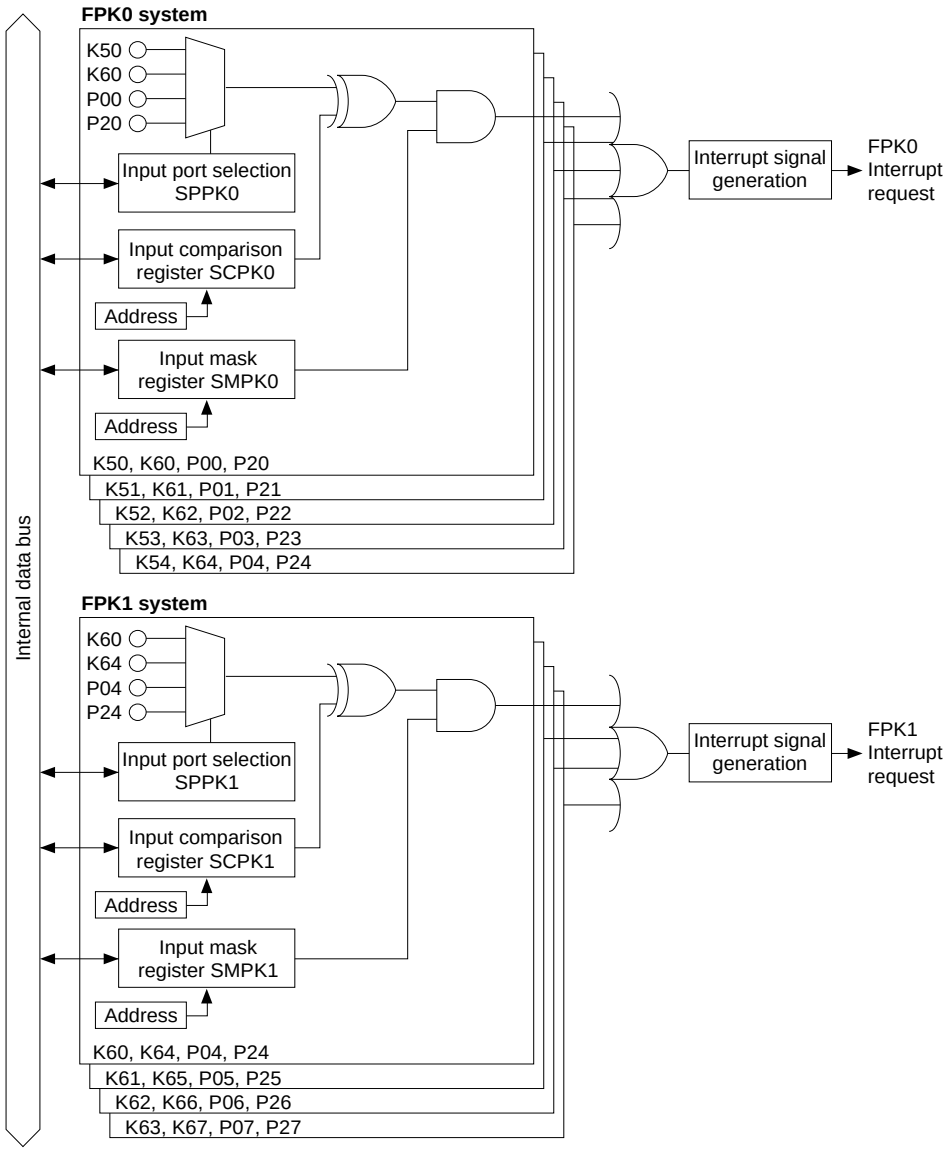


Figure III.11.4 Configuration of Key Input Interrupt Circuit

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 SCPK1[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.

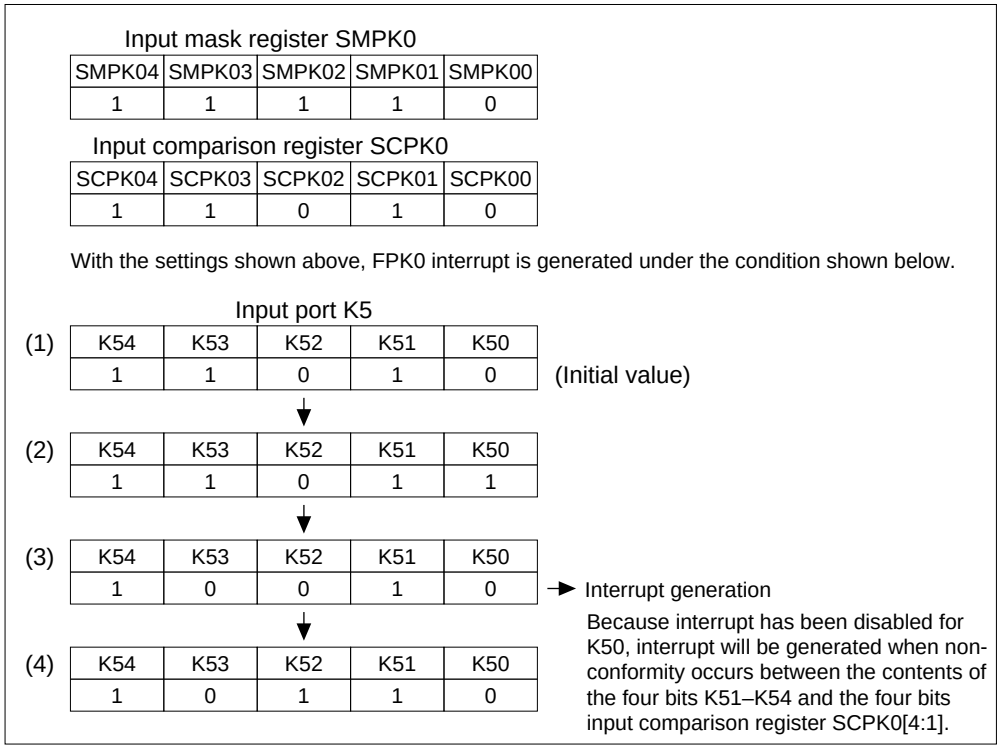


Figure III.11.5 FPK0 Interrupt Generation Example (when K5[4:0] is selected by SPPK[1:0])

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**			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		
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**			X		
		D4	**PP5L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PP4L2**	Port input 4 interrupt level	0 to 7	X	R/W	
		D1	**PP4L1**			X		
		D0	**PP4L0**			X		
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**			X		
		D4	**PP7L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PP6L2**	Port input 6 interrupt level	0 to 7	X	R/W	
		D1	**PP6L1**			X		
		D0	**PP6L0**			X		
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	

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	SPPK11	FPK1 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	–				–	–	0 when being read.
		D4	SMPK04	FPK04 input mask	1	Interrupt	0	Interrupt	0	R/W	
		D3	SMPK03	FPK03 input mask		enabled		disabled	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	0	Interrupt	0	R/W	
		D2	SMPK12	FPK12 input mask		enabled		disabled	0	R/W	
		D1	SMPK11	FPK11 input mask					0	R/W	
		D0	SMPK10	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.

SCPK13–SCPK10: FPK1 input comparison (D[3:0]) / FPK1 input comparison register (0x402CD)
SCPK04–SCPK00: 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

SCPK0[4:0] is compared with the input state of five bits of the FPK0 input ports, and SCPK1[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 1 (P4–P6, EFP)

The C33 STD 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 1

In the C33 STD, I/O ports that can be directed for input or output through the use of a program are extended. 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.

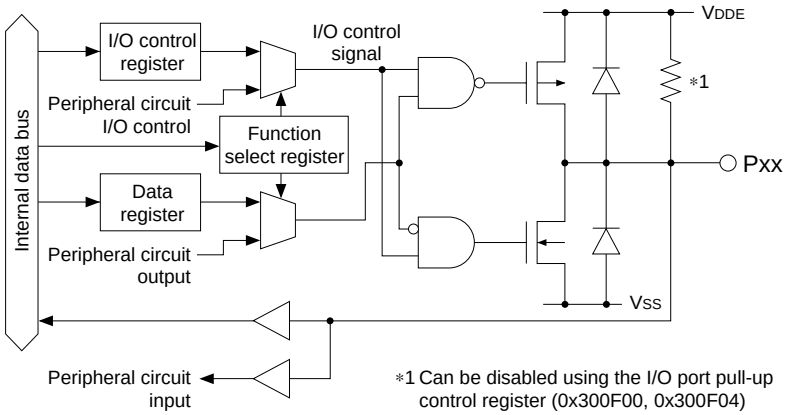


Figure III.12.1 Structure of Extended I/O Port

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 (P4–P6)

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
A25(**P40**)	I/O	Address bus (A25) / I/O port	EFP40[1:0](D[1:0])/P40–P43 port function extension register(0x300048)
A24(**P41**)	I/O	Address bus (A24) / I/O port	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**/#SDCE)	I/O	Area 7 chip enable / Area 7 DRAM row strobe / Area 13 chip enable / Area 13 DRAM row strobe / I/O port / SDRAM chip enable	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/SDCLK)	I/O	Bus clock output / I/O port / OSC1 clock output / SDRAM clock output	EFP60[1:0](D[1:0])/P60–P63 port function extension register(0x30004C)
P61(SDA10)	I/O	I/O port / SDRAM address A10	EFP61[1:0](D[3:2])/P60–P63 port function extension register(0x30004C)
P62(LDQM)	I/O	I/O port / SDRAM lower byte output mask	EFP62[1:0](D[5:4])/P60–P63 port function extension register(0x30004C)
P63(UDQM)	I/O	I/O port / SDRAM upper byte output mask	EFP63[1:0](D[7:6])/P60–P63 port function extension register(0x30004C)

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.
P61–P63 are configured as general-purpose I/O ports. When the SDRAM controller is used, these pins should be configured for the SDRAM signal outputs.
Refer to the "BCU (Bus Control Unit)" section for controlling the bus signals, "Low-speed (OSC1) Oscillation Circuit" section for the FOSC1 output, and "SDRAM Controller" section for the SDRAM signals.
At hot start, the pins retain their state from prior to the reset.

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 (V_{DDE} level) or "0" when the pin state is low (V_{SS} 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_{DDE} level); when the data is "0", the port outputs a low level (V_{SS} 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)
   The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.

I/O Memory of Extended I/O Ports 1

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	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	**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	–	–	–	0 when being read.
		D5	**P55D**	P55 I/O port data	1 High 0 Low	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	–	–	–	0 when being read.
		D5	**IOC55**	P55 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	**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	–	–	–	0 when being read.
		D3	**P63D**	P63 I/O port data	1 High 0 Low	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	–	–	–	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	**EFP031**	P03 port extended function	EFP03[1:0] Function	0	R/W	
		D6	**EFP030**		1 * reserved	0		
					0 1 #FSRDY0			
					0 0 P03/#SRDY0			
		D5	**EFP021**	P02 port extended function	EFP02[1:0] Function	0	R/W	
		D4	**EFP020**		1 * reserved	0		
					0 1 #FSCLK0			
					0 0 P02/#SCLK0			
		D3	**EFP011**	P01 port extended function	EFP01[1:0] Function	0	R/W	
		D2	**EFP010**		1 * reserved	0		
					0 1 FSOUT0			
					0 0 P01/SOUT0			
		D1	**EFP001**	P00 port extended function	EFP00[1:0] Function	0	R/W	
		D0	**EFP000**		1 * reserved	0		
					0 1 FSIN0			
					0 0 P00/SIN0			

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
P20–P21 port function extension register	**0300044 (B)**	D7–4	**–**	reserved	–			–	–	0 when being read.
		D3	**EFP211**	P21 port extended function	EFP21[1:0]		Function	0	R/W	
		D2	**EFP210**		1	*	reserved	0		
					0	1	#SDWE			
					0	0	P21/#DWE			
		D1	**EFP201**	P20 port extended function	EFP20[1:0]		Function	0	R/W	
		D0	**EFP200**		1	*	reserved	0		
					0	1	SDCKE			
					0	0	P20/#DRD			
P31–P33 port function extension register	**0300046 (B)**	D7	**EFP331**	P33 port extended function	EFP33[1:0]		Function	0	R/W	
		D6	**EFP330**		1	*	reserved	0		
					0	1	SDI			
					0	0	P33, etc.			
		D5	**EFP321**	P32 port extended function	EFP32[1:0]		Function	0	R/W	
		D4	**EFP320**		1	*	reserved	0		
					0	1	SPICLK			
					0	0	P32, etc.			
		D3	**EFP311**	P31 port extended function	EFP31[1:0]		Function	0	R/W	
		D2	**EFP310**		1	*	reserved	0		
					0	1	SDO			
					0	0	P31, etc.			
		D1–0	**–**	reserved	–			–	–	0 when being read.
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.			
P40–P43 port function extension register	**0300048 (B)**	D7	**EFP431**	P43 port extended function	EFP43[1:0]		Function	0	R/W	
		D6	**EFP430**		1	*	reserved	0		
					0	1	P43			
					0	0	A22			
		D5	**EFP421**	P42 port extended function	EFP42[1:0]		Function	0	R/W	
		D4	**EFP420**		1	*	reserved	0		
					0	1	P42			
					0	0	A23			
		D3	**EFP411**	P41 port extended function	EFP41[1:0]		Function	0	R/W	
		D2	**EFP410**		1	*	reserved	0		
					0	1	P41			
					0	0	A24			
		D1	**EFP401**	P40 port extended function	EFP40[1:0]		Function	0	R/W	
		D0	**EFP400**		1	*	reserved	0		
					0	1	P40			
					0	0	A25			
P44–P47 port function extension register	**0300049 (B)**	D7	**EFP471**	P47 port extended function	EFP47[1:0]		Function	0	R/W	
		D6	**EFP470**		1	*	reserved	0		
					0	1	P47			
					0	0	A18			
		D5	**EFP461**	P46 port extended function	EFP46[1:0]		Function	0	R/W	
		D4	**EFP460**		1	*	reserved	0		
					0	1	P46			
					0	0	A19			
		D3	**EFP451**	P45 port extended function	EFP45[1:0]		Function	0	R/W	
		D2	**EFP450**		1	*	reserved	0		
					0	1	P45			
					0	0	A20			
		D1	**EFP441**	P44 port extended function	EFP44[1:0]		Function	0	R/W	
		D0	**EFP440**		1	*	reserved	0		
					0	1	P44			
					0	0	A21			

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	1	reserved	0		
					1	0	#SDCE			
					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–P63 port function extension register	**030004C (B)**	D7	**EFP631**	P63 port extended function	EFP63[1:0]		Function	0	R/W	
		D6	**EFP630**		1	*	reserved	0		
					0	1	UDQM			
					0	0	P63			
		D5	**EFP621**	P62 port extended function	EFP62[1:0]		Function	0	R/W	
		D4	**EFP620**		1	*	reserved	0		
					0	1	LDQM			
					0	0	P62			
		D3	**EFP611**	P61 port extended function	EFP61[1:0]		Function	0	R/W	
		D2	**EFP610**		1	*	reserved	0		
					0	1	SDA10			
					0	0	P61			
		D1	**EFP601**	P60 port extended function	EFP60[1:0]		Function	0	R/W	
		D0	**EFP600**		1	1	SDCLK	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 (V_{DDE} level); if the data is "0", the port pin is set low (V_{SS} 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 (V_{DDE} level), "1" is read out as input data; if the pin voltage is low (V_{SS} 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)
EFP201–EFP200: P20 port extended function (D[1:0]) / P20–P21 port function extension register (0x300044)
EFP211–EFP210: P21 port extended function (D[3:2]) / P20–P21 port function extension register (0x300044)
EFP311–EFP310: P31 port extended function (D[3:2]) / P31–P33 port function extension register (0x300046)
EFP321–EFP320: P32 port extended function (D[5:4]) / P31–P33 port function extension register (0x300046)
EFP331–EFP330: P33 port extended function (D[7:6]) / P31–P33 port function extension register (0x300046)
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–P63 port function extension register (0x30004C)
EFP611–EFP610: P61 port extended function (D[3:2]) / P60–P63 port function extension register (0x30004C)
EFP621–EFP620: P62 port extended function (D[5:4]) / P60–P63 port function extension register (0x30004C)
EFP631–EFP630: P63 port extended function (D[7:6]) / P60–P63 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]	**P00** SIN0	FSIN0	–	–
P01	EFP01[1:0]	**P01** SOUT0	FSOUT0	–	–
P02	EFP02[1:0]	**P02** #SCLK0	#FSCLK0	–	–
P03	EFP03[1:0]	**P03** #SRDY0	#FSRDY0	–	–
P20	EFP20[1:0]	**P20** #DRD	SDCLK	–	–
P21	EFP21[1:0]	**P21** #DWE #GAAS	#SDWE	–	–
P31	EFP31[1:0]	**P31** #BUSGET #GARD	SDO	–	–
P32	EFP32[1:0]	**P32** #DMAACK0 #SRDY3	SPICLK	–	–
P33	EFP33[1:0]	**P33** #DMAACK1 SIN3	SDI	–	–
P34	EFP34[1:0]	**P34** #BUSREQ #CE6	#SMWE	–	–
P35	EFP35[1:0]	**P35** #BUSACK	#SMRE	–	–
P40	EFP40[1:0]	**A25**	P40	–	–
P41	EFP41[1:0]	**A24**	P41	–	–
P42	EFP42[1:0]	**A23**	P42	–	–
P43	EFP43[1:0]	**A22**	P43	–	–
P44	EFP44[1:0]	**A21**	P44	–	–
P45	EFP45[1:0]	**A20**	P45	–	–
P46	EFP46[1:0]	**A19**	P46	–	–
P47	EFP47[1:0]	**A18**	P47	–	–
P50	EFP50[1:0]	**#CE4** #CE11 #CE11&12	P50	–	–
P51	EFP51[1:0]	**#CE5** #CE15 #CE15&16	P51	–	–
P52	EFP52[1:0]	**#CE6** #CE7&8	P52	–	–
P53	EFP53[1:0]	**#CE7** #RAS0 #CE13 #RAS2	P53	#SDCE	–
P54	EFP54[1:0]	**#CE8** #RAS1 #CE14 #RAS3	P54	–	–
P55	EFP55[1:0]	**#CE9** #CE17 #CE17&18	P55	–	–
P60	EFP60[1:0]	**BCLK**	P60	FOSC1	SDCLK
P61	EFP61[1:0]	**P61**	SDA10	–	–
P62	EFP62[1:0]	**P62**	LDQM	–	–
P63	EFP63[1:0]	**P63**	UDQM	–	–

Default: EFPxx[1:0] = "00"

III-13 EXTENDED PORTS 2 (PA–PD, FP)

The S1C33L05 has 23 extended I/O ports in addition to the ports in the C33 STD. 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 2

In the S1C33L05, I/O ports that can be directed for input or output through the use of a program are extended. The extended ports are PA0 to PA2, PB0 to PB7, PC0 to PC3 and PD0 to PD7.
Figure III.13.1 shows the structure of an extended I/O port.

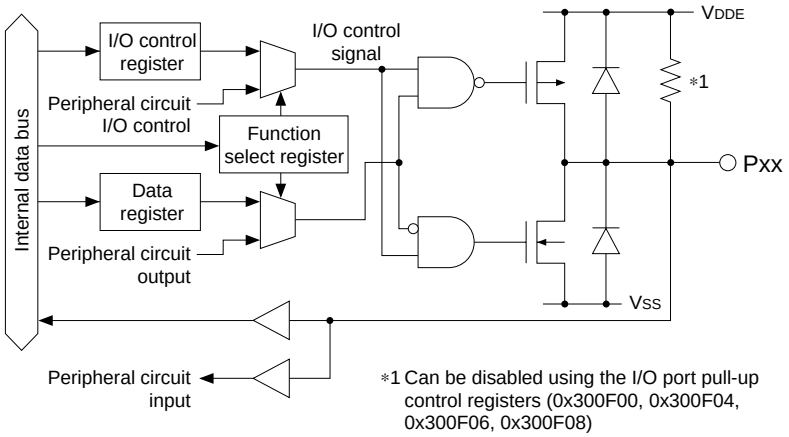


Figure III.13.1 Structure of Extended I/O Port

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 (PA–PD)

The extended I/O ports concurrently serve as the output pins for the SDRAMC, LCDC and sequential ROM interface, and the PA[2:0] ports also serve as the bus control signal output pins for the BCU as shown in Table III.13.1.

Table III.13.1 I/O Pins

Pin name	I/O	Function	Function select bit
#LCAS(**PA0**/#SDCAS)	I/O	DRAM column address strobe (low byte) / I/O port / SDRAM column address strobe	FPA0[1:0](D[1:0])/PA0–PA2 port function extension register(0x300F60)
#HCAS(**PA1**/#SDRAS)	I/O	DRAM column address strobe (high byte) / I/O port / SDRAM row address strobe	FPA1[1:0](D[3:2])/PA0–PA2 port function extension register(0x300F60)
P30(#WAIT/#CE4&5/**PA2**)	I/O	I/O port / Wait cycle request input / Areas 4&5 chip enable / I/O port *	FPA2[1:0](D[5:4])/PA0–PA2 port function extension register(0x300F60)
PB0(FPDAT0)	I/O	I/O port / LCD display data output	FPB0[1:0](D[1:0])/PB0–PB3 port function extension register(0x300F62)
PB1(FPDAT1)	I/O	I/O port / LCD display data output	FPB1[1:0](D[3:2])/PB0–PB3 port function extension register(0x300F62)
PB2(FPDAT2)	I/O	I/O port / LCD display data output	FPB2[1:0](D[5:4])/PB0–PB3 port function extension register(0x300F62)
PB3(FPDAT3)	I/O	I/O port / LCD display data output	FPB3[1:0](D[7:6])/PB0–PB3 port function extension register(0x300F62)
PB4(FPDAT4)	I/O	I/O port / LCD display data output	FPB4[1:0](D[1:0])/PB4–PB7 port function extension register(0x300F63)
PB5(FPDAT5)	I/O	I/O port / LCD display data output	FPB5[1:0](D[3:2])/PB4–PB7 port function extension register(0x300F63)
PB6(FPDAT6)	I/O	I/O port / LCD display data output	FPB6[1:0](D[5:4])/PB4–PB7 port function extension register(0x300F63)
PB7(FPDAT7)	I/O	I/O port / LCD display data output	FPB7[1:0](D[7:6])/PB4–PB7 port function extension register(0x300F63)
PC0(FPFRAME)	I/O	I/O port / LCD vertical scan start pulse output	FPC0[1:0](D[1:0])/PC0–PC3 port function extension register(0x300F64)
PC1(FPLINE)	I/O	I/O port / LCD display data latch clock output	FPC1[1:0](D[3:2])/PC0–PC3 port function extension register(0x300F64)
PC2(FPSHIFT)	I/O	I/O port / LCD display data shift clock output	FPC2[1:0](D[5:4])/PC0–PC3 port function extension register(0x300F64)
PC3(DRDY)	I/O	I/O port / LCD backplane bias signal output	FPC3[1:0](D[7:6])/PC0–PC3 port function extension register(0x300F64)
PD0(#SQRD)	I/O	I/O port / Sequential ROM read signal output	FPD0[1:0](D[1:0])/PD0–PD2 port function extension register(0x300F66)
PD1(SQLALE)	I/O	I/O port / Sequential ROM lower byte address latch enable	FPD1[1:0](D[3:2])/PD0–PD2 port function extension register(0x300F66)
PD2(SQUALE)	I/O	I/O port / Sequential ROM upper byte address latch enable	FPD2[1:0](D[5:4])/PD0–PD2 port function extension register(0x300F66)
PD3	I/O	I/O port	–
PD4	I/O	I/O port	–
PD5	I/O	I/O port	–
PD6	I/O	I/O port	–
PD7	I/O	I/O port	–

* When the SDRAM controller or Bus Arbiter is used, the P30 I/O port function is disabled because SWAITE (D0/0x4812E) must be set (enabled). PA2 can be used as a general-purpose I/O port in this case. When the SDRAM controller and Bus Arbiter is not used, the P30 I/O port function can be used because SWAITE (D0/0x4812E) can be cleared (disabled).

At cold start, PB0–PB7, PC0–PC3 and PD0–PD2 are configured as general-purpose I/O ports and PA0–PA2 are configured as the bus control signal output pins. When using the PA[2:0] ports as the SDRAM interface or general-purpose I/O ports, and the PB, PC and PD ports as the LCDC and/or sequential ROM interface, these pins should be configured for the interface using the function extension bit FPxx.
Refer to the "LCD Controller" section for controlling the LCD interface signals and "Sequential ROM Interface" section for the sequential ROM control signals.
At hot start, the pins retain their state from prior to the reset.

PD3–PD7 are exclusively used for general-purpose I/O ports.

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.
PA2–PA0 I/O control: IOCA[2:0] (D[2:0]) / PA I/O control register (0x300F41)
PB7–PB0 I/O control: IOCB[7:0] (D[7:0]) / PB I/O control register (0x300F43)
PC3–PC0 I/O control: IOCC[3:0] (D[3:0]) / PC I/O control register (0x300F45)
PD7–PD0 I/O control: IOCD[7:0] (D[7:0]) / PD I/O control register (0x300F47)

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_{DDE} level) or "0" when the pin state is low (V_{SS} 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_{DDE} level); when the data is "0", the port outputs a low level (V_{SS} 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 PA–PD and FP Registers

The data and I/O control registers of the PA0–PA2, PB0–PB7, PC0–PC3 and PD0–PD2 ports, and the FP registers for selecting extended port functions are mapped into Area 6 addresses 0x300F40 to 0x300F66. 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)
   The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.

I/O Memory of Extended I/O Ports 2

Table III.13.2 shows the control bits of the extended I/O ports.

Table III.13.2 Control Bits of Extended I/O Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PA I/O port data register	0300F40 (B)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	PA2D	PA2 I/O port data	1 High 0 Low	0	R/W	
		D1	PA1D	PA1 I/O port data		0	R/W	
		D0	PA0D	PA0 I/O port data		0	R/W	
PA I/O control register	0300F41 (B)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	IOCA2	PA2 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.
		D1	IOCA1	PA1 I/O control		0	R/W	
		D0	IOCA0	PA0 I/O control		0	R/W	
PB I/O port data register	0300F42 (B)	D7	PB7D	PB7 I/O port data	1 High 0 Low	0	R/W	
		D6	PB6D	PB6 I/O port data		0	R/W	
		D5	PB5D	PB5 I/O port data		0	R/W	
		D4	PB4D	PB4 I/O port data		0	R/W	
		D3	PB3D	PB3 I/O port data		0	R/W	
		D2	PB2D	PB2 I/O port data		0	R/W	
		D1	PB1D	PB1 I/O port data		0	R/W	
		D0	PB0D	PB0 I/O port data		0	R/W	
PB I/O control register	0300F43 (B)	D7	IOCB7	PB7 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.
		D6	IOCB6	PB6 I/O control		0	R/W	
		D5	IOCB5	PB5 I/O control		0	R/W	
		D4	IOCB4	PB4 I/O control		0	R/W	
		D3	IOCB3	PB3 I/O control		0	R/W	
		D2	IOCB2	PB2 I/O control		0	R/W	
		D1	IOCB1	PB1 I/O control		0	R/W	
		D0	IOCB0	PB0 I/O control		0	R/W	
PC I/O port data register	0300F44 (B)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	PC3D	PC3 I/O port data	1 High 0 Low	0	R/W	
		D2	PC2D	PC2 I/O port data		0	R/W	
		D1	PC1D	PC1 I/O port data		0	R/W	
		D0	PC0D	PC0 I/O port data		0	R/W	
PC I/O control register	0300F45 (B)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	IOCC3	PC3 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.
		D2	IOCC2	PC2 I/O control		0	R/W	
		D1	IOCC1	PC1 I/O control		0	R/W	
		D0	IOCC0	PC0 I/O control		0	R/W	
PD I/O port data register	0300F46 (B)	D7	PD7D	PD7 I/O port data	1 High 0 Low	0	R/W	
		D6	PD6D	PD6 I/O port data		0	R/W	
		D5	PD5D	PD5 I/O port data		0	R/W	
		D4	PD4D	PD4 I/O port data		0	R/W	
		D3	PD3D	PD3 I/O port data		0	R/W	
		D2	PD2D	PD2 I/O port data		0	R/W	
		D1	PD1D	PD1 I/O port data		0	R/W	
		D0	PD0D	PD0 I/O port data		0	R/W	
PD I/O control register	0300F47 (B)	D7	IOCD7	PD7 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.
		D6	IOCD6	PD6 I/O control		0	R/W	
		D5	IOCD5	PD5 I/O control		0	R/W	
		D4	IOCD4	PD4 I/O control		0	R/W	
		D3	IOCD3	PD3 I/O control		0	R/W	
		D2	IOCD2	PD2 I/O control		0	R/W	
		D1	IOCD1	PD1 I/O control		0	R/W	
		D0	IOCD0	PD0 I/O control		0	R/W	
PA0–PA2 port function extension register	0300F60 (B)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	FPA21	PA2 port extended function	FPA2[1:0] Function	0	R/W	
		D4	FPA20		1 * reserved	0		
					0 1 PA2			
					0 0 P30, etc.			
		D3	FPA11	PA1 port extended function	FPA1[1:0] Function	0	R/W	
		D2	FPA10		1 1 reserved	0		
					1 0 #SDRAS			
					0 1 PA1			
					0 0 #HCAS			
		D1	FPA01	PA0 port extended function	FPA0[1:0] Function	0	R/W	
		D0	FPA00		1 1 reserved	0		
					1 0 #SDCAS			
					0 1 PA0			
					0 0 #LCAS			

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
PB0–PB3 port function extension register	**0300F62 (B)**	D7	**FPB31**	PB3 port extended function	FPB3[1:0]		Function	0	R/W	
		D6	**FPB30**		1	*	reserved	0		
					0	1	FPDAT3			
					0	0	PB3			
		D5	**FPB21**	PB2 port extended function	FPB2[1:0]		Function	0	R/W	
		D4	**FPB20**		1	*	reserved	0		
					0	1	FPDAT2			
					0	0	PB2			
		D3	**FPB11**	PB1 port extended function	FPB1[1:0]		Function	0	R/W	
		D2	**FPB10**		1	*	reserved	0		
					0	1	FPDAT1			
					0	0	PB1			
		D1	**FPB01**	PB0 port extended function	FPB0[1:0]		Function	0	R/W	
		D0	**FPB00**		1	*	reserved	0		
					0	1	FPDAT0			
					0	0	PB0			
PB4–PB7 port function extension register	**0300F63 (B)**	D7	**FPB71**	PB7 port extended function	FPB7[1:0]		Function	0	R/W	
		D6	**FPB70**		1	*	reserved	0		
					0	1	FPDAT7			
					0	0	PB7			
		D5	**FPB61**	PB6 port extended function	FPB6[1:0]		Function	0	R/W	
		D4	**FPB60**		1	*	reserved	0		
					0	1	FPDAT6			
					0	0	PB6			
		D3	**FPB51**	PB5 port extended function	FPB5[1:0]		Function	0	R/W	
		D2	**FPB50**		1	*	reserved	0		
					0	1	FPDAT5			
					0	0	PB5			
		D1	**FPB41**	PB4 port extended function	FPB4[1:0]		Function	0	R/W	
		D0	**FPB40**		1	*	reserved	0		
					0	1	FPDAT4			
					0	0	PB4			
PC0–PC3 port function extension register	**0300F64 (B)**	D7	**FPC31**	PC3 port extended function	FPC3[1:0]		Function	0	R/W	
		D6	**FPC30**		1	*	reserved	0		
					0	1	DRDY			
					0	0	PC3			
		D5	**FPC21**	PC2 port extended function	FPC2[1:0]		Function	0	R/W	
		D4	**FPC20**		1	*	reserved	0		
					0	1	FPSHIFT			
					0	0	PC2			
		D3	**FPC11**	PC1 port extended function	FPC1[1:0]		Function	0	R/W	
		D2	**FPC10**		1	*	reserved	0		
					0	1	FPLINE			
					0	0	PC1			
		D1	**FPC01**	PC0 port extended function	FPC0[1:0]		Function	0	R/W	
		D0	**FPC00**		1	*	reserved	0		
					0	1	FPFRAME			
					0	0	PC0			
PD0–PD2 port function extension register	**0300F66 (B)**	D7–6	**–**	reserved	–			–	–	0 when being read.
		D5	**FPD21**	PD2 port extended function	FPD2[1:0]		Function	0	R/W	
		D4	**FPD20**		1	*	reserved	0		
					0	1	SQUALE			
					0	0	PD2			
		D3	**FPD11**	PD1 port extended function	FPD1[1:0]		Function	0	R/W	
		D2	**FPD10**		1	*	reserved	0		
					0	1	SQLALE			
					0	0	PD1			
		D1	**FPD01**	PD0 port extended function	FPD0[1:0]		Function	0	R/W	
		D0	**FPD00**		1	*	reserved	0		
					0	1	#SQRD			
					0	0	PD0			

PA2D–PA0D: PA[2:0] I/O port data (D[2:0]) / PA I/O port data register (0x300F40)
PB7D–PB0D: PB[7:0] I/O port data (D[7:0]) / PB I/O port data register (0x300F42)
PC3D–PC0D: PC[3:0] I/O port data (D[3:0]) / PC I/O port data register (0x300F44)
PD7D–PD0D: PD[7:0] I/O port data (D[7:0]) / PD I/O port data register (0x300F46)

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 (V_{DDE} level); if the data is "0", the port pin is set low (V_{SS} 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 (V_{DDE} level), "1" is read out as input data; if the pin voltage is low (V_{SS} 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.

IOCA2–IOCA0: PA[2:0] port I/O control (D[2:0]) / PA I/O control register (0x300F41)
IOCB7–IOCB0: PB[7:0] port I/O control (D[7:0]) / PB I/O control register (0x300F43)
IOCC3–IOCC0: PC[3:0] port I/O control (D[3:0]) / PC I/O control register (0x300F45)
IOCD7–IOCD0: PD[7:0] port I/O control (D[7:0]) / PD I/O control register (0x300F47)

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 FP register, the value written to the IOC register is read out as is. When another function is selected, the read value may not indicate the value written to the IOC register.

FPA01–FPA00: PA0 port extended function (D[1:0]) / PA0–PA2 port function extension register (0x300F60)
FPA11–FPA10: PA1 port extended function (D[3:2]) / PA0–PA2 port function extension register (0x300F60)
FPA21–FPA20: PA2 port extended function (D[5:4]) / PA0–PA2 port function extension register (0x300F60)
FPB01–FPB00: PB0 port extended function (D[1:0]) / PB0–PB3 port function extension register (0x300F62)
FPB11–FPB10: PB1 port extended function (D[3:2]) / PB0–PB3 port function extension register (0x300F62)
FPB21–FPB20: PB2 port extended function (D[5:4]) / PB0–PB3 port function extension register (0x300F62)
FPB31–FPB30: PB3 port extended function (D[7:6]) / PB0–PB3 port function extension register (0x300F62)
FPB41–FPB40: PB4 port extended function (D[1:0]) / PB4–PB7 port function extension register (0x300F63)
FPB51–FPB50: PB5 port extended function (D[3:2]) / PB4–PB7 port function extension register (0x300F63)
FPB61–FPB60: PB6 port extended function (D[5:4]) / PB4–PB7 port function extension register (0x300F63)
FPB71–FPB70: PB7 port extended function (D[7:6]) / PB4–PB7 port function extension register (0x300F63)
FPC01–FPC00: PC0 port extended function (D[1:0]) / PC0–PC3 port function extension register (0x300F64)
FPC11–FPC10: PC1 port extended function (D[3:2]) / PC0–PC3 port function extension register (0x300F64)
FPC21–FPC20: PC2 port extended function (D[5:4]) / PC0–PC3 port function extension register (0x300F64)
FPC31–FPC30: PC3 port extended function (D[7:6]) / PC0–PC3 port function extension register (0x300F64)
FPD01–FPD00: PD0 port extended function (D[1:0]) / PD0–PD2 port function extension register (0x300F66)
FPD11–FPD10: PD1 port extended function (D[3:2]) / PD0–PD2 port function extension register (0x300F66)
FPD21–FPD20: PD2 port extended function (D[5:4]) / PD0–PD2 port function extension register (0x300F66)

Selects the port extended function. See Table III.13.3 for the port functions that can be selected.
At cold start, FPxx[1:0] is set to "00". At hot start, FPxx[1:0] retains its state from prior to the initial reset.

Table III.13.3 List of Port Functions

Port	Function extension bit	FPxx[1:0] = "00"	FPxx[1:0] = "01"	FPxx[1:0] = "10"	FPxx[1:0] = "11"
PA0	FPA0[1:0]	**#LCAS**	PA0	#SDCAS	–
PA1	FPA1[1:0]	**#HCAS**	PA1	#SDRAS	–
PA2	FPA2[1:0]	**P30** #WAIT #CE4&5	PA2	–	–
PB0	FPB0[1:0]	**PB0**	FPDAT0	–	–
PB1	FPB1[1:0]	**PB1**	FPDAT1	–	–
PB2	FPB2[1:0]	**PB2**	FPDAT2	–	–
PB3	FPB3[1:0]	**PB3**	FPDAT3	–	–
PB4	FPB4[1:0]	**PB4**	FPDAT4	–	–
PB5	FPB5[1:0]	**PB5**	FPDAT5	–	–
PB6	FPB6[1:0]	**PB6**	FPDAT6	–	–
PB7	FPB7[1:0]	**PB7**	FPDAT7	–	–
PC0	FPC0[1:0]	**PC0**	FPFRAME	–	–
PC1	FPC1[1:0]	**PC1**	FPLINE	–	–
PC2	FPC2[1:0]	**PC2**	FPSHIFT	–	–
PC3	FPC3[1:0]	**PC3**	DRDY	–	–
PD0	FPD0[1:0]	**PD0**	#SQRD	–	–
PD1	FPD1[1:0]	**PD1**	SQLALE	–	–
PD2	FPD2[1:0]	**PD2**	SQUALE	–	–

Default: FPxx[1:0] = "00"

THIS PAGE IS BLANK.

III-14 NAND FLASH INTERFACE

Configuration of NAND Flash Interface

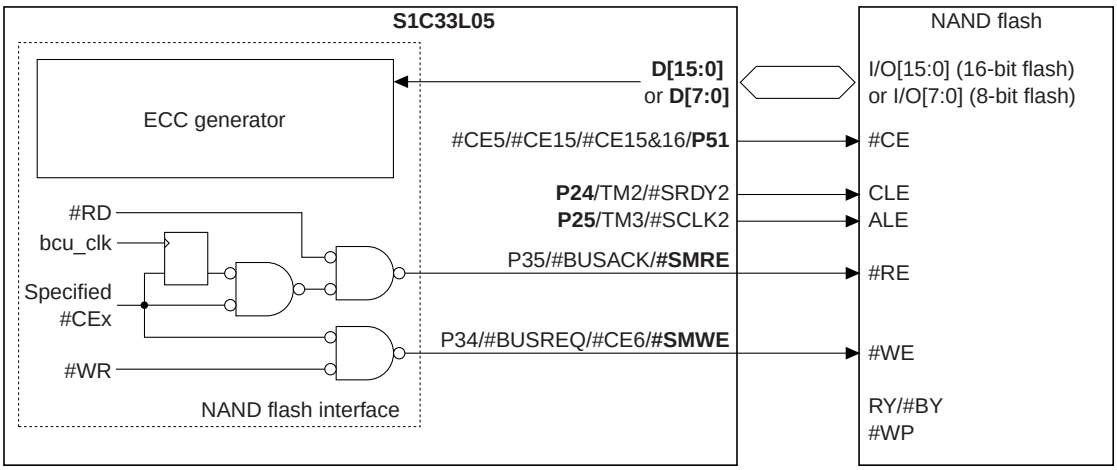
The S1C33L05 contains a NAND flash interface as an extended peripheral circuit.
This circuit supports all functions for 8-bit NAND flash chips, 16-bit NAND flash chips and SmartMedia cards.

Notes: • The SmartMedia card usually has additional pins such as LED, EJECT, LOCK for card use only. If it is necessary to use these SmartMedia card unique pins, they can be connected to I/O port pins of the S1C33L05 and used through software support.

• Since the SmartMedia card has the same functions and timing sequences as NAND flash chips. "NAND flash" will hereafter be used to denote both NAND flash chips and SmartMedia cards in the whole manual.

This circuit generates the following signals for connecting NAND flash to the bus:
#SMWE (write signal for NAND flash)
#SMRE (read signal for NAND flash)

Figure III.14.1 shows the structure of the NAND flash interface and typical connections.



(**BOLD**: Pin functions that should be selected for using the NAND flash I/F.)

Figure III.14.1 NAND flash Interface

This interface incorporates an ECC (Error Correction Code) generator for 8- and 16-bit Flash devices. Furthermore, this interface supports system booting from NAND flash. The current maximum memory size supported by NAND flash booting is 512M × 8 bits or 256M × 16 bits.

Output Pins of NAND Flash Interface

Table III.14.1 shows the pins that are used for the NAND flash interface.

Table III.14.1 Output Pins of NAND flash 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 / NAND flash 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 / NAND flash read signal	EFP35[1:0](D[3:2])/P34–P35 port function extension register(0x300047)
#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)
#CE7(#RAS0/#CE13/#RAS2/P53/ #SDCE)	I/O	Area 7 chip enable / Area 7 DRAM row strobe / Area 13 chip enable / Area 13 DRAM row strobe / I/O port / SDRAM chip enable	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)
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)

* These pins should be used when the NAND flash boot function is used.

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 NAND flash 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 signals, therefore the bus request function cannot be used when they are used as the NAND flash control signal outputs.
The CLE and ALE signals for NAND flash can be output by software control to any I/O ports that are not used for other functions. Note, however, to be sure and use the P24 port for outputting the CLE signal and the P25 port for outputting the ALE signal when using the booting function.
Furthermore, the #CE5 (P51) pin must be used for the #CE signal output when using the booting function.
At hot start, the register retains its status from prior the reset.

Selecting a #CE Area

Use SMCES[1:0] (D[1:0]) / NAND flash I/F #CE area select register (0x300100) to select which #CE signal is assigned for accessing NAND flash. As shown in Table III.14.2, there are four choices, however, the area to be allocated changes according to the CEFUNC[1:0] (D[A:9]/0x48130) setting.

Table III.14.2 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	#CE5	#CE15	#CE15+#CE16

(Default: SMCES = CEFUNC = "00")

The selected #CE signal is ORed with the #WR signal to generate the #SMWE signal. Similarly the #SMRE signal is generated from the selected #CE signal and the #RD signal. In this case, the #CE signal is ORed with the BCU clock before it is ORed with the #RD signal to bring the #SMRE signal into sync with the CPU clock.

Notes: • Area 5 (#CE5) must be used when using the NAND flash-boot function.

- The #CE signal output to NAND flash should be controlled by software after configuring the selected #CE5/7/8/9 pin to the P51/53/54/55 general-purpose I/O pin (output mode). Before switching the pin function to general-purpose output, be sure to set the port data register to "1" in order to avoid unnecessary selection of the NAND flash.

ECC Generator

In order to improve the NAND flash interface reliability, an ECC (Error Correction Code) generator is embedded in this module. The ECC generator generates a 22-bit ECC parity data, which consists of a 6-bit column parity code (CP) and a 16-bit line parity code (LP), for each 256 bytes (in the case of 8-bit devices) or 128 words (in the case of 16-bit devices) of flash data area. Each page of NAND flash consists of two data areas (Area 0 and Area 1), so two 22-bit ECC parity data are generated per page.
When reading data from a NAND flash, software can compare the ECC data read from the NAND flash with the data generated by the ECC generator. It makes it possible to detect two bit errors and correct one bit errors.
When writing data to a NAND flash, software can write the ECC data generated by the ECC generator to the NAND flash redundant area.

ECC algorithm

The flash data is treated as bit streams, and ECC parity data is generated as follows:
'^' denotes the XOR (exclusive OR) operation.

ECC for 8-bit NAND flash device:

```
CP[0] = Din[0] ^ Din[2] ^ Din[4] ^ Din[6] ^ org[0]
CP[1] = Din[1] ^ Din[3] ^ Din[5] ^ Din[7] ^ org[1]
CP[2] = Din[0] ^ Din[1] ^ Din[4] ^ Din[5] ^ org[2]
CP[3] = Din[2] ^ Din[3] ^ Din[6] ^ Din[7] ^ org[3]
CP[4] = Din[0] ^ Din[1] ^ Din[2] ^ Din[3] ^ org[4]
CP[5] = Din[4] ^ Din[5] ^ Din[6] ^ Din[7] ^ org[5]

Dall = Din[0] ^ Din[1] ^ Din[2] ^ Din[3] ^ Din[4] ^ Din[5] ^ Din[6] ^ Din[7]
Count [8:0] = Count [8:0] +1

LP[0]  =  ( ~Count[0] & Dall ) ^ org[0]
LP[2]  =  ( ~Count[1] & Dall ) ^ org[2]
LP[4]  =  ( ~Count[2] & Dall ) ^ org[4]
LP[6]  =  ( ~Count[3] & Dall ) ^ org[6]
LP[8]  =  ( ~Count[4] & Dall ) ^ org[8]
LP[10] =  ( ~Count[5] & Dall ) ^ org[10]
LP[12] =  ( ~Count[6] & Dall ) ^ org[12]
LP[14] =  ( ~Count[7] & Dall ) ^ org[14]
LP[1]  =  (  Count[0] & Dall ) ^ org[1]
LP[3]  =  (  Count[1] & Dall ) ^ org[3]
LP[5]  =  (  Count[2] & Dall ) ^ org[5]
LP[7]  =  (  Count[3] & Dall ) ^ org[7]
LP[9]  =  (  Count[4] & Dall ) ^ org[9]
LP[11] =  (  Count[5] & Dall ) ^ org[11]
LP[13] =  (  Count[6] & Dall ) ^ org[13]
LP[15] =  (  Count[7] & Dall ) ^ org[15]
```

ECC for 16-bit NAND flash device:

```
CP[0] = Din[0] ^ Din[2] ^ Din[4] ^ Din[6] ^ Din[8]  ^ Din[10] ^ Din[12] ^ Din[14] ^ org[0]
CP[1] = Din[1] ^ Din[3] ^ Din[5] ^ Din[7] ^ Din[9]  ^ Din[11] ^ Din[13] ^ Din[15] ^ org[1]
CP[2] = Din[0] ^ Din[1] ^ Din[4] ^ Din[5] ^ Din[8]  ^ Din[9]  ^ Din[12] ^ Din[13] ^ org[2]
CP[3] = Din[2] ^ Din[3] ^ Din[6] ^ Din[7] ^ Din[10] ^ Din[11] ^ Din[14] ^ Din[15] ^ org[3]
CP[4] = Din[0] ^ Din[1] ^ Din[2] ^ Din[3] ^ Din[8]  ^ Din[9]  ^ Din[10] ^ Din[11] ^ org[4]
CP[5] = Din[4] ^ Din[5] ^ Din[6] ^ Din[7] ^ Din[12] ^ Din[13] ^ Din[14] ^ Din[15] ^ org[5]

Dall_low  = Din[0] ^ Din[1] ^ Din[2] ^ Din[3] ^ Din[4] ^ Din[5] ^ Din[6] ^ Din[7]
Dall_high = Din[8] ^ Din[9] ^ Din[10] ^ Din[11] ^ Din[12] ^ Din[13] ^ Din[14] ^ Din[15]
Dall      = Dall_low ^ Dall_high

Count [8:0] = Count [8:0] +2

LP[0]  =  ( Dall_low ) ^ org[0]
LP[2]  =  ( ~Count[1] & Dall ) ^ org[2]
LP[4]  =  ( ~Count[2] & Dall ) ^ org[4]
LP[6]  =  ( ~Count[3] & Dall ) ^ org[6]
LP[8]  =  ( ~Count[4] & Dall ) ^ org[8]
LP[10] =  ( ~Count[5] & Dall ) ^ org[10]
LP[12] =  ( ~Count[6] & Dall ) ^ org[12]
LP[14] =  ( ~Count[7] & Dall ) ^ org[14]
LP[1]  =  (  Dall_high ) ^ org[1]
LP[3]  =  (  Count[1] & Dall ) ^ org[3]
LP[5]  =  (  Count[2] & Dall ) ^ org[5]
LP[7]  =  (  Count[3] & Dall ) ^ org[7]
LP[9]  =  (  Count[4] & Dall ) ^ org[9]
LP[11] =  (  Count[5] & Dall ) ^ org[11]
LP[13] =  (  Count[6] & Dall ) ^ org[13]
LP[15] =  (  Count[7] & Dall ) ^ org[15]
```

Data read flowchart

Figure III.14.2 shows a NAND flash data read flowchart.

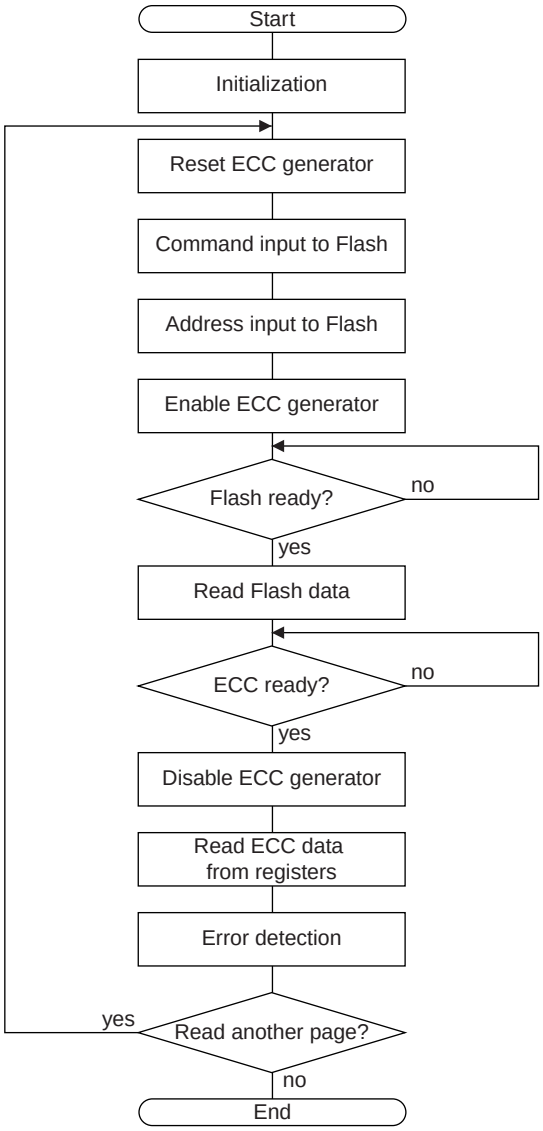


Figure III.14.2 Data Read Flowchart

Notes: • The ECC reset bit (D0/0x300101) should be set before using the ECC function.

• The ECC enable bit (D0/0x300102) should be set only while reading or writing flash data. It should be disabled during command input, address input or status reading.

Data write flowchart

Figure III.14.3 shows a NAND flash data write flowchart.

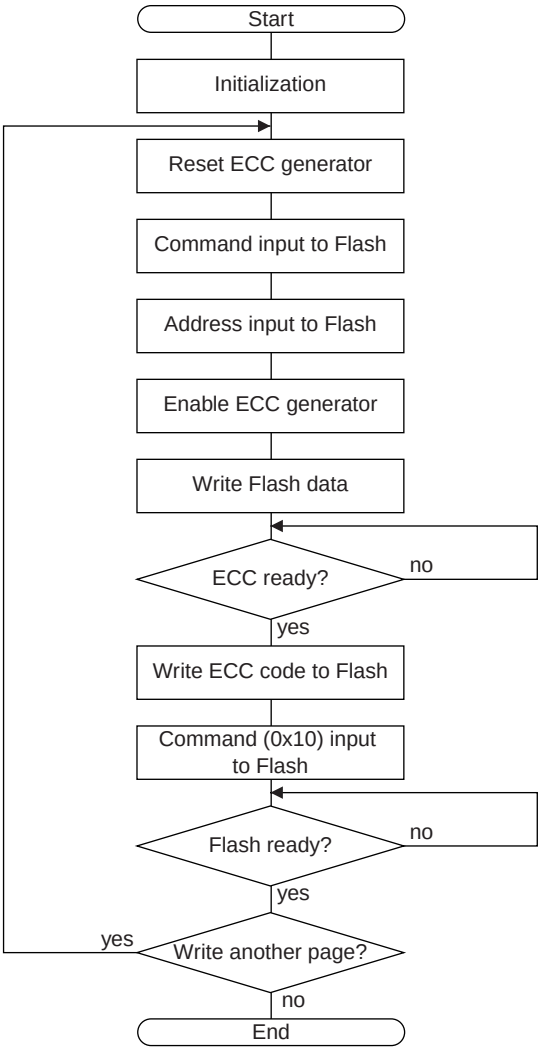


Figure III.14.3 Data Write Flowchart

Notes: • The ECC reset bit (D0/0x300101) should be set before using the ECC function.

• The ECC enable bit (D0/0x300102) should be set only while reading or writing flash data. It should be disabled during command input, address input or status reading.

Booting from NAND Flash

This NAND flash interface supports booting the system from an 8-bit or 16-bit NAND flash. The S1C33L05 has a built-in NAND-flash-boot circuit. This circuit loads 256-byte data, which contains up to 120 instructions except checksum, from the first or second page of a NAND flash to the internal SRAM after an initial reset if the system is set in NAND flash boot mode (the BOOT pin has been set to low). After booting, the CPU jumps to the beginning of the internal SRAM and executes the loaded instructions. The 256-byte data stored in the NAND flash can be modified for the next booting if necessary.

NAND flash boot conditions

To use this boot function, the conditions listed below must be satisfied.

1. SMCODE (D7) / NAND flash I/F #CE area select register (0x300100) is set to "1" (default).
2. The BOOT pin is set to "0" (Vss).
3. Area 5 (#CE5, default) is configured as the NAND flash area.
4. The P24 and P25 ports are used to control the CLE and ALE signals of NAND flash, respectively.

Bootstrap routine

The bootstrap routine located from address 0xc00000 should executed in the following sequence:

1. Initialize the BCU and I/O registers.
2. Reset NAND flash.
3. Read 256-byte data from NAND flash and transfer it to the internal RAM (0x0–0xff).
4. Execute the program from address 0x0 when it is successfully downloaded.
5. Boot the system from Area 10 if downloading failed.

A flowchart and a sample assembly source of NAND flash bootstrap routine are shown below.

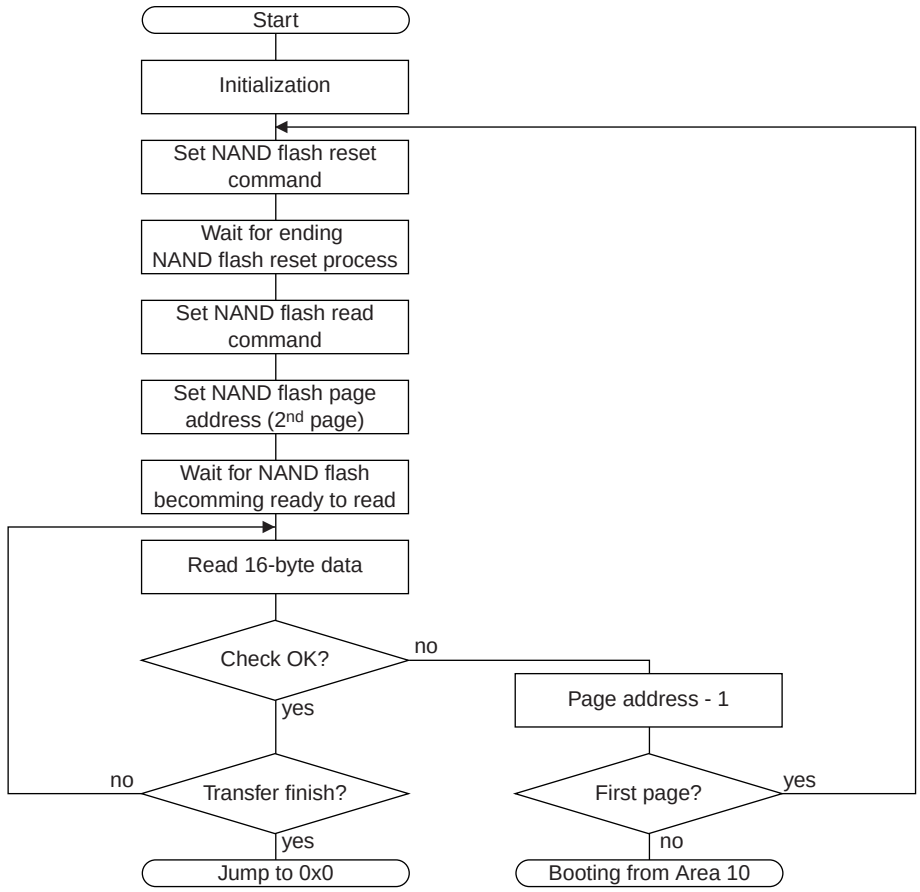


Figure III.14.4 Flowchart for NAND Flash Bootstrap Routine

Sample bootstrap routine

```
        .org  0xc00000
        .half 0x0004                ; reset vector address
        .half 0x00c0
        .org  0xc00004
;//-------------------------------------------------------------------------
;//  BCU & IO configuration registers initialize
;//-------------------------------------------------------------------------
       xld.w      %r8, 0x402d9       ;// r8  -- Port 2 address
       xld.w      %r9, 0x48133       ;// r9  -- Internal / External Access;
       xld.w      %r10, 0x300022     ;// r10 -- P51 data address
       xld.w      %r11, 0x200000     ;// r11 -- area 5 start address

       ld.w       %r0, 0x02
       ld.b       [%r9], %r0         ;// [0x48133] = 0x02 (Internal Access)

;//select P24, P25 as output
       xld.w      %r0, 0x30
       ext        0x01
       ld.b       [%r8], %r0         ;// [0x0402da] = 0x30

;//select P51 data is high
       ld.w       %r1, 0x02
       ld.b       [%r10], %r1        ;// [0x300022] = 0x02
;//select P51 as output
       ext        0x01
       ld.b       [%r10], %r1        ;// [0x300023] = 0x02

;//select P34, P35 function as NAND flash I/F function
       ld.w       %r0, 0x05
       ext        0x25
       ld.b       [%r10], %r0        ;// [0x300047] = 0x05

;//select P51 function as GPIO
       ld.w       %r1, 0x04
       ext        0x28
       ld.b       [%r10], %r1        ;// [0x30004a] = 0x04

;//select NAND flash as area 5
       xld.w      %r0, 0x80
       ext        0xde
       ld.b       [%r10], %r0        ;// [0x300100] = 0x80

;//-------------------------------------------------------------------------
;//  NAND flash reading
;//-------------------------------------------------------------------------
       ld.w       %r0, 0x0           ;// IRAM 0x0 address
       ld.w       %r1, 0x02          ;// %r1 used to make CE5 high
       xld.w      %r2, 0x30
       ld.w       %r4, 0x10
       xld.w      %r5, 0x20
       xld.w      %r9, 0xff
;//-------------------------------------------------------------------------
;//  sent reset command to NAND flash
;//-------------------------------------------------------------------------
       ld.w       %r7, 0x01          ;// when r7=1, CLE command=30h, no use, read 2 pages
                                     ;// when r7=0, 30h command use, read 2pages
read_2_page:
       ld.w       %r6, 0x01
start_reset:
       ld.w       %r0, 0x0
       ld.b       [%r8], %r4         ;// set CLE high
       ld.b       [%r10], %r0        ;// set CE5 low

       ld.h       [%r11], %r9        ;// sent reset command to NAND flash
       ld.b       [%r8], %r0         ;// set CLE low

       xld.w      %r12, 0x2500
delay_500us:
       sub        %r12, 0x01
       jrne       delay_500us
       ld.b       [%r10], %r1        ;// set CE5 high

;//-------------------------------------------------------------------------
;//  start to read 256 bytes from NAND flash
;//-------------------------------------------------------------------------
       ld.b       [%r8], %r4         ;// set CLE high
       ld.b       [%r10], %r0        ;// set CE5 low
       ld.h       [%r11], %r0        ;// sent read command to NAND flash
       ld.b       [%r8], %r0         ;// set CLE low
```

```
        ld.b       [%r8], %r5          ;// set ALE high
        ld.h       [%r11], %r0         ;// sent read address 0
        nop
        ld.h       [%r11], %r6         ;// sent read address 1 (page address)
        nop
        ld.h       [%r11], %r0         ;// sent read address 2
        nop
        ld.h       [%r11], %r0         ;// sent read address 3
        nop
        ld.b       [%r8], %r0          ;// set ALE low

        cmp        %r7, 0x01
        jreq       skip_30h

        ld.b       [%r8], %r4          ;// set CLE high
        ld.h       [%r11], %r2         ;// send read command 30h to NAND flash
        ld.b       [%r8], %r0          ;// set CLE low

skip_30h:
;// wait at least 25 us
       xld.w       %r12, 0x250
delay_25us:
       sub         %r12, 0x01
       jrne        delay_25us

       ld.w        %r15, 0xf           ;// total 15 group
read_one_page:
       ld.w        %r14, 0x0
       ld.w        %r12, 0x10          ;// one group contains 16 + 1 bytes
data_transfer:
       ld.ub       %r13, [%r11]        ;// read data from NAND flash
       add         %r14, %r13          ;// calculate checksum
       ld.b        [%r0]+, %r13
       sub         %r12, 0x01
       jrne        data_transfer
       not         %r14, %r14          ;// SUM Invert
       ld.ub       %r14, %r14          ;// mask MSB
       ld.ub       %r13, [%r11]        ;// read checksum in NAND flash
       cmp         %r13, %r14
       jrne        check_error
       sub         %r15, 0x01
       jrne        read_one_page
       ld.b        [%r10], %r1         ;// set CE5 high
       ld.w        %r0, 0x0            ;// jump to 0x0
       jp.d        %r0                 ;// check ok, jump to 0x0
       nop
check_error:
       ld.b        [%r10], %r1         ;// set CE5 high
       sub         %r6, 0x01           ;// set page address as "0"
       jreq        start_reset
       nop
       sub         %r7, 0x01           ;// Error: use 30h command to read
       jreq        read_2_page

       xld.w       %r3, 0xc000E6       ;// flash_booting address
       ld.w        %r0, 0x0
       ld.w        %r1, 0x8
load_code:
       ld.uh       %r2, [%r3]+
       ld.h        [%r0]+, %r2
       sub         %r1, 0x01           ;// %r1=8
       jrne        load_code

       ld.w        %r0, 0x0            ;// jump to 0x0
       jp.d        %r0
       nop

;;-------------------------------------------------------------------------------------
;flash_booting:
.org 0xc000E6                          ;// 0xc0de start from 7'h73 x2 =230(d)=E6(h)

 .half 0xc0de
 .half 0x34a0
 .half 0xc018
 .half 0xc000
 .half 0x6c00
 .half 0x3001
 .half 0x0781
 .half 0x0000
```

NAND flash data structure

In order to improve the reliability of the NAND flash booting, a checksum method is used for checking data read from NAND flash since the NAND flash booting function does not support ECC generation. The following diagram shows the data structure in NAND flash.

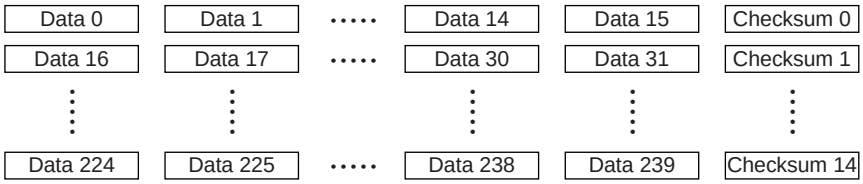


Figure III.14.5 NAND Flash Data Structure

Every 16 bytes of data are checked by one checksum. This means that only 240 bytes (120 instructions) can be used as valid data even though 255 bytes are transferred from the NAND flash to the internal RAM. The checksum is calculated by the following formula:
'+' denotes an arithmetic addition. '~' denotes bit inversion.

```
checksum0  = ~(Data0 + Data1 +     ... + Data14 + Data15)
checksum1  = ~(Data16 + Data17 +   ... + Data30 + Data31)
                          :
checksum14 = ~(Data224 + Data225 + ... + Data238 + Data239)
```

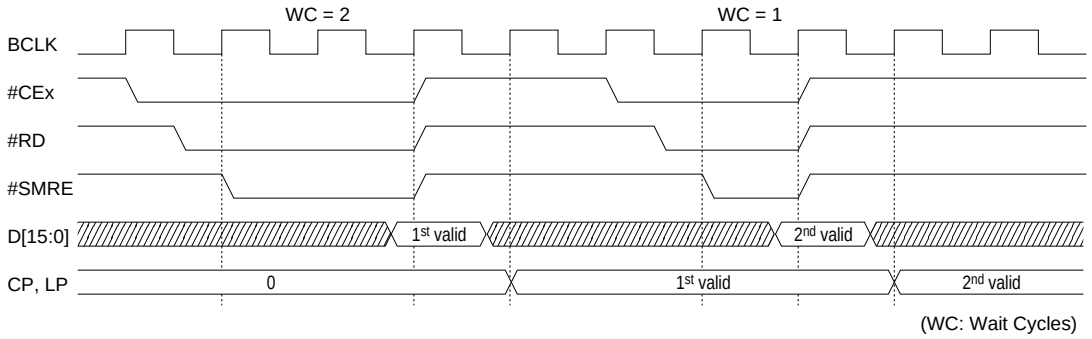
Note: For 16-bit flash devices, the flash I/O[15:8] data will be ignored.

NAND Flash Interface Timings

Data read timing chart

Figure III.14.6 shows the NAND flash data read timing chart.

x1 speed mode



x2 speed mode

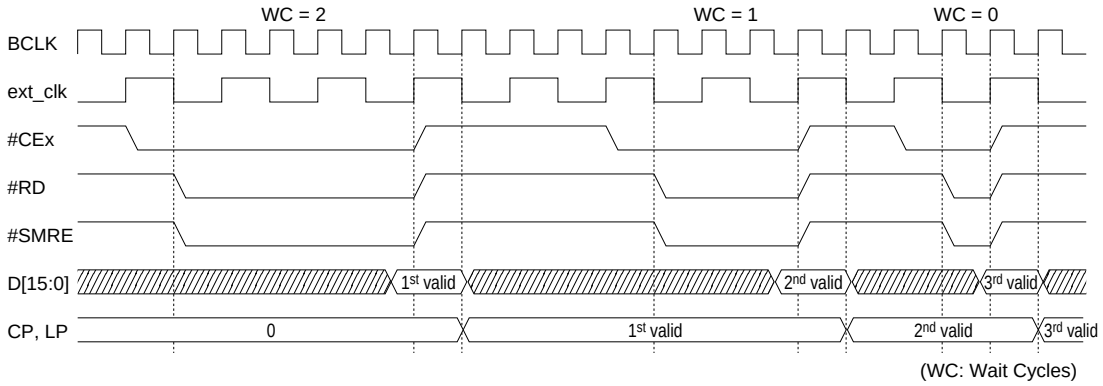


Figure III.14.6 Data Read Timing Chart

Note: In x1 speed mode, at least one wait cycle should be inserted.

Data write timing chart

Figure III.14.7 shows the NAND flash data write timing chart.

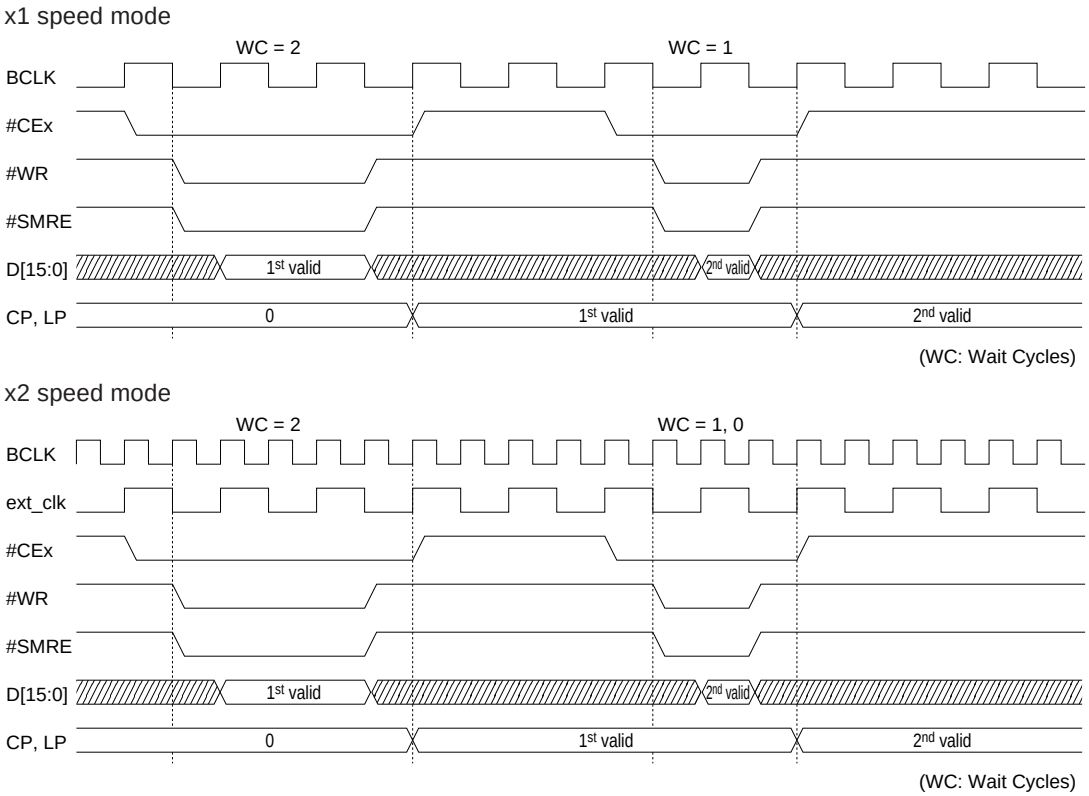


Figure III.14.7 Data Write Timing Chart

Setting BCU to Access to the NAND Flash I/F Registers

The control registers used for the NAND flash interface are mapped into Area 6. 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)
   The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.

I/O Memory of NAND Flash Interface

Table III.14.3 shows the NAND flash interface control bits.

Table III.14.3 NAND Flash 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]: 1 * reserved; 0 1 #SMRE; 0 0 P35, etc.	0	R/W	
		D2	**EFP350**			0		
		D1	**EFP341**	P34 port extended function	EFP34[1:0]: 1 * reserved; 0 1 #SMWE; 0 0 P34, etc.	0	R/W	
		D0	**EFP340**			0		
NAND flash I/F #CE area select register	**0300100 (B)**	D7	**SMCODE**	Booting code enable	1 Enabled 0 Disabled	1	R/W	
		D6–2	**–**	reserved	–	–	–	0 when being read.
		D1	**SMCES1**	NAND flash I/F #CE area selection (according of CEFUNC)	SMCES[1:0]: 1 1 #CE9/#CE17(+18); 1 0 #CE8/#CE14; 0 1 #CE7/#CE13; 0 0 #CE5/#CE15(+16)	0	R/W	
		D0	**SMCES0**			0		
ECC reset/ready register	**0300101 (B)**	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**ECCRST**	ECC circuit reset	1 Reset 0 No effect	–	W	
			ECCRDY	Parity data ready status	1 Ready 0 Busy	0	R	
ECC enable register	**0300102 (B)**	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**ECCEN**	ECC circuit enable	1 Enabled 0 Disabled	0	R/W	
Mode register	**0300103 (B)**	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**MODE**	Flash device mode	1 16 bits 0 8 bits	0	R/W	
Area 0 ECC column parity data register	**0300104 (B)**	D7	**CP05**	Area 0 column parity data	0x0 to 0x3F	1	R	
		D6	**CP04**			1		
		D5	**CP03**			1		
		D4	**CP02**			1		
		D3	**CP01**			1		
		D2	**CP00**			1		
		D1	**–**	Unused bit	–	1	R	1 when being read.
		D0	**–**	Unused bit	–	1	R	
Area 0 ECC line parity register 0 (low-order byte)	**0300105 (B)**	D7	**LP07**	Area 0 ECC line parity low-order byte	0x0 to 0xFF	1	R	
		D6	**LP06**			1		
		D5	**LP05**			1		
		D4	**LP04**			1		
		D3	**LP03**			1		
		D2	**LP02**			1		
		D1	**LP01**			1		
		D0	**LP00**			1		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Area 0 ECC line parity register 1 (high-order byte)	**0300106 (B)**	D7	**LP015**	Area 0 ECC line parity high-order byte	0x0 to 0xFF	1	R	
		D6	**LP014**			1		
		D5	**LP013**			1		
		D4	**LP012**			1		
		D3	**LP011**			1		
		D2	**LP010**			1		
		D1	**LP09**			1		
		D0	**LP08**			1		
Area 1 ECC column parity data register	**0300107 (B)**	D7	**CP15**	Area 1 column parity data	0x0 to 0x3F	1	R	
		D6	**CP14**			1		
		D5	**CP13**			1		
		D4	**CP12**			1		
		D3	**CP11**			1		
		D2	**CP10**			1		
		D1	**–**	Unused bit	–	1	R	1 when being read.
		D0	**–**	Unused bit	–	1	R	
Area 1 ECC line parity register 0 (low-order byte)	**0300108 (B)**	D7	**LP17**	Area 1 ECC line parity low-order byte	0x0 to 0xFF	1	R	
		D6	**LP16**			1		
		D5	**LP15**			1		
		D4	**LP14**			1		
		D3	**LP13**			1		
		D2	**LP12**			1		
		D1	**LP11**			1		
		D0	**LP10**			1		
Area 1 ECC line parity register 1 (high-order byte)	**0300109 (B)**	D7	**LP115**	Area 1 ECC line parity high-order byte	0x0 to 0xFF	1	R	
		D6	**LP114**			1		
		D5	**LP113**			1		
		D4	**LP112**			1		
		D3	**LP111**			1		
		D2	**LP110**			1		
		D1	**LP19**			1		
		D0	**LP18**			1		

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.14.4 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 NAND flash 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]) / NAND flash I/F #CE area select register (0x300100)

Selects a #CE area to be assigned to NAND flash.

Table III.14.5 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	#CE5	#CE15	#CE15+#CE16

(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" (#CE5).

SMCODE: Booting code enable (D7) / NAND flash I/F #CE area select register (0x300100)

Enables reading the NAND flash booting code.

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

When SMCODE is set to "1", the NAND flash boot function is enabled. If the BOOT pin is set to "0" and this bit is set to "1", the NAND flash booting code that is located on addresses 0xc00000 to 0xc0007f is executed after an initial reset. When the BOOT pin is set to "1", the NAND flash booting code is not executed even if this bit is set to "1".
At initial reset, SMCODE is set to "1" (enabled).

ECCRST: ECC circuit reset (D0) / ECC reset/ready register (0x300101)
ECCRDY: Parity data ready status (D0) / ECC reset/ready register (0x300101)

Resets the ECC generator and indicates whether the parity data is ready for reading or not.

When writing (ECCRST):

Write "1": Reset
Write "0": Invalid

Writing "1" to this bit (ECCRST) resets the ECC generator. This operation is required in every page before reading or writing data (before setting ECCEN to "1").

When reading (ECCRDY):

Read "1": Ready for read
Read "0": Not ready

When "1" is read from this bit (ECCRDY), the ECC generator has generated the column and line parity data and it is ready for reading. When "0" is read, the column and line parity data has not been generated. Make sure that this bit is "1" before reading the ECC data registers.

At initial reset, this bit is set to "0" (not ready).

ECCEN: ECC circuit enable (D0) / ECC enable register (0x300102)

Enables the ECC generator.

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

Set ECCEN to "1" before reading or writing data from/to a page and set to "0" after 512-byte data is read/written from/to the page. (This ECC generator supports 512-byte mode only.) This bit must be set to "0" during command cycles, address cycles, and status output cycles.
At initial reset, ECCEN is set to "0" (disabled).

MODE: Flash device mode (D0) / Mode register (0x300103)

Selects the flash device size.

Write "1": 16-bit flash device
Write "0": 8-bit flash device
Read: Valid

Set MODE to "1" when using a 16-bit NAND flash device or set to "0" when using an 8-bit NAND flash device.
At initial reset, MODE is set to "0" (8-bit flash device).

CP05–CP00: Area 0 column parity data (D[7:2]) / Area 0 ECC column parity data register (0x300104)
CP15–CP10: Area 1 column parity data (D[7:2]) / Area 1 ECC column parity data register (0x300107)

The column parity data that is generated by the ECC generator from the 256-byte flash data in the area is stored in this register.
The data read from this register is valid when ECCRDY (D0/0x300101) is read as "1".
At initial reset, CP is set to "1".

LP015–LP08: Area 0 ECC line parity high-order byte (D[7:0]) / Area 0 ECC line parity register 1 (0x300106)
LP07–LP00: Area 0 ECC line parity low-order byte (D[7:0]) / Area 0 ECC line parity register 0 (0x300105)
LP115–LP18: Area 1 ECC line parity high-order byte (D[7:0]) / Area 1 ECC line parity register 1 (0x300109)
LP17–LP10: Area 1 ECC line parity low-order byte (D[7:0]) / Area 1 ECC line parity register 0 (0x300108)

The line parity data that is generated by the ECC generator from the 256-byte flash data in the area is stored in these registers.
The data read from these registers is valid when ECCRDY (D0/0x300101) is read as "1".
At initial reset, LP is set to "1".

Programming Notes

Configuring the flash connection pins

- The chip select (#CE7/#CE13, #CE8/#CE14, #CE5/#CE15, or #CE9/#CE17), CLE and ALE signal output ports must be used as general-purpose I/O ports. Before switching the port function from chip select output to general-purpose I/O, make sure that the I/O port data register has been set to "1" in order to avoid that the port will go low to activate the chip select signal for NAND flash.

Setting up the system

- The numbers of output disable delay cycles and wait cycles should be decided according to the specification of NAND flash.
- In x1 speed mode, at least one wait cycle should be inserted. This limitation does not apply to x2 speed mode.
- Only little endian is supported for the flash area.
- Both A0 and BSL system modes are supported in the NAND flash interface module.
- This NAND flash interface does not support byte accesses to 16-bit flash devices. When reading or writing 16-bit flash devices, do not use byte operation instructions (e.g. "ld.b" in assembly programs, "char" definition in C programs).

ECC functions

- The ECC generator supports two flash data organization modes: 512-byte × 8 bit mode and 256-word × 16-bit mode. 256-byte × 8-bit mode is not supported as it is a seldom-used feature.
- The flash I/O signals must be connected to the D[15:0] pins when using the ECC function.
- The ECC reset bit (D0/0x300101) should be set before using the ECC function.
- The ECC enable bit (D0/0x300102) should be set only while reading or writing flash data. It should be disabled during command input, address input or status reading.
- To calculate ECC parity codes when continuously reading flash data in x1 speed mode, for example, when word reading (using "ld.w") from a 16-bit flash device or half-word read (using "ld.h") from an 8-bit flash device is performed, a one read-hold cycle must be inserted into the read cycles.

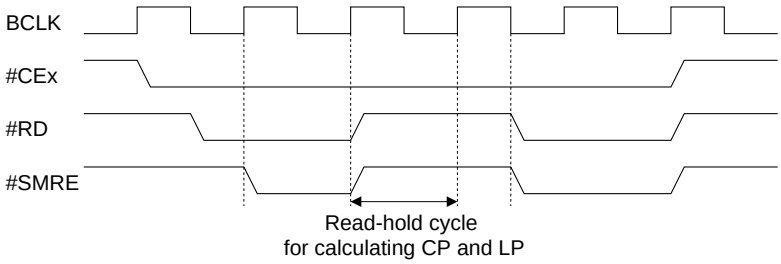


Figure III.14.8 Inserting Read-hold Cycle

NAND Flash Operation Speed

This NAND flash interface module supports up to 26 MHz clock in x1 speed mode or up to 48 MHz clock in x2 speed mode. 32 MHz clock in x1 speed mode is not allowed since the data hold time is shorter for writing data to NAND flash devices.

The solution to increase the data-hold time when writing data to NAND flash devices lies in using the BCUCLK positive edge to trigger data output from C33 and using the chip select signal output from the BCU for the flash area. However, one additional wait cycle is needed to meet the NAND flash data setup time requirement when writing data to NAND flash devices.

The performance comparison table is shown below.

Table III.14.6 Performance Comparison Table

Operation	Item	Solution to support up to 26 MHz x1 speed			Solution to support 32 MHz x1 speed
		24 MHz x1 speed	32 MHz x2 speed	48 MHz x2 speed	32 MHz x1 speed
Read	No. of cycles (min.)	2	4	4	2
	1 data read time (min.)	83 ns	125 ns	83 ns	94 ns
Write	No. of cycles (min.)	2	4	4	3
	1 data write time (min.)	83 ns	125 ns	83 ns	94 ns

It can be seen that the current performance of 24 MHz in x1 speed mode or 48 MHz in x2 speed mode is better than 32 MHz in x1 speed mode. So, this module only supports up to 26 MHz in x1 speed mode instead of 32 MHz in x1 speed mode as other S1C33L05 modules.

III-15 MULTIMEDIACARD (SPI MODE) INTERFACE

Configuration of SPI

The S1C33L05 contains an SPI (Serial Peripheral Interface) as an extended peripheral circuit that allows connection of an MMC (MultiMediaCard) in SPI mode. The following shows its features:

- Operates in Master mode (using three pins: data input, data output, and clock output).
- Supports 1 to 16-bit data transfer.
- Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
- 1 to 65536 clocks of delay can be inserted between transfers.
- Generates transmit data register empty and receive data register full interrupts.

Figure III.15.1 shows the structure of the SPI circuit.

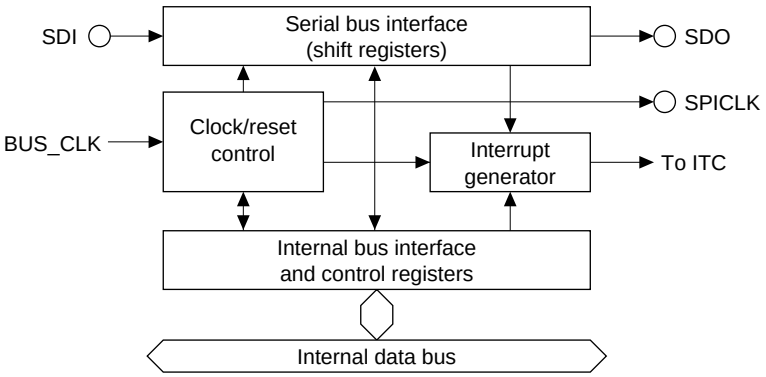


Figure III.15.1 Structure of SPI

I/O Pins of SPI

Table III.15.1 shows the input/output pins for the SPI.

Table III.15.1 I/O Pins of SPI

Pin name	I/O	Function	Function select bit
P31(#BUSGET/#GARD/**SDO**)	I/O	I/O port / Bus status monitor singal output / Area read signal output / SPI data output	EFP31[1:0](D[3:2])/P31–P33 port function extension register(0x300046)
P32(#DMAACK0/#SRDY3/**SPICLK**)	I/O	I/O port / #DMAACK0 output / Serial I/F Ch. 3 ready signal input/output / SPI clock output	EFP32[1:0](D[5:4])/P31–P33 port function extension register(0x300046)
P33(#DMAACK1/SIN3/**SDI**)	I/O	I/O port / #DMAACK1 output / Serial I/F Ch.3 data input / SPI data input	EFP33[1:0](D[7:6])/P31–P33 port function extension register(0x300046)

The SDO, SDI and SPICLK pins are shared with the P31, P33 and P32 I/O ports, respectively. At cold start, these pins are all set for the I/O port. When using these pins for the SPI, set EFP31[1:0] (D[3:2]), EFP32[1:0] (D[5:4]) and EFP33[1:0] (D[7:6]) / P31–P33 port function extension register (0x300046) to "01".
These pins are also shared with other I/O signals and they cannot be used when the SPI is used.
At hot start, the register retains its status from prior to the reset.

Setting BCU to Access the SPI Control Registers

The P31–P33 function extension register used for pin configuration and the SPI control registers are mapped into Area 6. 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)
   The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.

The P31–P33 port function extension register is allocated into the 8-bit device area (0x300046) in Area 6 and the SPI control registers are allocated into the 16-bit device area (0x3A0000 to 0x3A000C).

Notes: • Be sure to use 16-bit access instructions for reading/writing from/to the SPI control registers (0x3A0000 to 0x3A000C). The SPI control registers do not allow reading/writing using 8-bit access instructions.

• Be sure to configure all the P31–P33 ports for the SDI, SDO, and SPICLK functions, even if only SPI receive mode or SPI transmit mode is used.

Setting Transfer Conditions

Transfer data bit length

The bit length of a transfer data can be specified within the range of 1 to 16 bits using the BPT[3:0] (D[D:A]) / SPI control register 1 (0x3A0004). The transfer data bit length is set to the BPT[3:0] set value + 1.

Clock and data transfer timing

This SPI operates only in the master mode. The transfer clock SPICLK is generated based on an internal clock. It is used to operate the shift register during data transfer and is output from the SPICLK pin to the slave device. The source clock from which SPICLK is generated is BUS_CLK (bus clock), and a division ratio can be specified using the MCBR[2:0] (D[6:4]) / SPI control register 1 (0x3A0004) to setup the bit rate for transfer.

Table III.15.2 Setting Up the Bit Rate

MCBR2	MCBR1	MCBR0	Division ratio
1	1	1	f_{BUS_CLK}/512
1	1	0	f_{BUS_CLK}/256
1	0	1	f_{BUS_CLK}/128
1	0	0	f_{BUS_CLK}/64
0	1	1	f_{BUS_CLK}/32
0	1	0	f_{BUS_CLK}/16
0	0	1	f_{BUS_CLK}/8
0	0	0	f_{BUS_CLK}/4

(f_{BUS_CLK}: Bus clock frequency)

The polarity and phase of the SPICLK clock can be selected using the control bits below.
Selecting the SPICLK polarity: CPOL (D8) / SPI control register 1 (0x3A0004)
Selecting the SPICLK phase: CPHA (D9) / SPI control register 1 (0x3A0004)

These two control bits configure the transfer timing as follows:

Data output timing for transmitting

	CPOL = "0" (active high)	CPOL = "1" (active low)
CPHA = "0"	SPICLK falling edge	SPICLK rising edge
CPHA = "1"	SPICLK rising edge	SPICLK falling edge

Data input timing for receiving (latching into the shift register)

	CPOL = "0" (active high)	CPOL = "1" (active low)
CPHA = "0"	SPICLK rising edge	SPICLK falling edge
CPHA = "1"	SPICLK falling edge	SPICLK rising edge

Figure III.15.2 shows the clock and the data transfer timing.

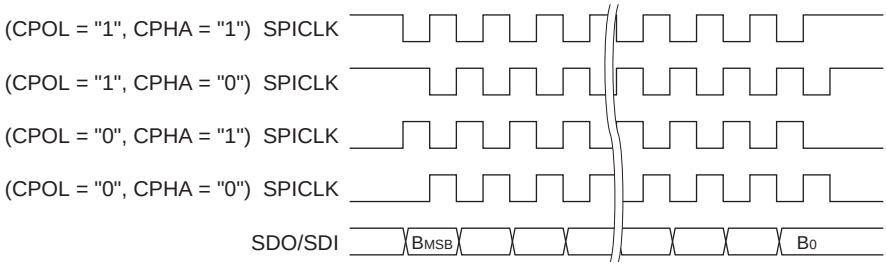


Figure III.15.2 Clock and Data Transfer Timing

Wait cycles between data transfers

1 to 65536 SPICLK clocks of delay time can be inserted between data transfers (in each transfer for specified number of data bits) using the SPI wait register (0x3A0008). The value set in the register (0 to 65535) + 1 is used as the number of wait cycles.

Note: Make sure that the ENA (D0) / SPI control register 1 (0x3A0004) is set to "0" (SPI circuit is off) before setting up the transfer conditions mentioned above.

Controlling Data Transfers

Data transmission

The following shows the data transmitting procedure:

1. Set up the transfer conditions as described in the previous section.
2. Set up the interrupt conditions using the ITC registers and the SPI interrupt control register (explained later). The SPI interrupt factor flag in the ITC must be cleared before enabling the interrupt.
3. Write "1" to the ENA (D0) / SPI control register 1 (0x3A0004) to turn the SPI circuit on.
   The SPI circuit starts frequency division of the source clock.
4. Write the transmit data to the SPI transmit data register SPITD[F:0] (0x3A0002).
   The SPI circuit loads the data written to the SPI transmit data register into the shift register for transmission and starts outputting of the clock from the SPICLK pin. The data bits in the shift register are shifted one by one at the rising or falling edge configured with CPHA (D9/0x3A0004) and CPOL (D8/0x3A0004), and are output from the SDO pin. The MSB of data is transmitted first.

   The SPI circuit provides the transmit data empty flag TDEF (D4) / SPI status register (0x3A000A) to indicate the transmit data register status. This flag is reset to "0" (not empty) when data is written to the transmit data register and is set to "1" (empty) when the written data is loaded into the shift register. An interrupt can be generated simultaneous with this flag set to "1". Check to see if the TDEF flag is set to "1" by polling or using this interrupt before the next transmit data can be written to the SPI transmit data register. (The TDEF flag should be read to check whether the interrupt has occurred due to the data transmission, since all the SPI interrupt factors use the same interrupt line to the ITC.)

   The SPI circuit continues data output from the SDO pin and clock output from the SPICLK pin until data transmission for the number of bits specified with the BPT[3:0] (D[D:A]/0x3A0004) is finished.
   If the next transmit data exists in the transmit data register when a data transfer has finished, the SPI circuit repeats the same transmit operation as above. However, the SPI circuit delays starting the next transmission for the number of SPICLK cycles specified with the SPI wait register (0x3A0008). The data written to the transmit data register is not loaded into the shift register until after the expiration of the delay time.

   The transmitter status sets/resets the transfer busy flag BSYF (D6) / SPI status register (0x3A000A). BSYF is set to "1" when transmission is in progress or in the wait cycles specified with the SPI wait register, and reset to "0" upon completion of a transmit operation. Use this flag to check if a transmission has completed.
5. After all transmissiona have completed, write "0" to ENA to turn the SPI circuit off.

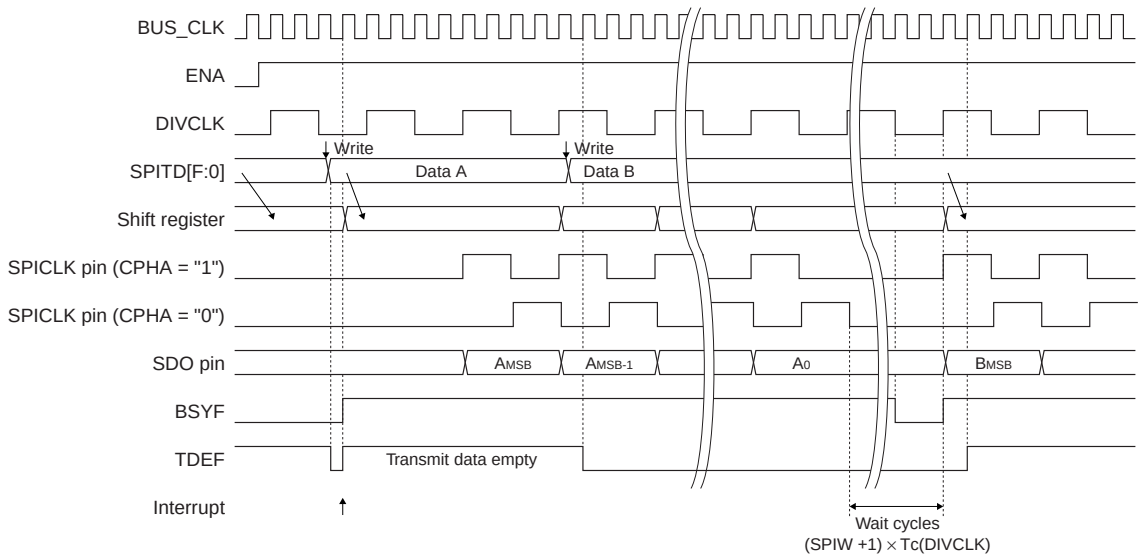


Figure III.15.3 Data Transmit Timing Chart

Data receiving

The following shows the data receiving procedure:

1. Set up the transfer conditions as described in the previous section.
2. Set up the interrupt conditions using the ITC registers and the SPI interrupt control register (explained later). The SPI interrupt factor flag in the ITC must be cleared before enabling the interrupt.
3. Write "1" to the ENA (D0) / SPI control register 1 (0x3A0004) to turn the SPI circuit on.
   The SPI circuit starts frequency division of the source clock.
4. Write dummy data to the SPI transmit data register SPITD[F:0] (0x3A0002). Writing to the SPI transmit data register is used as the trigger for data receiving as well as start of data transmission.
   The SPI circuit starts output of the generated clock from the SPICLK pin.

   The data bits transferred from the slave device are shifted in the shift register for data receiving one by one at the rising or falling edge configured with CPHA (D9/0x3A0004) and CPOL (D8/0x3A0004). The MSB of data is received first.
   When the specified number of bit data is received in the shift register, the received data is loaded into the SPI receive data register SPIRD[F:0] (0x3A0000). At the same time, the receive data full flag RDFF (D2) / SPI status register (0x3A000A) is set to "1" (data full) to indicate that the receive data can be read from the SPI receive data register and a data receive interrupt can be generated.
5. Check to see if the RDFF flag is set to "1" by polling or using the interrupt and read data from the SPI receive data register. (The RDFF flag should be read to check whether the interrupt has occurred due to data receiving, since all the SPI interrupt factors use the same interrupt line to the ITC.)
   When data is read from the SPI receive data register, the RDFF flag is cleared to "0".

   To receive data successively, write dummy data to the SPI transmit data register every time data is received. The SPI circuit continues the receive operation when dummy data has been written to the SPI transmit data register. However, the SPI circuit delays starting the next receiving for the number of SPICLK cycles specified with the SPI wait register (0x3A0008). The clock output is suspended until after the expiration of the delay time.
6. Repeat Steps 4 and 5 until all data are received.

   In the same manner as transmission, the transfer busy flag BSYF (D6/0x3A000A) is set to "1" when a receive operation is in progress.
7. After all data has been received, write "0" to ENA to turn the SPI circuit off.

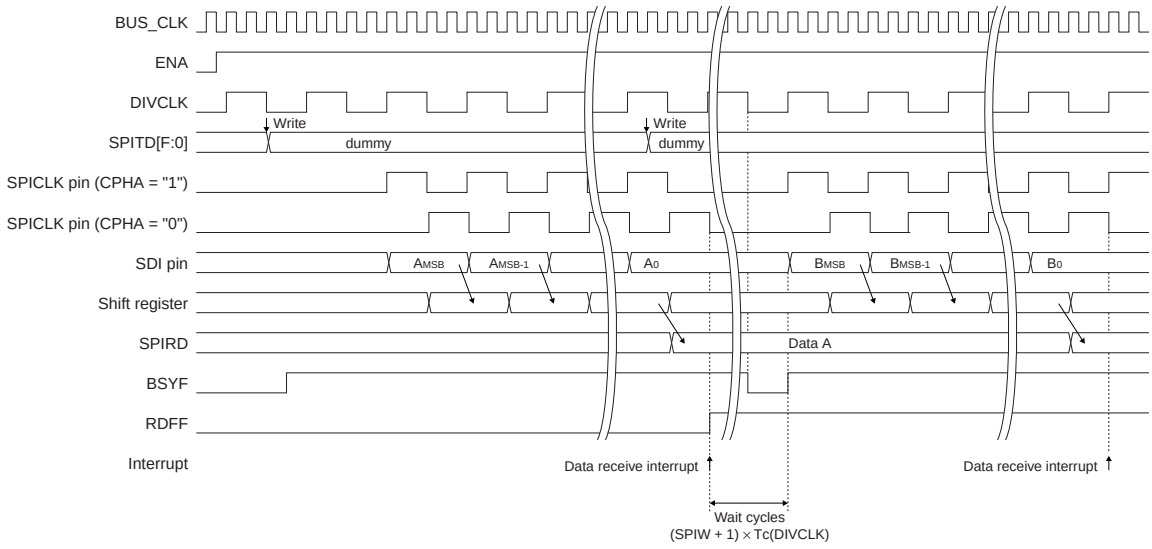


Figure III.15.4 Data Receiving Timing Chart

Data transmit/receive trigger and precautions

This SPI circuit starts data transmission and receiving simultaneously when data is written to the SPI transmit data register.

When data transmission is started, a data receive operation is also performed. Therefore, if received data that has not been read exists in the receive data register, it will be overwritten. Furthermore, undesired interrupts other than a data transmit interrupt (data receive interrupt) occur if the interrupts for receiving are enabled. These interrupts should be disabled before starting data transmission. Before performing data reception after data is transmitted, read the receive data register to clear the RDFF flag, since the flag has been set due to the receive operation performed simultaneously with the previous data transmission.

When receiving data, the data transmit interrupt should be disabled like the data transmission.

SPI Interrupts and DMA

The SPI circuit can generate the following two types of interrupts:
- Data transmit interrupt
- Data receive interrupt

Data transmit interrupt

When the transmit data written to the transmit data register is loaded into the shift register, the transmit data empty flag TDEF (D4/0x3A000A) is set to "1" and the data transmit interrupt factor occurs. To output this interrupt signal to the ITC, both the data transmit interrupt enable bit TEIE (D4) and interrupt request enable bit IRQE (D0) in the SPI interrupt control register (0x3A000C) should be set to "1". This makes it possible to set the interrupt factor flag FSPII in the ITC to "1" due to the occurrence of the data transmit interrupt factor. If the interrupt conditions set using the ITC registers 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 data to be written to the transmit data register by means of a DMA transfer.

Data receive interrupt

When the data received in the shift register is loaded into the receive data register, the receive data full flag RDFF (D2/0x3A000A) is set to "1" and the data receive interrupt factor occurs. To output this interrupt signal to the ITC, both the data receive interrupt enable bit RFIE (D2) and interrupt request enable bit IRQE (D0) in the SPI interrupt control register (0x3A000C) should be set to "1". This makes it possible to set the interrupt factor flag FSPII in the ITC to "1" due to the occurrence of the data receive interrupt factor. If the interrupt conditions set using the ITC registers 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.

Interrupt request signal to the ITC

The above two interrupt factors use the same signal line to request interrupts to the ITC. Figure III.15.5 shows the SPI interrupt request circuit. The interrupt requests are controlled by the interrupt enable bit corresponding to each interrupt factor and the interrupt request enable bit.

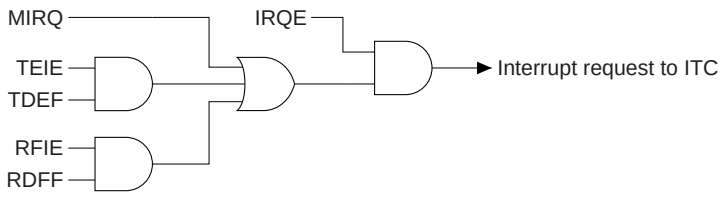


Figure III.15.5 SPI Interrupt Request Circuit

In addition to the two interrupt factors, the MIRQ (D1) / SPI interrupt control register (0x3A000C) is provided to control the interrupt request signal manually. When MIRQ is set to "1" the interrupt request signal is asserted and when it is set to "0", the interrupt request signal is negated. This allows the software to generate SPI interrupts. This interrupt request is also effective when the interrupt request enable bit IRQE is "1". Furthermore, setting MIRQ to "0" cannot clear the interrupt factor flag FSPII in the ITC.

Control registers of the interrupt controller

Table III.15.3 shows the ITC's control registers for the SPI interrupts.

Table III.15.3 Control Registers of Interrupt Controller

Interrupt factor	Interrupt factor flag	Interrupt enable register	Interrupt priority register
Receive data full	FSPII (D7/0x402A9)	ESPII (D7/0x402A6)	PSPII[2:0] (D[6:4]/0x402A3)
Transmit data empty			
Manual IRQ control			

When the interrupt factor described above occurs, the 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 SPI interrupt factors can be used to invoke intelligent DMA (IDMA). This enables successive transmit/receive operations between memory and the transmit/receive-registers to be performed by means of a DAM transfer.
The IDMA channel numbers set for the SPI interrupt is 0x2D.

The IDMA request and enable bits shown in Table III.15.4 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.15.4 Control Bits for IDMA Transfer

Interrupt factor	IDMA request bit	IDMA enable bit
Receive data full	RSPII (D7/0x402AC)	DESPII (D7/0x402AE)
Transmit data empty		
Manual IRQ control		

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)".

Trap vectors

The trap-vector address for the SPI interrupt is set to 0x0C0016C by default.
The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

I/O Memory of SPI

Table III.15.5 shows the SPI control bits.

Note: Be sure to use 16-bit access instructions for reading/writing from/to the SPI control registers (0x3A0000 to 0x3A000C). The SPI control registers do not allow reading/writing using 8-bit access instructions.

Table III.15.5 SPI Control Bits

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P31–P33 port function extension register	**0300046 (B)**	D7	**EFP331**	P33 port extended function	EFP33[1:0] Function	0	R/W	
		D6	**EFP330**		1 * reserved	0		
					0 1 SDI			
					0 0 P33, etc.			
		D5	**EFP321**	P32 port extended function	EFP32[1:0] Function	0	R/W	
		D4	**EFP320**		1 * reserved	0		
					0 1 SPICLK			
					0 0 P32, etc.			
		D3	**EFP311**	P31 port extended function	EFP31[1:0] Function	0	R/W	
		D2	**EFP310**		1 * reserved	0		
					0 1 SDO			
					0 0 P31, etc.			
		D1–0	**–**	reserved	–	–	–	0 when being read.
SPI receive data register	**03A0000 (HW)**	DF	**SPIRDF**	Received data from serial input	0x0 to 0xFFFF	X	R	
		DE	**SPIRDE**	SPIRDF = MSB		X		
		DD	**SPIRDD**	SPIRD0 = LSB		X		
		DC	**SPIRDC**			X		
		DB	**SPIRDB**			X		
		DA	**SPIRDA**			X		
		D9	**SPIRD9**			X		
		D8	**SPIRD8**			X		
		D7	**SPIRD7**			X		
		D6	**SPIRD6**			X		
		D5	**SPIRD5**			X		
		D4	**SPIRD4**			X		
		D3	**SPIRD3**			X		
		D2	**SPIRD2**			X		
		D1	**SPIRD1**			X		
		D0	**SPIRD0**			X		
SPI transmit data register	**03A0002 (HW)**	DF	**SPITDF**	Transmit data to serial output	0x0 to 0xFFFF	X	W	
		DE	**SPITDE**	SPITDF = MSB		X		
		DD	**SPITDD**	SPITD0 = LSB		X		
		DC	**SPITDC**			X		
		DB	**SPITDB**			X		
		DA	**SPITDA**			X		
		D9	**SPITD9**			X		
		D8	**SPITD8**			X		
		D7	**SPITD7**			X		
		D6	**SPITD6**			X		
		D5	**SPITD5**			X		
		D4	**SPITD4**			X		
		D3	**SPITD3**			X		
		D2	**SPITD2**			X		
		D1	**SPITD1**			X		
		D0	**SPITD0**			X		
SPI control register 1	**03A0004 (HW)**	DF–E	**–**	reserved	–	–	–	0 when being read.
		DD	**BPT3**	Number of data bits per transfer	Number of data bits per transfer = BPT + 1	0	R/W	
		DC	**BPT2**			0		
		DB	**BPT1**			0		
		DA	**BPT0**			0		
		D9	**CPHA**	SPICLK phase selection	1 Phase 1 0 Phase 0	0	R/W	
		D8	**CPOL**	SPICLK polarity selection	1 Active low 0 Active high	0	R/W	
		D7	**–**	reserved	–	–	–	0 when being read.
		D6	**MCBR2**	Master clock bit rate (in master mode only)	Master clock divided value = 4×2^{MCBR}	0	R/W	
		D5	**MCBR1**			0		
		D4	**MCBR0**			0		
		D3	**CLKS**	Shift clock source selection	Fixed set "0"	0	R/W	
		D2	**–**	reserved	–	–	–	0 when being read.
		D1	**MODE**	SPI mode selection	Fixed set "1"	0	R/W	
		D0	**ENA**	SPI enable	1 Enabled 0 Disabled	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI control register 2	**03A0006 (HW)**	DF–B	**–**	reserved	–	–	–	0 when being read.
		DA	**SS**	#SCS control	reserved, fixed at "0"	0	R/W	Master mode
		D9	**SSP**	#SCS polarity selection	reserved, fixed at "0"	0	R/W	Slave mode
		D8	**SSC**	#SCS configuration	reserved, fixed at "0"	0	R/W	Master mode
		D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RDYS**	#SRDY type selection	reserved, fixed at "0"	0	R/W	
		D0	**RDYE**	#SRDY enable	reserved, fixed at "0"	0	R/W	
SPI wait register	**03A0008 (HW)**	DF	**SPIWF**	Wait cycle control (in master mode only) SPIWF = MSB SPIW0 = LSB	0x0 to 0xFFFF	0	R/W	
		DE	**SPIWE**			0		
		DD	**SPIWD**			0		
		DC	**SPIWC**			0		
		DB	**SPIWB**			0		
		DA	**SPIWA**			0		
		D9	**SPIW9**			0		
		D8	**SPIW8**			0		
		D7	**SPIW7**			0		
		D6	**SPIW6**			0		
		D5	**SPIW5**			0		
		D4	**SPIW4**			0		
		D3	**SPIW3**			0		
		D2	**SPIW2**			0		
		D1	**SPIW1**			0		
		D0	**SPIW0**			0		
SPI status register	**03A000A (HW)**	DF–7	**–**	reserved	–	–	–	0 when being read.
		D6	**BSYF**	Transfer busy flag	1 Busy 0 Idle	0	R	Master mode
		D5	**–**	reserved	–	–	–	0 when being read.
		D4	**TDEF**	Transmit data empty flag	1 Empty 0 Not empty	1	R	
		D3	**–**	reserved	–	–	–	
		D2	**RDFF**	Receive data full flag	1 Full 0 Not full	0	R	
		D1–0	**–**	reserved	–	–	–	0 when being read.
SPI interrupt control register	**03A000C (HW)**	DF–6	**–**	reserved	–	–	–	0 when being read.
		D5	**MFIE**	Mode-fault interrupt enable	reserved, fixed at "0"	0	R/W	
		D4	**TEIE**	Data transmit interrupt enable	1 Enabled 0 Disabled	0	R/W	
		D3	**ROIE**	Receive overflow interrupt enable	reserved, fixed at "0"	0	R/W	
		D2	**RFIE**	Data receive interrupt enable	1 Enabled 0 Disabled	0	R/W	
		D1	**MIRQ**	Manual IRQ set/clear	1 Set 0 Clear	0	R/W	
		D0	**IRQE**	Interrupt request enable	1 Enabled 0 Disabled	0	R/W	
USB, SPI interrupt priority register	**00402A3 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PSPII2**	SPI interrupt level	0 to 7	X	R/W	
		D5	**PSPII1**			X		
		D4	**PSPII0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PUSBI2**	USB interrupt level	0 to 7	X	R/W	
		D1	**PUSBI1**			X		
		D0	**PUSBI0**			X		
LCDC, USB, SPI interrupt enable register	**00402A6 (B)**	D7	**ESPII**	SPI interrupts	1 Enabled 0 Disabled	0	R/W	
		D6	**EUSBI**	USB interrupts		0	R/W	
		D5	**ELCDCI**	LCDC interrupts		0	R/W	
		D4	**–**	Extended interrupt (reserved)		0	R/W	Do not write 1.
		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	
LCDC, USB, SPI interrupt factor flag register	**00402A9 (B)**	D7	**FSPII**	SPI interrupts	1 Factor is generated 0 No factor is generated	X	R/W	
		D6	**FUSBI**	USB interrupts		X	R/W	
		D5	**FLCDCI**	LCDC interrupts		X	R/W	
		D4	**–**	Extended interrupt (reserved)		X	R/W	Do not write 1.
		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	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCDC, USB, SPI IDMA request register	**00402AC (B)**	D7	**RSPII**	SPI interrupts	1	IDMA request	0	Interrupt request	0	R/W	
		D6	**RUSBI**	USB interrupts					0	R/W	
		D5	**RLCDCI**	LCDC interrupts					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	Do not write 1.
		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	
LCDC, USB, SPI IDMA enable register	**00402AE (B)**	D7	**DESPII**	SPI interrupts	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D6	**DEUSBI**	USB interrupts					0	R/W	
		D5	**DELCDCI**	LCDC interrupts					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	Do not write 1.
		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	

EFP311–EFP310: P31 port extended function (D[3:2]) / P31–P33 port function extension register (0x300046)
EFP321–EFP320: P32 port extended function (D[5:4]) / P31–P33 port function extension register (0x300046)
EFP331–EFP330: P33 port extended function (D[7:6]) / P31–P33 port function extension register (0x300046)

Switches the P31–P33 port functions.

Table III.15.6 P31–P33 Port Extended Functions

Port	Function extension bit	EFPxx[1:0] = "00"	EFPxx[1:0] = "01"	EFPxx[1:0] = "10"	EFPxx[1:0] = "11"
P31	EFP31[1:0]	**P31** #BUSGET #GARD	SDO	–	–
P32	EFP32[1:0]	**P32** #DMAACK0 #SRDY3	SPICLK	–	–
P33	EFP33[1:0]	**P33** #DMAACK1 SIN3	SDI	–	–

Set EFP31[1:0], EFP32[1:0], and EFP33[1:0] to "01" when using the SPI circuit. The P31, P32, and P33 pins are configured as the SDO, SPICLK, and SDI pins, respectively.
At initial reset, EFP is set to "00".

SPIRDF–SPIRD0: SPI receive data (D[F:0]) / SPI receive data register (0x3A0000)

Stores received data.
When a receive operation is completed and the data received in the shift register is loaded to this register, RDFF is set to "1" (data full). At the same time, a data receive 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 a receive overflow error to occur.
The serial data input from the SDI pin is converted into parallel data beginning with the MSB, 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 register.
When the number of data bits per transfer is set to less than 16 using BPT[3:0], the received data is loaded to the specified number of low-order bits in this register.
This register is a read-only register, so no data can be written to it.
At initial reset, the content of SPIRD becomes indeterminate.

SPITDF–SPITD0: SPI transmit data (D[F:0]) / SPI transmit data register (0x3A0002)

Sets transmit data.
When data is written to this register after "1" is written to ENA, a transmit operation begins.
TDEF is set to "1" (empty) when the data is transferred to the shift register. A data transmit interrupt factor is simultaneously generated. The next transmit data can be written to the register at any time thereafter, even when the SPI is sending data.
The serial-converted data is output from the SDO pin beginning with the MSB, in which the bits set to "1" are output as high-level signals and those set to "0" output as low-level signals.
When the number of data bits per transfer is set to less than 16 using BPT[3:0], only the specified number of low-order bits in this register is transmitted.
This is a write-only register, so the register data cannot be read out.
Writing to this register starts a data receive operation simultaneously with data transmission.
At initial reset, the content of SPITD becomes indeterminate.

BPT3–BPT0: Number of data bits per transfer (D[D:A]) / SPI control register 1 (0x3A0004)

Sets the number of transfer data bits.
The set value in this register + 1 (1 to 16) is the number of bits to be transmitted/received per data transfer.
At initial reset, BPT is set to "0" (one bit).

CPHA: SPICLK phase selection (D9) / SPI control register 1 (0x3A0004)

Selects the phase of the SPI clock.
This bit controls the data transfer timing in conjunction with the CPOL bit (see Figure III.15.6).
This register can be read as well as written.
At initial reset, CPHA is set to "0".

CPOL: SPICLK polarity selection (D8) / SPI control register 1 (0x3A0004)

Selects the polarity of the SPI clock.

Write "1": Active low
Write "0": Active high
Read: Valid

This bit controls the data transfer timing in conjunction with the CPHA bit (see Figure III.15.6).
At initial reset, CPOL is set to "0" (active high).

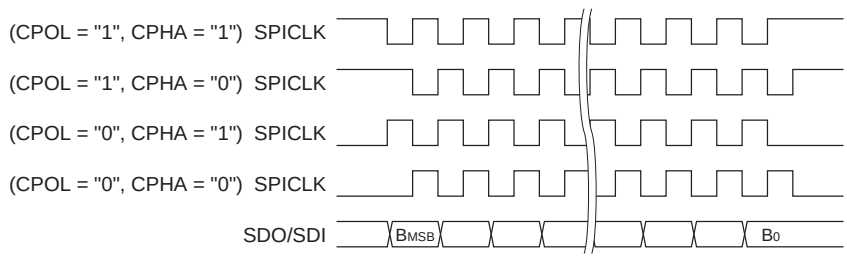


Figure III.15.6 Clock and Data Transfer Timing

MCBR2–MCBR0: Master clock bit rate (D[6:4]) / SPI control register 1 (0x3A0004)

Sets the source clock division ratio for generating the SPI clock. The bit rate is determined with this setting.

Table III.15.7 Setting Bit Rate

MCBR2	MCBR1	MCBR0	Division ratio
1	1	1	f_{BUS_CLK}/512
1	1	0	f_{BUS_CLK}/256
1	0	1	f_{BUS_CLK}/128
1	0	0	f_{BUS_CLK}/64
0	1	1	f_{BUS_CLK}/32
0	1	0	f_{BUS_CLK}/16
0	0	1	f_{BUS_CLK}/8
0	0	0	f_{BUS_CLK}/4

(f_{BUS_CLK}: Bus clock frequency)

At initial reset, MCBR is set to "0" (one bit).

CLKS: Clock source selection (D3) / SPI control register 1 (0x3A0004)

This bit must be fixed at "0".
At initial reset, CLKS is set to "0".

MODE: SPI mode selection (D1) / SPI control register 1 (0x3A0004)

This bit must be fixed at "1".
At initial reset, MODE is set to "0".

ENA: SPI enable (D0) / SPI control register 1 (0x3A0004)

Enables/disables operation of the SPI circuit.

Write "1": Enabled (on)
Write "0": Disabled (off)
Read: Valid

When ENA is set to "1", the SPI circuit starts operating and data transfer is enabled.
By writing data to the transmit data register after this setting, the SPI circuit starts data transmission and receiving.
Write dummy data to the register to start data receiving.
When ENA is set to "0", the SPI circuit goes off and data transfer is disabled.
Make sure that this bit is "0" before setting up data transfer conditions using the SPI registers.
At initial reset, ENA is set to "0" (disabled).

SS: #SCS control (DA) / SPI control register 2 (0x3A0006)

This bit must be fixed at "0".
At initial reset, SS is set to "0".

SSP: #SCS polarity selection (D9) / SPI control register 2 (0x3A0006)

This bit must be fixed at "0".
At initial reset, SSP is set to "0".

SSC: #SCS configuration (D8) / SPI control register 2 (0x3A0006)

This bit must be fixed at "0".
At initial reset, SSC is set to "0".

RDYS: #SRDY type selection (D1) / SPI control register 2 (0x3A0006)

This bit must be fixed at "0".
At initial reset, RDYS is set to "0".

RDYE: #SRDY enable (D0) / SPI control register 2 (0x3A0006)

This bit must be fixed at "0".
At initial reset, RDYE is set to "0".

SPIWF–SPIW0: Wait cycle control (D[F:0]) / SPI wait register (0x3A0008)

Sets the number of wait cycles to be inserted between data transfers.
The set value in this register + 1 is the number of wait cycles.
1 to 65536 SPICLK clock cycles can be specified.
At initial reset, SPIW is set to "0" (no wait).

BSYF: Transfer busy flag (D6) / SPI status register (0x3A000A)

Indicates whether the SPI is in transmit/receive operation.

Read "1": Busy
Read "0": Idle
Write: Invalid

BSYF is set to "1" when the SPI starts data transmission/receiving and keeps "1" while data transmission/receiving is in progress including the wait cycles inserted. BSYF is cleared to "0" upon completion of transmit/receive operation.
At initial reset, BSYF is set to "0" (idle).

TDEF: Transmit data empty flag (D4) / SPI status register (0x3A000A)

Indicates the transmit data register status.

Read "1": Empty
Read "0": Not empty
Write: Invalid

TDEF is cleared to "0" when transmit data is written to the transmit data register and is set to "1" when the written data is transferred to the shift register (transmit operation started).
Transmit data can be written to the transmit data register when this bit = "1".
At initial reset, TDEF is set to "1" (empty).

RDFF: Receive data full flag (D2) / SPI status register (0x3A000A)

Indicates the receive data register status.

Read "1": Data full
Read "0": Not full
Write: Invalid

RDFF is set to "1" when the data received in the shift register is loaded 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, RDFF is set to "0" (not full).

MFIE: Mode-fault interrupt enable (D5) / SPI interrupt control register (0x3A000C)

This bit must be fixed at "0".
At initial reset, MFIE is set to "0".

TEIE: Data transmit interrupt enable (D4) / SPI interrupt control register (0x3A000C)

Enables/disables data transmission interrupts.

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

When TEIE is set to "1", data transmit interrupt requests to the ITC are enabled. However, IRQE must also be set to "1". A data transmit interrupt request occurs when the data written to the transmit data register is transferred to the shift register (transmit operation started). At this time the interrupt factor flag FSPII in the ITC is set to "1" if both TEIE and IRQE have been set to "1" (enabled).
When TEIE is set to "0", data transmit interrupts are not generated.
At initial reset, TEIE is set to "0" (disabled).

ROIE: Receive overflow interrupt enable (D3) / SPI interrupt control register (0x3A000C)

This bit must be fixed at "0".
At initial reset, ROIE is set to "0".

RFIE: Data receive interrupt enable (D2) / SPI interrupt control register (0x3A000C)

Enables/disables data receive interrupts.

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

When RFIE is set to "1", data receive interrupt requests to the ITC are enabled. However, IRQE must also be set to "1". A data receive interrupt request occurs when the data received in the shift register is loaded to the receive data register (receive operation completed). At this time the interrupt factor flag FSPII in the ITC is set to "1" if both RFIE and IRQE have been set to "1" (enabled).
When RFIE is set to "0", data receive interrupts are not generated.
At initial reset, RFIE is set to "0" (disabled).

MIRQ: Manual IRQ set/clear (D1) / SPI interrupt control register (0x3A000C)

Generates an interrupt request to the ITC by manual control.

Write "1": Set IRQ
Write "0": Clear IRQ
Read: Valid

If MIRQ is set to "1" when IRQE is set to "1", the SPI interrupt request signal to be delivered to the ITC becomes active. As a result, the interrupt factor flag FSPII in the ITC is set to "1".
When MIRQ is set to "0", the SPI interrupt request signal becomes inactive (interrupt request is cleared). However, the interrupt factor flag FSPII cannot be cleared to "0" by writing "0" to this bit.
At initial reset, MIRQ is set to "0" (clear IRQ).

IRQE: Interrupt request enable (D0) / SPI interrupt control register (0x3A000C)

Enables/disables interrupt requests to the ITC.

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

When IRQE is set to "1", interrupt requests to the ITC are enabled. An interrupt request is generated and the interrupt factor flag FSPII is set to "1" when an enabled SPI interrupt factor occurs or MIRQ is set to "1" manually.
When IRQE is set to "0", an interrupt request to the ITC is not generated even if the SPI interrupt for each factor is enabled. Also manual interrupt requests using MIRQ are disabled.
At initial reset, IRQE is set to "0" (disabled).

PSPII2–PSPII0: SPI interrupt level (D[6:4]) / USB, SPI interrupt priority register (0x402A3)

Sets the priority level of the SPI interrupt.
The interrupt priority level can be set for each channel in the range of 0 to 7.
At initial reset, PSPII becomes indeterminate.

ESPII: SPI interrupt enable (D7) / LCDC, USB, SPI interrupt enable register (0x402A6)

Enable or disable interrupt generation to the CPU.

Write "1": Interrupt enabled
Write "0": Interrupt disabled
Read: Valid

The ESPII bit is the interrupt enable bit corresponding to all the SPI interrupt factors. When this bit is set to "1", the SPI interrupt is enabled, and when set to "0", the interrupt is disabled.
At initial reset, ESPII is set to "0" (interrupts disabled).

FSPII: SPI interrupt factor flag (D7) / LCDC, USB, SPI interrupt factor flag register (0x402A9)

Indicate the status of SPI 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

FSPII is the interrupt factor flag corresponding to the SPI interrupt. The flag is set to "1" when an SPI 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 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, FSPII becomes indeterminate, so be sure to reset it in the software.

RSPII: SPI IDMA request (D7) / LCDC, USB, SPI IDMA request register (0x402AC)

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 RSPII bit is the IDMA request bit corresponding to the SPI interrupt factors. 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, RSPII is set to "0" (interrupt request).

DESPII: SPI IDMA enable (D7) / LCDC, USB, SPI IDMA enable register (0x402AE)

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 DESPII bit is the IDMA enable bit corresponding to the SPI interrupt factors. 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, DESPII is set to "0" (IDMA disabled).

Programming Notes

(1) Be sure to use 16-bit access instructions for reading/writing from/to the SPI control registers (0x3A0000 to 0x3A000C). The SPI control registers do not allow reading/writing using 8-bit access instructions.

(2) Do not access the SPI control register 1 (0x3A0004), SPI control register 2 (0x3A0006), and SPI wait register (0x3A0008) while the BSYF flag is set to "1" (during data transfer).

(3) To prevent malfunctions, write 0x00 to the SPI interrupt control register (0x3A000C) to disable all the SPI interrupt requests, before disabling the SPI circuit (before setting ENA (D0/0x3A0004) to "0").

THIS PAGE IS BLANK.

III-16 SEQUENTIAL ROM INTERFACE

Configuration of Sequential ROM Interface

The S1C33L05 contains a Sequential ROM Interface as an extended peripheral circuit. This circuit generates the following signals for connecting the sequential ROM (MX23L12813) directly to the bus:
#SQRD (read signal)
SQLALE (lower address latch enable)
SQUALE (upper address latch enable)

Figure III.16.1 shows the structure of the sequential ROM interface.

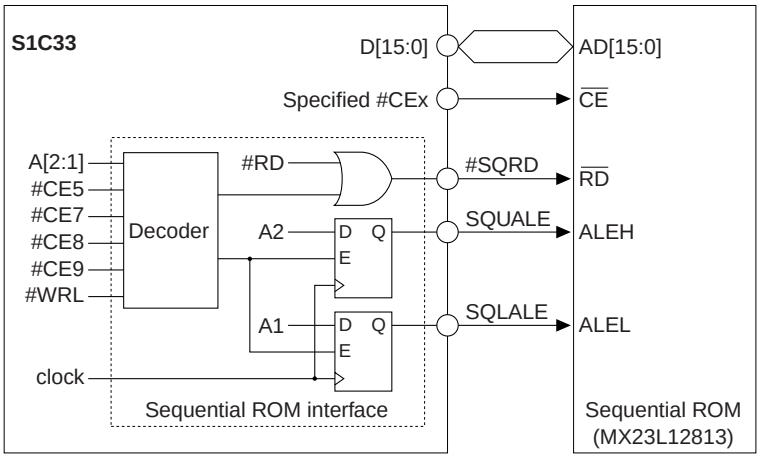


Figure III.16.1 Structure of Sequential ROM Interface

Output Pins of Sequential ROM Interface

Table III.16.1 shows the pins that are used to output the #SQRD, SQLALE, and SQUALE signals.

Table III.16.1 Output Pins of Sequential ROM Interface

Pin name	I/O	Function	Function select bit
PD0(**#SQRD**)	I/O	I/O port / Sequential ROM read signal output	FPD0[1:0](D[1:0])/PD0–PD2 port function extension register(0x300F66)
PD1(**SQLALE**)	I/O	I/O port / Sequential ROM lower byte address latch enable output	FPD1[1:0](D[3:2])/PD0–PD2 port function extension register(0x300F66)
PD2(**SQUALE**)	I/O	I/O port / Sequential ROM upper byte address latch enable output	FPD2[1:0](D[5:4])/PD0–PD2 port function extension register(0x300F66)

The #SQRD, SQLALE, and SQUALE pins are shared with the PD0, PD1, and PD2 I/O ports, respectively. At cold start, these pins are all set for the I/O port. When using the sequential ROM interface, configure these pins for the sequential ROM interface using the PD0–PD2 port function extension register (0x300F66).
At hot start, the register retains its status from prior to the reset.

Setup of the Sequential ROM Interface

When using the sequential ROM, the following settings must be made before the sequential ROM can be accessed:

1. Setting BCU to access the sequential ROM interface control registers
2. Configuring the control signal output pins
3. Selecting a #CE area
4. Selecting the access conditions of the selected area

1. Setting BCU to access the sequential ROM interface control registers

The sequential ROM I/F control registers are mapped to addresses 0x300F66 and 0x3A0100 in Area 6. 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.

- Configure Area 6 as an internal device area by setting A6IO (D9) / Access control register (0x48132) to "1".
- Select little endian as the Area 6 data format by setting A6EC (D1) / Access control register (0x48132) to "0".
- Set the number of wait cycles to be inserted when Area 6 is accessed using A6WT[2:0] (D[A:8]) / Areas 6–4 setup register (0x4812A). The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.

2. Configuring the control signal output pins

Set FPD0[1:0] (D[1:0]), FPD1[1:0] (D[3:2]) and FPD2[1:0] (D[5:4]) / PD0–PD2 port function extension register (0x300F66) to "01" to configure the PD0, PD1, and PD2 pins for outputting the #SQRD, SQLALE, and SQUALE signals, respectively.

3. Selecting a #CE area

Use SQCES[1:0] (D[1:0]) / Sequential ROM I/F #CE area select register (0x3A0100) to select which #CE signal is assigned for accessing sequential ROM. As shown in Table III.16.2, 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.16.2 Selecting a #CE Area

SQCES1	SQCES0	#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	#CE5	#CE15	#CE15+#CE16

(Default: SQCES = CEFUNC = "00")

4. Selecting the access conditions of the selected area

- Select 16 bits for the device size using the area set-up register in the BCU for the selected area.
- Set the number of wait cycles inserted when the sequential ROM is accessed in accordance with the ROM specifications and the bus speed mode.
  In x1 speed mode, at least 3 wait cycles should be set.
  In x2 speed mode, at least 2 wait cycles should be set.
- Enable insertion of the read hold cycle when the sequential ROM is accessed using the Read cycle hold time control register (0x4813C).

Controlling the Sequential ROM

Data read sequence

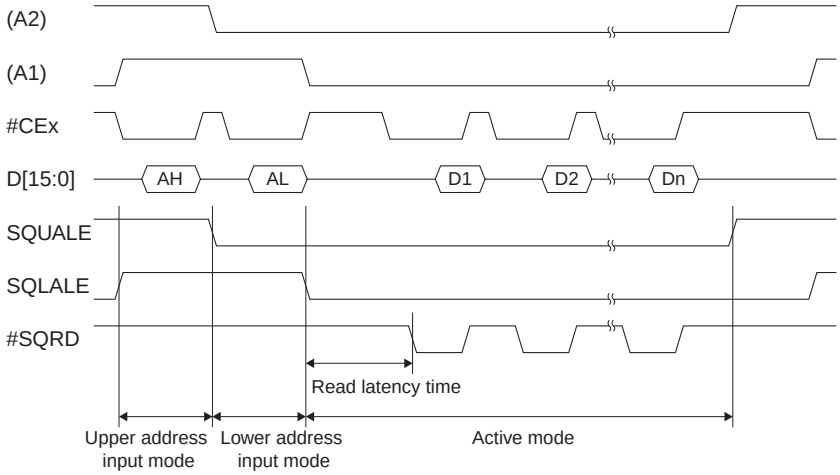


Figure III.16.2 Timing Chart of Sequential ROM Read

The following shows the sequential ROM data read procedure:

1. Write the high-order 8 bits of the sequential ROM read start address. This address data must be written to an address in the area that has been selected for the sequential ROM and the address bits A[2:1] must be "11". A[2:1] = "11" means that the SQUALE (sequential ROM upper address latch enable) and SQLALE (sequential ROM lower address latch enable) signals will go high ("1") to set the sequential ROM into upper address input mode.
   The address bits A[2:1] are used to control the SQUALE and SQLALE signals as shown in Table III.16.3.

Table III.16.3 Corresponding between A[2:1] and SQUALE/SQLALE Signals

A[2:1]	SQUALE	SQLALE	Sequential ROM mode
11	1	1	Upper address input mode
10	1	0	10→00, Standby mode
01	0	1	Lower address input mode
00	0	0	Active mode (for data read)

2. Write the low-order 8 bits of the sequential ROM read start address. In this time, A[2:1] in the address to which the address data is written must be "01".
   Upon completion of Steps 1 and 2, the sequential ROM enters Active mode allowing the CPU to read data sequentially.

3. Wait for the read latency time. This latency time is required after an address is input to the sequential ROM. It is not required during data reading.
   Refer to the specification of the sequential ROM for the read latency time.

4. Read data from an address in the area that has been selected for the sequential ROM. The data at the start address set is read. Data can be read from any address in the sequential ROM area. A[2:1] does not affect the SQUALE and SQLALE signals during reading (fixed at "0").

5. Repeat reading until the desired data is obtained. The address is incremented in the sequential ROM every time data is read.

6. To change the data read address, perform the same procedure from the Step 1.
   When the required data is read, set the sequential ROM into standby mode.

Standby mode in sequential ROM

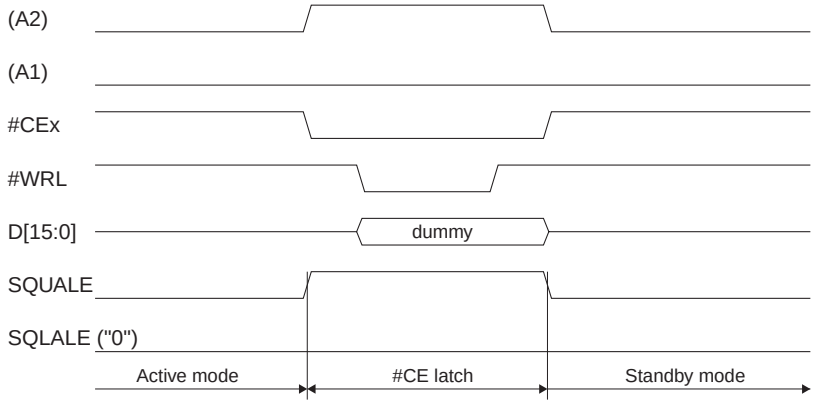


Figure III.16.3 Sequential ROM Standby Sequence

The sequential ROM has a standby mode that can be controlled using the SQUALE and SQLALE signals. When reading data has been finished, set the sequential ROM into standby mode to reduce current consumption by writing dummy data to an address in the sequential ROM area. In this case the address bit A[2:1] must be "10". Refer to the specification of the sequential ROM for details of the standby mode.

I/O Memory of Sequential ROM Interface

Table III.16.4 shows the sequential ROM interface control bits.

Table III.16.4 Sequential ROM Interface Control Bits

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
PD0–PD2 port function extension register	**0300F66 (B)**	D7–6	**–**	reserved	–			–	–	0 when being read.
		D5	**FPD21**	PD2 port extended function	FPD2[1:0]		Function	0	R/W	
		D4	**FPD20**		1	*	reserved	0		
					0	1	SQUALE			
					0	0	PD2			
		D3	**FPD11**	PD1 port extended function	FPD1[1:0]		Function	0	R/W	
		D2	**FPD10**		1	*	reserved	0		
					0	1	SQLALE			
					0	0	PD1			
		D1	**FPD01**	PD0 port extended function	FPD0[1:0]		Function	0	R/W	
		D0	**FPD00**		1	*	reserved	0		
					0	1	#SQRD			
					0	0	PD0			
Sequential ROM I/F #CE area select register	**03A0100 (HW)**	DF–2	**–**	reserved	–			–	–	0 when being read.
		D1	**SQCES1**	Sequential ROM I/F #CE area selection (depends on CEFUNC)	SQCES[1:0]		#CE area	0	R/W	
		D0	**SQCES0**		1	1	#CE9/#CE17(+18)	0		
					1	0	#CE8/#CE14			
					0	1	#CE7/#CE13			
					0	0	#CE5/#CE15(+16)			

FPD01–FPD00: PD0 port extended function (D[1:0]) / PD0–PD2 port function extension register (0x300F66)
FPD11–FPD10: PD1 port extended function (D[3:2]) / PD0–PD2 port function extension register (0x300F66)
FPD21–FPD20: PD2 port extended function (D[5:4]) / PD0–PD2 port function extension register (0x300F66)

Switches the PD0–PD2 port functions.

Table III.16.5 PD0–PD2 Port Extended Functions

FPDx1	FPDx0	PD0	PD1	PD2
1	*	reserved	reserved	reserved
0	1	#SQRD	SQLALE	SQUALE
0	0	PD0	PD1	PD2

Set FPD0[1:0], FPD1[1:0], and FPD2[1:0] to "01" when using the sequential ROM interface. The PD0, PD1, and PD2 pins are configured as the #SQRD, SQLALE and SQUALE pins, respectively.
At initial reset, FPD is set to "00".

SQCES1–SQCES0: #CE area selection (D[1:0]) / Sequential ROM I/F #CE area select register (0x3A0100)

Selects a #CE area to be assigned to sequential ROM.

Table III.16.6 Selecting a #CE Area

SQCES1	SQCES0	#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	#CE5	#CE15	#CE15+#CE16

(Default: SQCES = CEFUNC = "00")

At initial reset, SQCES is set to "00" (#CE5).

THIS PAGE IS BLANK.

III-17 S1C33L05 CLOCK SYSTEM AND MISCELLANEOUS REGISTERS

Configuration of the S1C33L05 Clock System

The S1C33L05 clock system consists of the CLG (Clock generator) and additional clock control circuits for the extended peripheral circuits.
This allows software to control the system clock frequency in fine steps and the clock supply to the extended circuits individually. By operating the system with as low a frequency as possible and disabling clock supply to unnecessary peripheral circuits, current consumption can be decreased.

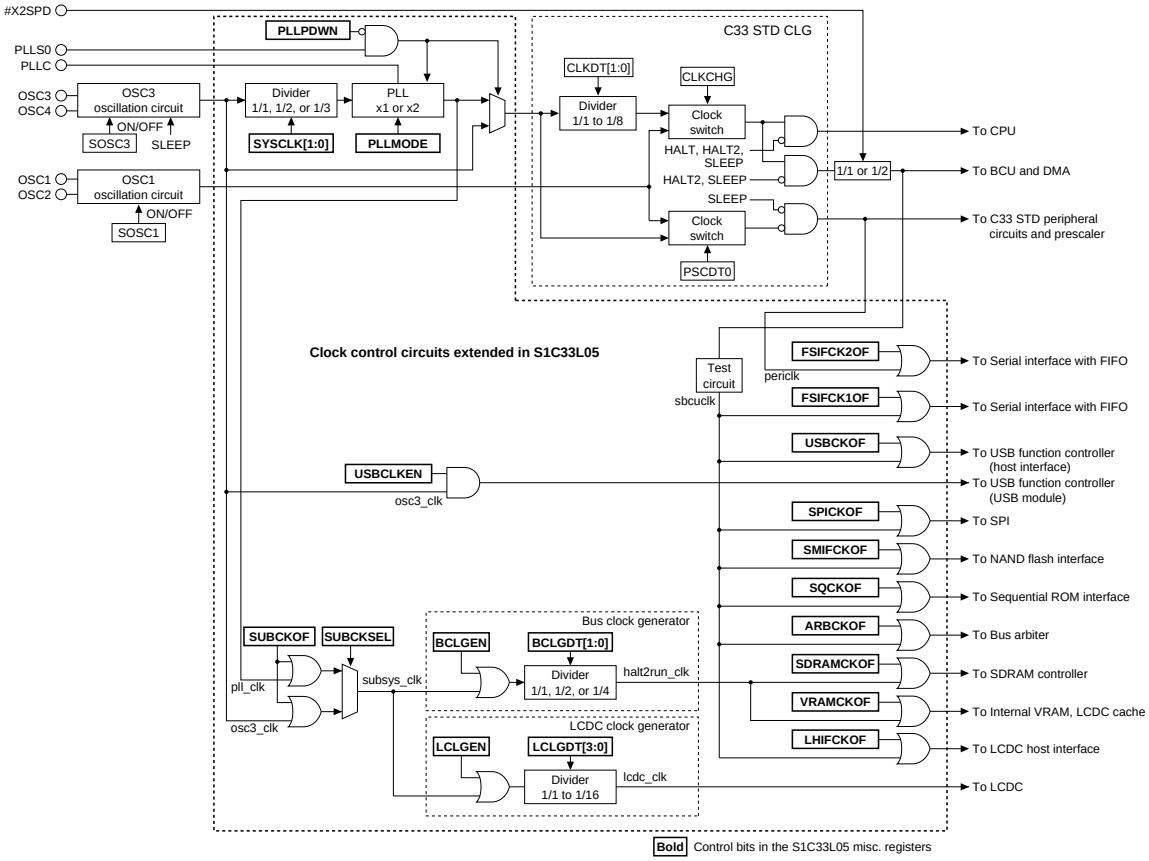


Figure III.17.1 S1C33L05 Clock System

Miscellaneous Registers

Miscellaneous (Misc) registers (0x300F20–0x300F39) are provided to control the extended clock system.
The registers have the functions listed below.
1. Controlling the system clock (PLL configuration)
2. Controlling the clock supply to the extended peripheral circuits
3. Miscellaneous settings for the extended peripheral circuits

Setting BCU to access the Misc registers

The Misc registers are mapped in Area 6. 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)
   The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.

Write protection for Misc registers

The Misc registers (0x300F20, 0x300F30 to 0x300F39) are normally disabled against writing in order to prevent it from malfunctioning due to unnecessary writing.
To enable these registers for writing, the Misc write protect register (0x300F2F) must be set to 0b10010110 (0x96).
After data is written to a Misc register other than the Misc write protect register (0x300F2F), the Misc write protect register (0x300F2F) is automatically reset to 0x00 to disable writing to the Misc registers again. Therefore, the Misc write protect register (0x300F2F) must be set to 0x96 each time a Misc register is written to. The write-enabled status can also be changed to write-protected status by writing a value other than 0x96 to the Misc write protect register (0x300F2F).
The Misc write protect register (0x300F2F) does not affect the readout from the Misc registers.

Controlling the System Clock

Selecting a PLL input clock frequency

An additional frequency divider is placed between the OSC3 oscillation circuit and the PLL. This makes it possible to select the PLL input clock frequency from three kinds using SYSCLK[1:0] (D[1:0]) / System clock division register (0x300F30).

Table III.17.1 Selecting a PLL Input Clock Frequency

SYSCLK1	SYSCLK0	PLL input clock
1	1	OSC3 / 3
1	0	OSC3 / 2
0	1	OSC3
0	0	

This selection affects the operating frequency of the C33 STD core when the PLL is used.

Controlling the PLL

In the extended clock system, the PLL mode that is configured using the PLLS0 pin in the existing models can be configured by software. Use the control bits shown below for this control.

PLLPDWN (D1) / PLL control register (0x300F31)
PLLPDWN = "0": PLL is used
PLLPDWN = "1": PLL is not used

When the PLL is not used, the OSC3 clock (not divided) is used as the system clock.

PLLMODE (D0) / PLL control register (0x300F31)
PLLMODE = "0": x1 (bypass mode)
PLLMODE = "1": x2 (frequency-doubling mode)

Note: To control the PLL by software, the PLLS0 pin must be set to "1" (VDDE). When the PLLS0 pin is set to "0" (VSS), the software control using the above PLL control bits is disabled.

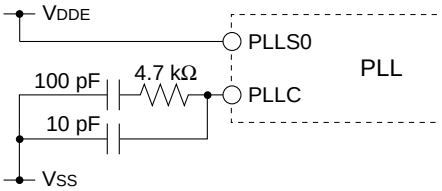


Figure III.17.2 PLL Pins

List of system clock settings

Table III.17.2 lists the system clock settings according to the PLL input clock frequency and PLL mode set with the control bits above.

Table III.17.2 List of System Clock Settings

PLLPDWN	PLLMODE	SYSCLK1	SYSCLK0	CLG input clock	PLL
1	*	*	*	OSC3 (PLL bypassed)	Not used
0	1	1	1	(OSC3 / 3) × 2	Used
		1	0	(OSC3 / 2) × 2	
		0	*	(OSC3 / 1) × 2	
	0	1	1	OSC3 / 3	
		1	0	OSC3 / 2	
		0	*	OSC3 / 1	

The system clock configured here is input to the CLG of the C33 STD and is not directly used as the CPU or BCU clock. It is necessary to configure the CLG to generate the CPU and BCU operating clocks. Refer to Section II-6, "CLG (Clock Generator)", for setting the CPU operating clock.

Precautions on system clock setup

If the settings of the divider and PLL are changed while the CPU core is running with the OSC3 clock, the CPU malfunctions. Therefore, the system clock should be configured after switching the operating clocks of the CPU core and peripheral circuits to OSC1 in accordance with the procedure described below.

1. Write "1" to SOSC1 (D0) / Power control register (0x40180) to turn the low-speed (OSC1) oscillation circuit on. (Notes 1 and 2)
2. Wait until the OSC1 oscillation stabilizes. (Notes 1 and 4)
3. Stop the peripheral circuits that use the OSC3 clock, or switch the prescaler clock to OSC1.
4. Write "0" to CLKCHG (D2) / Power control register (0x40180) to switch the CPU operating clock from OSC3 to OSC1. (Notes 2 and 5)
5. Write "0" to SOSC3 (D1) / Power control register (0x40180) to turn the high-speed (OSC3) oscillation circuit off. (Notes 2 and 5)
6. Configure the system clock using the System clock division register (0x300F30) and the PLL control register (0x300F31).
7. Write "1" to SOSC3 (D1) / Power control register (0x40180) to turn the high-speed (OSC3) oscillation circuit on. (Note 2)
8. Wait until the OSC3 oscillation stabilizes. (Note 4)
9. Write "1" to CLKCHG (D2) / Power control register (0x40180) to switch the CPU operating clock from OSC1 to OSC3. (Notes 2 and 7)

Notes: 1. Steps 1 and 2 are required only when the low-speed (OSC1) oscillation circuit is inactive.

2. The Power control protect register (0x4019E) must be set to 0b10010110 (0x96) to enable the Power control register (0x40180) for writing before a value can be written to the Power control register. This operation is required every time the Power control register is written to.
3. The Misc write protect register (0x300F2F) must be set to 0b10010110 (0x96) to enable the Misc registers for writing before a value can be written to a Misc register. This operation is required every time a Misc register is written to.
4. Refer to "Electrical Characteristics" for the oscillation stabilization times.
5. 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.
6. When changing the divider setting using SYSCLK[1:0] (D[1:0]/0x300F30), BCLGDT[1:0] (D[1:0]/0x300F33), or LCLGDT[3:0] (D[3:0]/0x300F34), these control bits must be changed in bit units one by one to avoid occurrence of narrow pulses on the clock tree. For example, when changing the set value from 0b00 to 0b11, first write 0b01 (or 0b10) then write 0b11. Changing two or more bits at the same time (0b00 to 0b11 in this example) may cause a malfunction of the circuits that use the changed clock.
7. Also remember to switch the prescaler clock from OSC1 to OSC3.

Controlling the Clock Supply to the Extended Peripheral Circuits

The Misc registers provide the control bits that enable/disable supplying the clocks to the extended peripheral circuits individually (see Figure III.17.1). By stopping the clock supply to unused peripheral circuits using these control bits, current consumption can be reduced.

Clock control for SPI, NAND flash I/F, sequential ROM I/F, and bus arbiter

The SPI, NAND flash interface, sequential ROM interface, and bus arbiter operate with the sbcuclk clock that is equivalent to the BCU clock. Use the control bit shown below to disable the clock supply.

SPI:	SPICKOF (D4) / Module clock control register 0 (0x300F35)
NAND flash I/F:	SMIFCKOF (D3) / Module clock control register 0 (0x300F35)
Sequential ROM I/F:	SQCKOF (D2) / Module clock control register 0 (0x300F35)
Bus arbiter:	ARBCKOF (D1) / Module clock control register 0 (0x300F35)

These bits are initialized to "0", so the sbcuclk clock is supplied to the peripheral circuit modules after an initial reset. To stop supplying the clock, set the control bit corresponding to the circuit to be stopped to "1".
In HALT2 and SLEEP mode, the sbcuclk clock stops regardless of the settings of these bits.

Serial interface with FIFO

Two clocks (sbcuclk and periclk) are input to the serial interface with FIFO and can be enabled/disabled to be supplied individually.

The sbcuclk clock is absolutely necessary to operate the serial interface with FIFO. Supplying the clock to the serial interface with FIFO can be controlled using the FSIFCK1OF (D6) / Module clock control register 0 (0x300F35). This bit is initialized to "0", so the sbcuclk clock is supplied to the serial interface with FIFO after an initial reset. When the serial interface with FIFO is not used, set FSIFCK1OF to "1" to stop supplying the clock. In HALT2 and SLEEP mode, the sbcuclk clock stops regardless of the setting of this bit.

The periclk clock is the peripheral circuit operating clock (OSC3 clock/PLL output clock or OSC1 clock) that is selected using the PSCDT0 (D0) / Prescaler clock select register (0x40181). In the serial interface with FIFO, the baud-rate timer uses this clock. When an external clock is used as the transfer clock, this clock supply to the serial interface with FIFO can be disabled using the FSIFCK2OF (D5) / Module clock control register 0 (0x300F35). This bit is initialized to "0", so the periclk clock is supplied to the serial interface with FIFO after an initial reset. When the baud-rate time or the serial interface with FIFO is not used, set FSIFCK2OF to "1" to stop supplying the clock. In SLEEP mode, the periclk clock stops regardless of the setting of this bit.

USB function controller

Two clocks (sbcuclk and osc3_clk) are input to the USB function controller and can be enabled/disabled to be supplied individually.

The sbcuclk clock is used in the host interface of the USB function controller. Supplying the clock to the USB function controller can be controlled using the USBCKOF (D7) / Module clock control register 0 (0x300F35). This bit is initialized to "0", so the sbcuclk clock is supplied to the USB function controller after an initial reset. When the USB function controller is not used, set USBCKOF to "1" to stop supplying the clock.

The osc3_clk clock (OSC3 clock) is used as the USB clock. When the USB controller enters Snooze mode or when the USB function controller is not used, this clock supply can be disabled using the USBCLKEN (D4) / Macro control register (0x300F20). USBCLKEN should be set to "1" to supply the clock to the USB function controller, or set to "0" to stop supplying the clock. In SLEEP mode, the osc3_clk clock stops regardless of the setting of this bit.

SDRAM controller, internal VRAM, and LCDC cache

The SDRAM controller, internal VRAM and LCDC cache are connected to the bus in the extended block and are accessed from the C33 STD via a bridge. The extended block includes the bus clock generator to generate the clock for driving this bus and it can be controlled using a Misc register.

Controlling the bus clock generator

1. Selecting the source clock

   Use SUBCKSEL (D0) / Sub system clock select register (0x300F32) to select the source clock for the bus clock generator.
   SUBCKSEL = "0": OSC3 clock
   SUBCKSEL = "1": PLL output clock

   The OSC3 clock and the PLL output clock are supplied to the source clock selector by default. When the SDRAM controller and the LCD controller are not used, the clock supply to the source clock selector can be disabled by writing "1" to the SUBCKOF (D0) / Sub system clock control register (0x300F37) to reduce current consumption.

   Be sure to set SUBCKOF (D0/0x300F37) to "1" before switching the source clock between the OSC3 clock and the PLL output clock.

2. Setting up the clock frequency (source clock division ratio)

   The bus clock generator divides the source clock to generate the bus clock. Use the BCLGDT[1:0] (D[1:0]) / Bus clock generator control register (0x300F33) to set the division ratio as shown in Table III.17.3.

Table III.17.3 Source Clock for Generating Bus Clock

BCLGDT1	BCLGDT0	BCLG source clock
1	1	subsys_clk / 4
1	0	subsys_clk / 2
0	1	subsys_clk
0	0	

   Do not change the BCLGDT1 and BCLGDT0 control bits at the same time to avoid occurrence of narrow pulses in the halt2run_clk clock tree.

3. Enabling/disabling the clock output

   The clock output from the bus clock generator can be controlled using the BCLGEN (D7) / Bus clock generator control register (0x300F33).
   BCLGEN = "0": Clock output is disabled.
   BCLGEN = "1": Clock output is enabled.

   By writing "1" to BCLGEN (D7/0x300F33), the bus clock generator starts outputting the halt2run_clk clock configured in Steps 1 and 2 above. In SLEEP mode, the halt2run_clk clock stops regardless of the setting of this bit. In HALT or HALT2 mode, this bit is effective.

   Be sure to set BCLGEN (D7/0x300F33) to "0" before setting up the clock (Step 2).

Clock supply to each module

In the initialized condition, the halt2run_clk clock that is generated by the bus clock generator is supplied to the SDRAM controller, internal VRAM and LCDC cache by writing "1" to BCLGEN (D7/0x300F33) only.
In addition to this common control bit, the SDRAM controller and internal VRAM/LCDC cache have clock control bits individually allowing software to enable/disable supplying each clock.

Supplying the clock to the SDRAM controller can be controlled using the SDRAMCKOF (D1) / Module clock control register 1 (0x300F36). This bit is initialized to "0", so the halt2run_clk clock is supplied to the SDRAM controller if BCLGEN (D7/0x300F33) is set to "1". When the SDRAM controller is not used, set SDRAMCKOF to "1" to stop supplying the clock.

Supplying the clock to the internal VRAM and LCDC cache can be controlled using the VRAMCKOF (D0) / Module clock control register 1 (0x300F36). This bit is initialized to "0", so the halt2run_clk clock is supplied to the internal VRAM and LCDC cache if BCLGEN (D7/0x300F33) is set to "1". When the internal VRAM and LCD controller is not used, set VRAMCKOF to "1" to stop supplying the clock.

LCD controller

Two clocks (sbcuclk and lcdc_clk) are input to the LCD controller and can be enabled/disabled to be supplied individually.

The sbcuclk clock is used in the host interface of the LCD controller. Supplying the clock to the LCD controller can be controlled using the LHIFCKOF (D0) / Module clock control register 0 (0x300F35). This bit is initialized to "0", so the sbcuclk clock is supplied to the LCD controller after an initial reset. When the LCD controller is not used, set LHIFCKOF to "1" to stop supplying the clock.

The LCDC clock (lcdc_clk) is used as the pixel clock and other clocks for the LCD interface and is generated by the LCDC clock generator. The LCDC clock generator can be controlled using a Misc register.

Controlling the LCDC clock generator

1. Selecting the source clock

   Use SUBCKSEL (D0) / Sub system clock select register (0x300F32) to select the source clock for the LCDC clock generator the same as for the bus clock generator.
   SUBCKSEL = "0": OSC3 clock
   SUBCKSEL = "1": PLL output clock

   The OSC3 clock and the PLL output clock are supplied to the source clock selector by default. When the SDRAM controller and the LCD controller are not used, the clock supply to the source clock selector can be disabled by writing "1" to the SUBCKOF (D0) / Sub system clock control register (0x300F37) to reduce current consumption.

   Be sure to set SUBCKOF (D0/0x300F37) to "1" before switching the source clock between the OSC3 clock and the PLL output clock.

2. Setting up the clock frequency (source clock division ratio)

   The LCDC clock generator divides the source clock to generate the LCDC clock. Use the LCLGDT[3:0] (D[3:0]) / LCDC clock generator control register (0x300F34) to set the division ratio as shown in Table III.17.4.

Table III.17.4 Source Clock for Generating LCDC Clock

LCLGDT3	LCLGDT2	LCLGDT1	LCLGDT0	LCLG source clock
1	1	1	1	subsys_clk / 16
1	1	1	0	subsys_clk / 15
1	1	0	1	subsys_clk / 14
1	1	0	0	subsys_clk / 13
1	0	1	1	subsys_clk / 12
1	0	1	0	subsys_clk / 11
1	0	0	1	subsys_clk / 10
1	0	0	0	subsys_clk / 9
0	1	1	1	subsys_clk / 8
0	1	1	0	subsys_clk / 7
0	1	0	1	subsys_clk / 6
0	1	0	0	subsys_clk / 5
0	0	1	1	subsys_clk / 4
0	0	1	0	subsys_clk / 3
0	0	0	1	subsys_clk / 2
0	0	0	0	subsys_clk

   Do not change the LCLGDT3, LCLGDT2, LCLGDT1, and LCLGDT0 control bits at same time to avoid occurrence of narrow pulse in the LCDC clock tree.

3. Enabling/disabling the clock output

   The clock output from the LCDC clock generator can be controlled using the LCLGEN (D7) / LCDC clock generator control register (0x300F34).
   LCLGEN = "0": Clock output is disabled.
   LCLGEN = "1": Clock output is enabled.

   By writing "1" to LCLGEN (D7/0x300F34), the LCDC clock generator starts outputting the lcdc_clk clock configured in Steps 1 and 2 above. In SLEEP mode, the lcdc_clk clock stops regardless of the setting of this bit. In HALT or HALT2 mode, this bit is effective.

   Be sure to set LCLGEN (D7/0x300F34) to "0" before setting up the clock (Step 2).

Miscellaneous Settings for the Extended Peripheral Circuits

The Misc registers provide the control bits for setting the functions below.

Selecting A0/#BSL bus interface mode for the extended circuit

The BSLSEL (D0) / A0/#BSL select register (0x300F38) is used to select the bus interface method for the extended peripheral circuit.
BSLSEL = "0": A0 mode
BSLSEL = "1": #BSL mode

This interface method must be the same as that of the BCU set using the SBUSST (D3) / Bus control register (0x4812E).
Furthermore, SBUSST (D3/0x4812E) for the BCU must be set before BSLSEL (D0/0x300F38).

Bus speed modes for SDRAM and internal VRAM

The C33 STD BCU provides the #X2SPD pin and the Bus speed setting register (0x4813E) in order to set the bus speed mode (x1 speed mode and x2 speed mode) for each area. However, the bus speed mode set in the BCU is not applied to the SDRAM and internal VRAM since they are accessed via the bus arbiter. Therefore, a separate bus speed mode controller is provided in the extended block and can be controlled using a Misc register.
Use the SDRAMX2 (D1) / x2 speed mode control register (0x300F39) and the IVRAMX2 (D0) / x2 speed mode control register (0x300F39) to set the bus speed modes for the SDRAM and the internal VRAM, respectively.
When this control bit is set to "1", x1 speed mode (CPU - bus clock ratio is 1 : 1) is set. The bus clock for the SDRAM/internal VRAM and the CPU system clock will be the same.
When this control bit is set to "0", x2 speed mode (CPU - bus clock ratio is 2 : 1) is set. The bus clock frequency for the SDRAM/internal VRAM becomes half of the CPU system clock.
This control is effective only when the #X2SPD pin is set to "0" (x2 speed mode) the same as the Bus speed setting register (0x4813E) in the BCU. If the #X2SPD pin is set to "1", the mode is fixed at x1 speed mode.

Wait and snooze control for the USB function controller

The USB function controller needs a different wait control from other internal blocks and the USB bus wait control bits USBWT[2:0] (D[2:0]) / Macro control register (0x300F20) are provided. Wait states selected by USBWT[2:0] will be inserted into the USB register access cycle along with the wait states selected by the A12WT[2:0] (D[2:0]) / Areas 12–11 set-up register (0x48124).
USBWT[2:0] should be set as follows according to the CPU operating clock frequency:

Table III.17.5 Setting Wait State for USB Bus

USBWT2	USBWT1	USBWT0	CPU clock	Wait cycle
1	1	1	40 MHz or higher	BCU wait setting + 140 ns
1	1	0		BCU wait setting + 120 ns
1	0	1	24 MHz	BCU wait setting + 100 ns
1	0	0		BCU wait setting + 80 ns
0	1	1		BCU wait setting + 60 ns
0	1	0	12 MHz	BCU wait setting + 40 ns
0	0	1		BCU wait setting + 20 ns
0	0	0	6 MHz	BCU wait setting + 0 ns

Also the macro control register (0x300F20) contains the USBSNZ bit (D5) that controls the Snooze mode for the USB function controller.

Refer to Section VI-2, "USB Function Controller" for details on control of the USB function controller.

Misc Register Map

Table III.17.6 shows the Misc registers.

Table III.17.6 Misc Registers

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Macro control register	0300F20 (B)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	USBSNZ	USB snooze control	1 Enabled 0 Disabled	0	R/W	
		D4	USBCLKEN	USB clock enable	1 Enabled 0 Disabled	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	USBWT2	USB bus wait count	0 to 7	1	R/W	
		D1	USBWT1			1		
		D0	USBWT0			1		
Misc write protect register	0300F2F (B)	D7	WRPROT7	Misc register write protect	Writing 10010110 (0x96) removes the write protection of the Misc registers (0x300F20, 0x300F30–0x300F39). Writing another value to this register or writing to a Misc register (0x300F20, 0x300F30–0x300F39) will set the write protection again.	0	R/W	
		D6	WRPROT6			0		
		D5	WRPROT5			0		
		D4	WRPROT4			0		
		D3	WRPROT3			0		
		D2	WRPROT2			0		
		D1	WRPROT1			0		
		D0	WRPROT0			0		
System clock division register	0300F30 (B)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	SYSCLK1	PLL input clock setup	SYSCLK[1:0] 1 1 OSC3 / 3; 1 0 OSC3 / 2; 0 * OSC3 / 1	1	R/W	
		D0	SYSCLK0	(OSC3 clock division ratio)		0		
PLL control register	0300F31 (B)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	PLLPDWN	PLL software power down (when PLLS0 is set to "1")	1 PLL not used (CLK=OSC3) 0 PLL used	0	R/W	
		D0	PLLMODE	PLL multiple mode select	1 x2 0 x1	0	R/W	
Sub system clock select register	0300F32 (B)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SUBCKSEL	Sub system source clock select	1 PLL clock 0 OSC3 clock	0	R/W	
Bus clock generator control register	0300F33 (B)	D7	BCLGEN	Bus clock generator enable	1 Enabled 0 Disabled	0	R/W	
		D6–2	–	reserved	–	–	–	0 when being read.
		D1	BCLGDT1	Bus clock setup	BCLGDT[1:0] Division ratio: 1 1 subsys_clk / 4; 1 0 subsys_clk / 2; 0 * subsys_clk / 1	0	R/W	
		D0	BCLGDT0	(Sub system clock division ratio)		0		
LCDC clock generator control register	0300F34 (B)	D7	LCLGEN	LCDC clock generator enable	1 Enabled 0 Disabled	0	R/W	
		D6–4	–	reserved	–	–	–	0 when being read.
		D3	LCLGDT3	LCDC clock setup	LCDC clock frequency = subsys_clk / (LCLGDT[3:0] + 1) [Hz]	0	R/W	
		D2	LCLGDT2	(Sub system clock division ratio)		0		
		D1	LCLGDT1			0		
		D0	LCLGDT0			0		
Module clock control register 0	0300F35 (B)	D7	USBCKOF	USB clock (sbcuclk)	1 Disabled 0 Enabled	0	R/W	
		D6	FSIFCK1OF	FSIF clock (bcuclk)		0	R/W	
		D5	FSIFCK2OF	FSIF clock (periclk)		0	R/W	
		D4	SPICKOF	SPI clock (sbcuclk)		0	R/W	
		D3	SMIFCKOF	SMIF clock (sbcuclk)		0	R/W	
		D2	SQCKOF	SQROM clock (sbcuclk)		0	R/W	
		D1	ARBCKOF	Bus arbiter clock (sbcuclk)		0	R/W	
		D0	LHIFCKOF	LCDC host I/F clock (sbcuclk)		0	R/W	
Module clock control register 1	0300F36 (B)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	SDRAMCKOF	SDRAMC clock (halt2run_clk)	1 Disabled 0 Enabled	0	R/W	
		D0	VRAMCKOF	Internal VRAM clock (halt2run_clk)		0	R/W	
Sub system clock control register	0300F37 (B)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SUBCKOF	OSC3/PLL clock disable	1 Disabled 0 Enabled	0	R/W	
A0/#BSL select register	0300F38 (B)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	BSLSEL	A0/#BSL mode select	1 #BSL mode 0 A0 mode	0	R/W	
x2 speed mode control register	0300F39 (B)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	SDRAMX2	SDRAM x2 speed mode disable	1 x1 speed mode 0 x2 speed mode	0	R/W	
		D0	IVRAMX2	IVRAM x2 speed mode disable		0	R/W	

WRPROT7–WRPROT0: Misc register write protect (D[7:0]) / Misc write protect register (0x300F2F)

These bits remove the protection against writing to the Misc registers.

Write "0b10010110": Write protection removed
Write other than the above: Write-protected
Read: Valid

Before writing to a Misc register (0x300F20, 0x300F30–0x300F39), set WRPROT[7:0] to "0b10010110" (0x96) to remove the protection against writing to these addresses.
The removed write protection will be set again by writing to a Misc register (0x300F20, 0x300F30–0x300F39) or writing a value other than "0b10010110" to WRPROT[7:0].
At initial reset, WRPROT is set to "0b00000000" (write-protected).

USBSNZ: USB snooze control (D5) / Macro control register (0x300F20)

Enables/disables the USB snooze control.

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

When this bit is set to "1", the USB controller performs a transition sequence and then it enters Snooze mode. When this bit is set to "0", the USB controller resumes operating. For details of the snooze sequence, refer to "Snooze" in Section VI-2, "USB Function Controller".
At initial reset, USBSNZ is set to "0" (disabled).

USBCLKEN: USB clock enable (D4) / Macro control register (0x300F20)

Enables/disables the OSC3 clock output to the USB function controller.

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

The OSC3 clock is output to the USB function controller by writing "1" to USBCLKEN. The clock output is disabled by writing "0".
At initial reset, USBCLKEN is set to "0" (disabled).

USBWT2–USBWT0: USB bus wait count (D[2:0]) / Macro control register (0x300F20)

Set the number of wait cycles to be inserted into the USB bus cycles for accessing the USB registers.
The USB function controller in Area 11 needs a different wait control from other internal blocks and USBWT[2:0] is used to adjust the USB register read/write timing. The wait control using the BCU register is effective and the wait state specified with USBWT[2:0] is added to that of the BCU when USB registers are accessed.
Basically configure the Area 11 access conditions using the BCU register as follows:

When the CPU operating clock is 40 MHz or higher:
Number of wait cycles = 2 ... A12WT[2:0](D[2:0]/0x48124) = "010"
Output disable delay time = 0.5 cycles ... A12DF[1:0] (D[5:4]/0x48124) = "00"

When the CPU operating clock is less than 40 MHz:
Number of wait cycles = 1 ... A12WT[2:0](D[2:0]/0x48124) = "001"
Output disable delay time = 0.5 cycles ... A12DF[1:0] (D[5:4]/0x48124) = "00"

When the CPU is running with a 40 MHz or more operating clock under the conditions above, the USB registers need more than 140 ns of wait state.

USBWT[2:0] should be set as follows according to the CPU operating clock frequency:

Table III.17.7 Setting Wait State for USB Bus

USBWT2	USBWT1	USBWT0	CPU clock	Wait cycle
1	1	1	40 MHz or higher	BCU wait setting + 140 ns
1	1	0		BCU wait setting + 120 ns
1	0	1	24 MHz	BCU wait setting + 100 ns
1	0	0		BCU wait setting + 80 ns
0	1	1		BCU wait setting + 60 ns
0	1	0	12 MHz	BCU wait setting + 40 ns
0	0	1		BCU wait setting + 20 ns
0	0	0	6 MHz	BCU wait setting + 0 ns

Note that the 48 MHz clock is counted to generate the adjustment wait state from the USBWT[2:0].
At initial reset, USBWT[2:0] is set to "7".

SYSCLK1–SYSCLK0: PLL input clock setup (D[1:0]) / System clock division register (0x300F30)

Sets the PLL input clock frequency (OSC3 clock division ratio).

Table III.17.8 Setting of PLL Input Clock

SYSCLK1	SYSCLK0	PLL input clock
1	1	OSC3 / 3
1	0	OSC3 / 2
0	1	OSC3
0	0	

Do not change the SYSCLK1 and SYSCLK0 control bits at same time to avoid occurrence of narrow pulse in the halt2run_clk clock tree.
At initial reset, SYSCLK is set to "0b10" (OSC3 / 2).

PLLPDWN: PLL software power down (D1) / PLL control register (0x300F31)

Selects whether the PLL is used or not.

Write "1": Not used
Write "0": Used
Read: Valid

When using the OSC3 clock directly as the system clock, the PLL can be bypassed by writing "1" to PLLPDWN in order to reduce current consumption. When the PLL is used, set PLLPDWN to "0".
This control bit is effective only when the PLLS0 pin is set to "1". If the PLLS0 pin is set to "0", the PLL is bypassed regardless of the PLLPDWN setting.
At initial reset, PLLPDWN is set to "0" (used).

PLLMODE: PLL multiple mode select (D0) / PLL control register (0x300F31)

Selects the PLL multiple mode.

Write "1": x2 mode
Write "0": x1 mode
Read: Valid

When "1" is written to PLLMODE, the PLL enters x2 mode and the PLL input clock frequency is doubled before it is output to the CLG. When "0" is written, the PLL enters x1 mode and the PLL input clock is output to the CLG without doubling.
At initial reset, PLLMODE is set to "0" (x1 mode).

Note: Before changing the System clock division register (0x300F30) and/or PLL control register (0x300F31), the clock source of the CPU and peripheral operating clocks must be switched to OSC1. Otherwise, the CPU malfunctions.

SUBCKSEL: Sub system source clock select (D0) /Sub system clock select register (0x300F32)

Selects the source clock for the bus clock and LCDC clock generators.

Write "1": PLL output clock
Write "0": OSC3 clock
Read: Valid

When "1" is written to SUBCKSEL, the PLL output clock is selected as the source clock (subsys_clk) for the bus clock and LCDC clock generators in the extended block. When "0" is written, the OSC3 clock is selected.
At initial reset, SUBCKSEL is set to "0" (OSC3 clock).

BCLGEN: Bus clock generator enable (D7) / Bus clock generator control register (0x300F33)

Enables/disables the bus clock generator.

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

When "1" is written to BCLGEN, the source clock supply is enabled and the bus clock generator starts outputting the halt2run_clk clock for the SDRAM and internal VRAM. When "0" is written, the bus clock generator stops outputting the halt2run_clk clock.
At initial reset, BCLGEN is set to "0" (disabled).

BCLGDT1–BCLGDT0: Bus clock setup (D[1:0]) / Bus clock generator control register (0x300F33)

Configures the frequency divider in the bus clock generator to set the halt2run_clk clock frequency.

Table III.17.9 Setting the Division Ratio for Bus Clock

BCLGDT1	BCLGDT0	BCLG source clock
1	1	subsys_clk / 4
1	0	subsys_clk / 2
0	1	subsys_clk
0	0	

Do not change the BCLGDT1 and BCLGDT0 control bits at same time to avoid occurrence of narrow pulse in the halt2run_clk clock tree.
At initial reset, BCLGDT is set to "0" (subsys_clk).

LCLGEN: LCDC clock generator enable (D7) / LCDC clock generator control register (0x300F34)

Enables/disables the LCDC clock generator.

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

When "1" is written to LCLGEN, the source clock supply is enabled and the LCDC clock generator starts outputting the lcdc_clk clock for the LCD controller. When "0" is written, the LCDC clock generator stops outputting the lcdc_clk clock.
At initial reset, LCLGEN is set to "0" (disabled).

LCLGDT3–LCLGDT0: LCDC clock setup (D[3:0]) / LCDC clock generator control register (0x300F34)

Configures the frequency divider in the LCDC clock generator to set the lcdc_clk clock frequency.

Table III.17.10 Setting the Division Ratio for LCDC Clock

LCLGDT3	LCLGDT2	LCLGDT1	LCLGDT0	LCLG source clock
1	1	1	1	subsys_clk / 16
1	1	1	0	subsys_clk / 15
1	1	0	1	subsys_clk / 14
1	1	0	0	subsys_clk / 13
1	0	1	1	subsys_clk / 12
1	0	1	0	subsys_clk / 11
1	0	0	1	subsys_clk / 10
1	0	0	0	subsys_clk / 9
0	1	1	1	subsys_clk / 8
0	1	1	0	subsys_clk / 7
0	1	0	1	subsys_clk / 6
0	1	0	0	subsys_clk / 5
0	0	1	1	subsys_clk / 4
0	0	1	0	subsys_clk / 3
0	0	0	1	subsys_clk / 2
0	0	0	0	subsys_clk

Do not change the LCLGDT3, LCLGDT2, LCLGDT1, and LCLGDT0 control bits at same time to avoid occurrence of narrow pulse in the LCDC clock tree.
At initial reset, LCLGDT is set to "0" (subsys_clk).

USBCKOF: USB module clock control (D7) / Module clock control register 0 (0x300F35)

Disables the clock supply to the USB module (host interface).

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to USBCKOF, the sbcuclk clock supply to the USB module is disabled and the USB function controller stops operating. When "0" is written, the clock is supplied to the USB module.
This bit does not affect supplying the USB clock (OSC3 clock).
At initial reset, USBCKOF is set to "0" (enabled).

FSIFCK1OF: FSIF module clock control 1 (D6) / Module clock control register 0 (0x300F35)

Disables the clock supply to the serial interface with FIFO.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to FSIFCK1OF, the sbcuclk clock supply to the FSIF module is disabled and the serial interface with FIFO stops operating. When "0" is written, the clock is supplied to the FSIF module.
At initial reset, FSIFCK1OF is set to "0" (enabled).

FSIFCK2OF: FSIF module clock control 2 (D5) / Module clock control register 0 (0x300F35)

Disables the clock supply to the baud-rate timer of the serial interface with FIFO.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to FSIFCK2OF, the periclk clock supply to the FSIF module (baud-rate timer) is disabled. When "0" is written, the clock is supplied to the FSIF module.
This clock can be disabled when an external transfer clock is used in the serial interface with FIFO, as disabling does not affect the serial transfer operation.
At initial reset, FSIFCK2OF is set to "0" (enabled).

SPICKOF: SPI module clock control (D4) / Module clock control register 0 (0x300F35)

Disables the clock supply to the SPI module.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to SPICKOF, the sbcuclk clock supply to the SPI module is disabled and the SPI circuit stops operating. When "0" is written, the clock is supplied to the SPI module.
At initial reset, SPICKOF is set to "0" (enabled).

SMIFCKOF: SMIF module clock control (D3) / Module clock control register 0 (0x300F35)

Disables the clock supply to the NAND flash interface module.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to SMIFCKOF, the sbcuclk clock supply to the NAND flash interface module is disabled and the NAND flash interface circuit stops operating. When "0" is written, the clock is supplied to the NAND flash interface module.
At initial reset, SMIFCKOF is set to "0" (enabled).

SQCKOF: SQROM module clock control (D2) / Module clock control register 0 (0x300F35)

Disables the clock supply to the sequential ROM interface module.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to SQCKOF, the sbcuclk clock supply to the sequential ROM interface module is disabled and the sequential ROM interface circuit stops operating. When "0" is written, the clock is supplied to the sequential ROM interface module.
At initial reset, SQCKOF is set to "0" (enabled).

ARBCKOF: Bus arbiter module clock control (D1) / Module clock control register 0 (0x300F35)

Disables the clock supply to the bus arbiter module.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to ARBCKOF, the sbcuclk clock supply to the bus arbiter module is disabled and the bus arbiter stops operating. When "0" is written, the clock is supplied to the bus arbiter module.
At initial reset, ARBCKOF is set to "0" (enabled).

LHIFCKOF: LCDC host I/F clock control (D0) / Module clock control register 0 (0x300F35)

Disables the clock supply to the LCDC host interface.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to LHIFCKOF, the sbcuclk clock supply to the LCDC host interface is disabled and the LCDC host interface stops operating. When "0" is written, the clock is supplied to the LCDC host interface.
This bit does not affect supplying the LCDC clock generated by the LCDC clock generator.
At initial reset, LHIFCKOF is set to "0" (enabled).

SDRAMCKOF: SDRAMC clock control (D1) / Module clock control register 1 (0x300F36)

Disables the clock supply to the SDRAM controller.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to SDRAMCKOF, the halt2run_clk clock supply to the SDRAM controller is disabled and the SDRAM controller stops operating. When "0" is written, the clock is supplied to the SDRAM controller.
At initial reset, SDRAMCKOF is set to "0" (enabled).

VRAMCKOF: VRAM clock control (D0) / Module clock control register 1 (0x300F36)

Disables the clock supply to the internal VRAM.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to VRAMCKOF, the halt2run_clk clock supply to the internal VRAM is disabled. When "0" is written, the clock is supplied to the internal VRAM.
At initial reset, VRAMCKOF is set to "0" (enabled).

SUBCKOF: OSC3/PLL clock disable (D0) / Sub system clock control register (0x300F37)

Disables the OSC3/PLL output clock supply to the sub system clock selector.

Write "1": Disables
Write "0": Enables
Read: Valid

When "1" is written to SUBCKOF, the OSC3 clock and PLL output clock are not supplied to the sub system clock selector for the bus clock generator and LCDC clock generator. This setting reduces current consumption, note, however, the SDRAM controller and LCD controller cannot be used. When SUBCKOF is set to "0", these clocks are supplied to the sub system clock selector.
At initial reset, SUBCKOF is set to "0" (enabled).

BSLSEL: A0/#BSL mode select (D0) / A0/#BSL select register (0x300F38)

Select the interface method of the extended peripheral block.

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

When the #BSL system is selected in the BCU, write "1" to BSLSEL. When the A0 system is selected in the BCU, write "0" to BSLSEL. Be sure to set the interface method for the BCU using the SBUSST (D3) / Bus control register (0x4812E) before setting BSLSEL.
At initial reset, BSLSEL is set to "0" (A0 mode).

SDRAMX2: SDRAM x2 speed mode disable (D1) / x2 speed mode control register (0x300F39)

Sets the bus speed mode for the SDRAM.

Write "1": x1 speed mode
Write "0": x2 speed mode
Read: Valid

When "1" is written to SDRAMX2, x1 speed mode (CPU - bus clock ratio is 1 : 1) is set. The bus clock for the SDRAM/internal VRAM and the CPU system clock will be the same.
When "0" is written, x2 speed mode (CPU - bus clock ratio is 2 : 1) is set. The bus clock frequency for the SDRAM/internal VRAM becomes half of the CPU system clock.
However, this control is effective only when the #X2SPD pin is set to "0" (x2 speed mode), otherwise the speed mode is fixed at x1.
At initial reset, SDRAMX2 is set to "0" (x2 speed mode).

Programming Notes

(1) Before changing the System clock division register (0x300F30) and/or PLL control register (0x300F31), the clock source of the CPU and peripheral operating clocks must be switched to OSC1. Otherwise, the CPU malfunctions.

(2) When changing the divider setting using SYSCLK[1:0] (D[1:0]/0x300F30), BCLGDT[1:0] (D[1:0]/0x300F33), or LCLGDT[3:0] (D[3:0]/0x300F34), these control bits must be changed in bit units one by one to avoid occurrence of narrow pulses on the clock tree. For example, when changing the set value from 0b00 to 0b11, first write 0b01 (or 0b10) then write 0b11. Changing two or more bits at the same time (0b00 to 0b11 in this example) may cause a malfunction of the circuits that use the changed clock.

IV ANALOG BLOCK

IV-1 INTRODUCTION

The analog block consists of an A/D converter with five input channels.

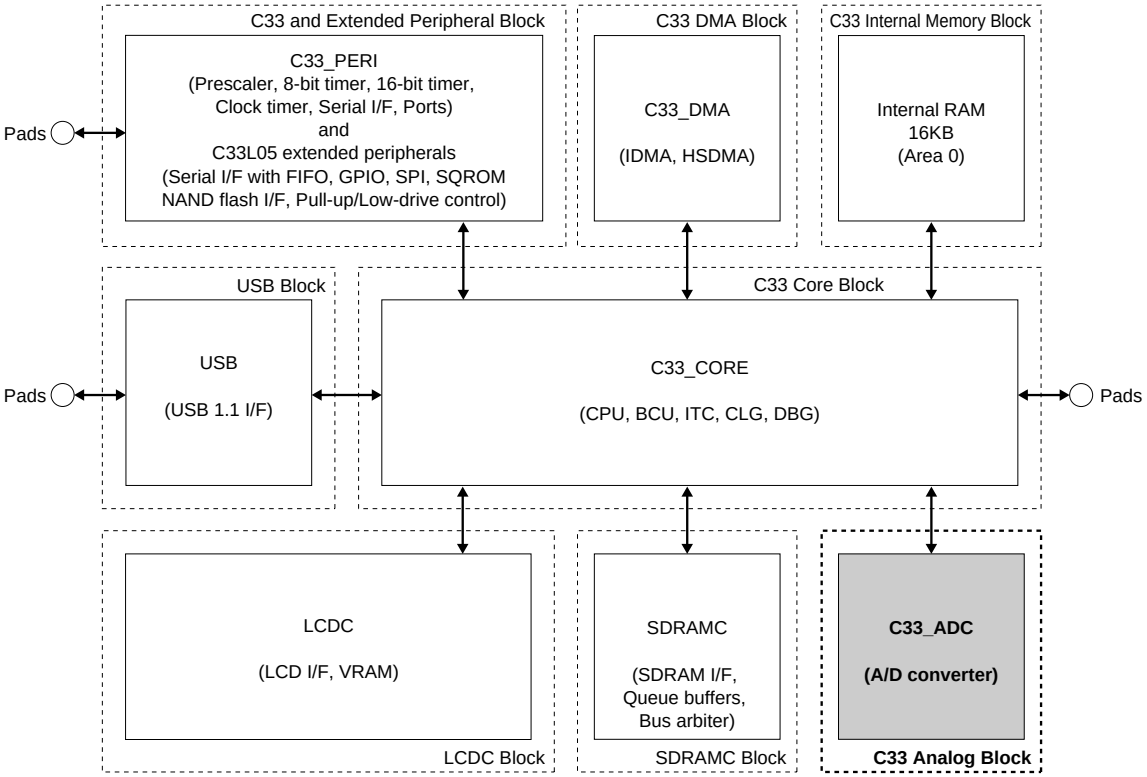


Figure IV.1.1 Analog Block

Note: The C33 Analog Block supports eight input channels (AD0–AD7). Note, however, that the S1C33L05 provides five analog input pins (AD0–AD4). AD5–AD7 cannot be used.

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 5
- A/D converter input clock: Maximum of 2 MHz, minimum of 16 kHz
- Conversion time: Minimum of 10 µs (when a 2-MHz input clock is selected)
  Maximum of 1250 µs (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.

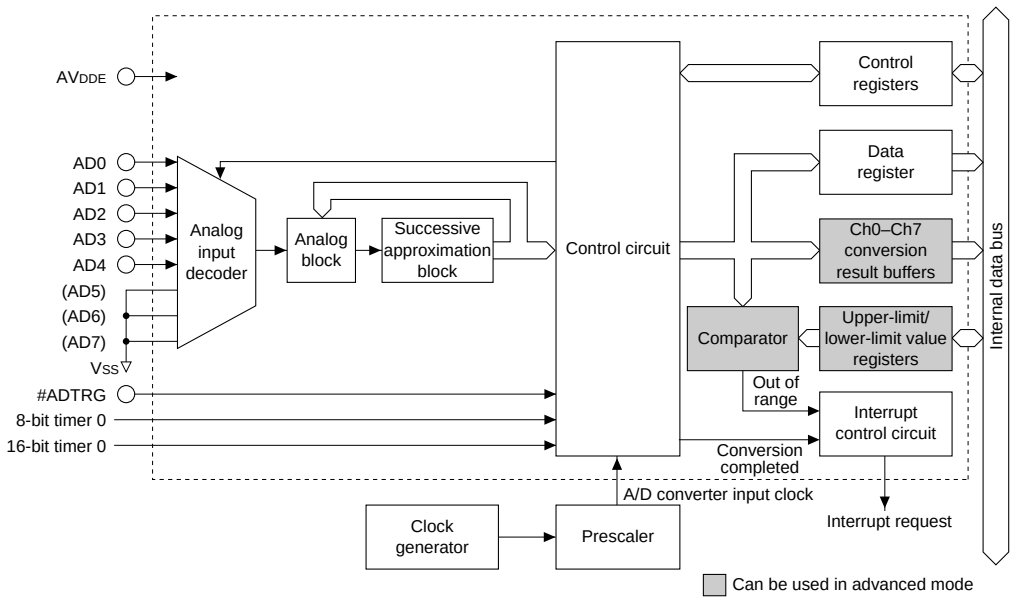


Figure IV.2.1 Structure of A/D Converter

Note: The C33 Analog Block supports a maximum of eight input channels (AD0–AD7). Note, however, that this chip supports 5 channels only. Be aware that the AD5–AD7 ports return invalid conversion results and the K65–K67 ports returns invalid input data.

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(**#ADTRG**)	I	Input port / AD trigger	CFK52(D2)/K5 function select register(0x402C0)
K60(**AD0**)	I	Input port / AD converter input 0	CFK60(D0)/K6 function select register(0x402C3)
K61(**AD1**)	I	Input port / AD converter input 1	CFK61(D1)/K6 function select register(0x402C3)
K62(**AD2**)	I	Input port / AD converter input 2	CFK62(D2)/K6 function select register(0x402C3)
K63(**AD3**)	I	Input port / AD converter input 3	CFK63(D3)/K6 function select register(0x402C3)
K64(**AD4**)	I	Input port / AD converter input 4	CFK64(D4)/K6 function select register(0x402C3)
AVDDE	–	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[4:0] (analog-signal input pins)

The analog input pins AD4 (Ch.4) through AD0 (Ch.0) are shared with input port pins K64 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[4:0] 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	f_{PSCIN}/256
1	1	0	f_{PSCIN}/128
1	0	1	f_{PSCIN}/64
1	0	0	f_{PSCIN}/32
0	1	1	f_{PSCIN}/16
0	1	0	f_{PSCIN}/8
0	0	1	f_{PSCIN}/4
0	0	0	f_{PSCIN}/2

f_{PSCIN}: 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 out-of-range 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".

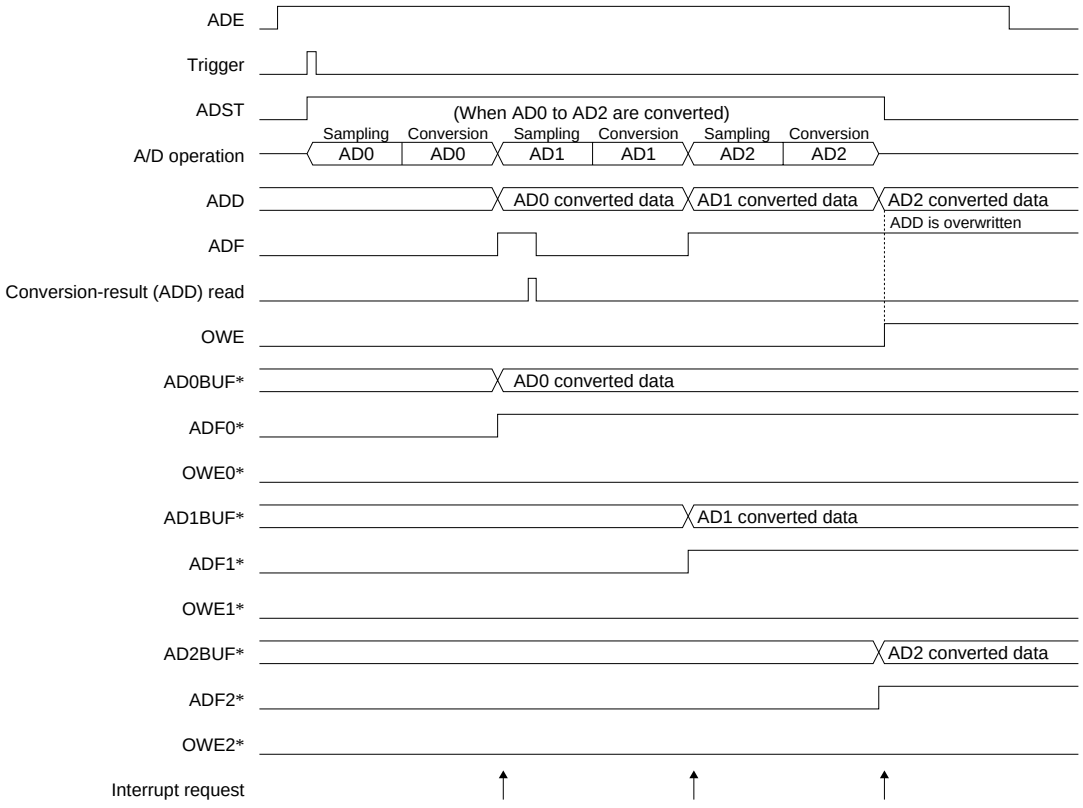
4. Masking conversion-complete interrupt for the specified channels

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.

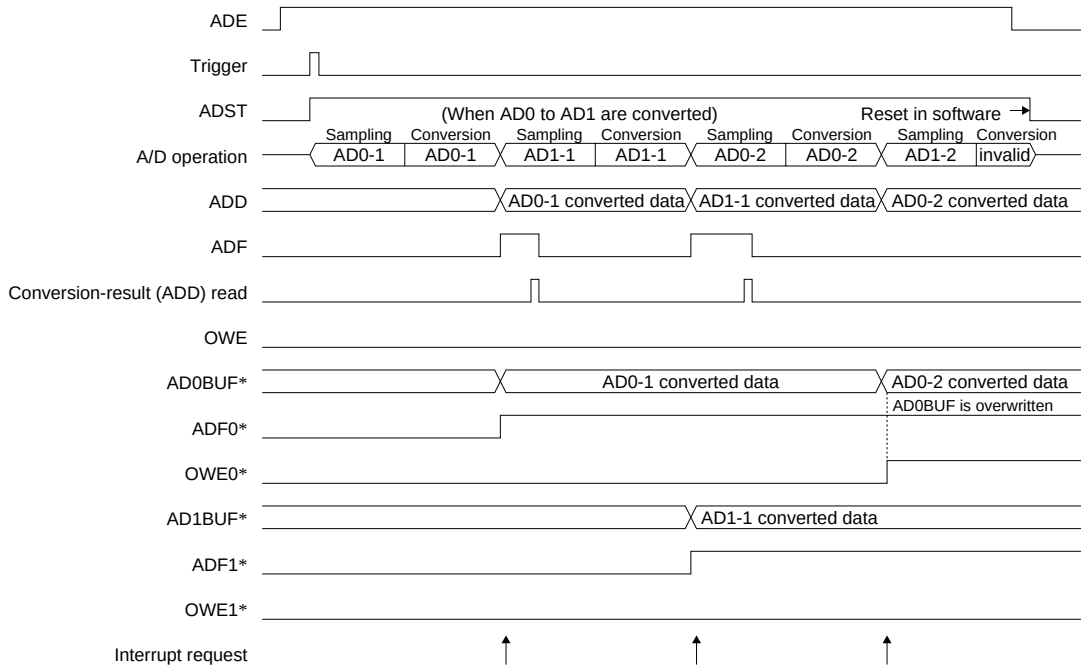
At initial reset, the INTMASK bits are all set to "1" to enable conversion-complete interrupts.

Control and Operation of A/D Conversion

Figure IV.2.2 shows the operation of the A/D converter.



(1) Normal mode



(2) Continuous mode

∗ 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-orange 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	0x0 to 0x3FF	0	R	
		D6	**ADD6**	(low-order 8 bits)	(low-order 8 bits)	0		
		D5	**ADD5**	ADD0 = LSB		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	0x0 to 0x3FF	0	R	
		D0	**ADD8**	(high-order 2 bits) ADD9 = MSB	(high-order 2 bits)	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			
A/D channel register	**0040243 (B)**	D7–6	**–**	–	–	–	–	0 when being read.
		D5	**CE2**	A/D converter	CE[2:0] End channel	0	R/W	
		D4	**CE1**	end channel selection	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	CS[2:0] Start channel	0	R/W	
		D1	**CS1**	start channel selection	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**	Upper/lower limit 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 1 = 9 clocks; 1 0 = 7 clocks; 0 1 = 5 clocks; 0 0 = 3 clocks	1	R/W	Use with 9 clocks.
		D0	**ST0**			1		
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".
		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	Reset when ADBUFx is read.
		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".
		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	Reset by writing 0.
		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) AD0BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	**AD0BUF8**			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) AD1BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	**AD1BUF8**			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) AD2BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	**AD2BUF8**			0		

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**			0		
		D5	**AD3BUF5**			0		
		D4	**AD3BUF4**			0		
		D3	**AD3BUF3**			0		
		D2	**AD3BUF2**			0		
		D1	**AD3BUF1**			0		
		D0	**AD3BUF0**			0		
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**			0		
		D5	**AD4BUF5**			0		
		D4	**AD4BUF4**			0		
		D3	**AD4BUF3**			0		
		D2	**AD4BUF2**			0		
		D1	**AD4BUF1**			0		
		D0	**AD4BUF0**			0		
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**			0		
		D5	**AD5BUF5**			0		
		D4	**AD5BUF4**			0		
		D3	**AD5BUF3**			0		
		D2	**AD5BUF2**			0		
		D1	**AD5BUF1**			0		
		D0	**AD5BUF0**			0		
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**			0		
		D5	**AD6BUF5**			0		
		D4	**AD6BUF4**			0		
		D3	**AD6BUF3**			0		
		D2	**AD6BUF2**			0		
		D1	**AD6BUF1**			0		
		D0	**AD6BUF0**			0		
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**			0		
		D5	**AD7BUF5**			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	–	–	–	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**			0		
		D5	**ADUPR5**			0		
		D4	**ADUPR4**			0		
		D3	**ADUPR3**			0		
		D2	**ADUPR2**			0		
		D1	**ADUPR1**			0		
		D0	**ADUPR0**			0		

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) ADUPR9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R/W	Can be used when ADCADV = "1".
		D0	ADUPR8			0		
A/D lower limit value (low-order) register	004025A (B)	D7	ADLWR7	A/D conversion lower limit value (low-order 8 bits) ADLWR0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R/W	Can be used when ADCADV = "1".
		D6	ADLWR6			0		
		D5	ADLWR5			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	–	–	–	0 when being read.
		D1	ADLWR9	A/D conversion lower limit value (high-order 2 bits) ADLWR9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R/W	Can be used when ADCADV = "1".
		D0	ADLWR8			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			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		
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	
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	
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
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	
K6 function select register	00402C3 (B)	D7	CFK67	K67 function selection	1	AD7	0	K67	0	R/W	
		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	

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. ADD8 and ADD9 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 ADFx = "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, ADFx must be reset by reading out the converted data before the next A/D conversion is completed.
At initial reset, ADFx 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_{DDE} and V_{SS}, 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 θ/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, S1C33L05)

(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.

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.

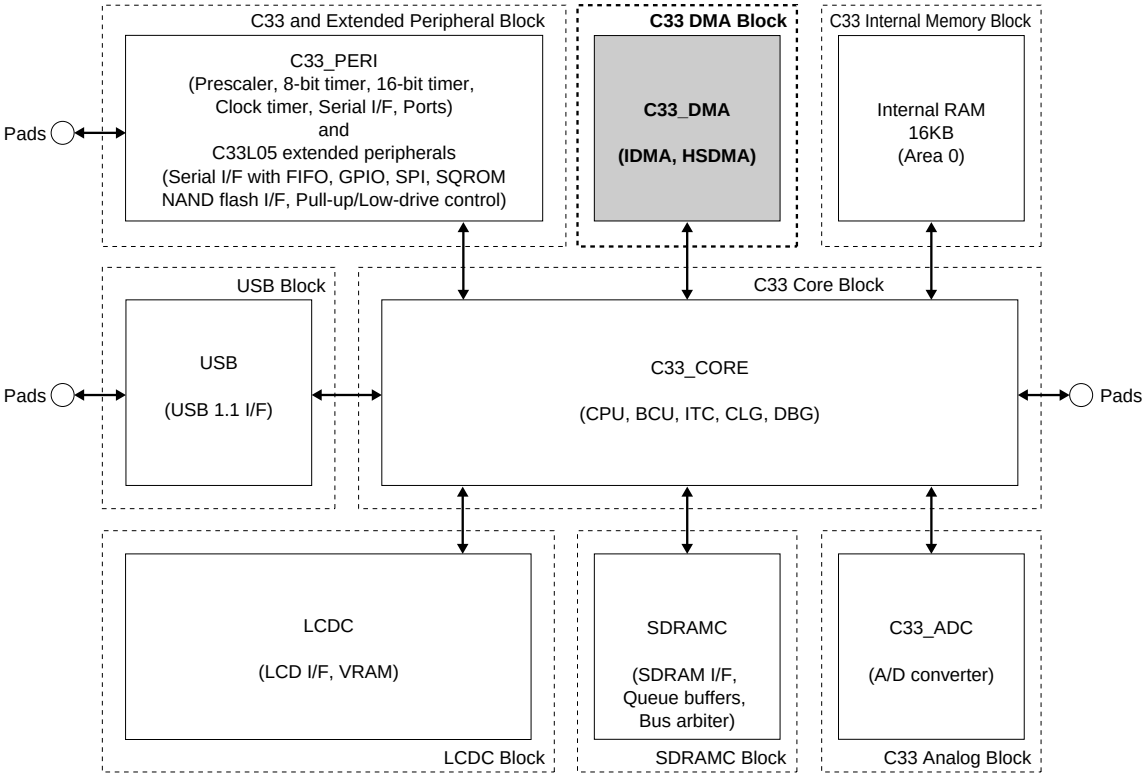


Figure V.1.1 DMA Block

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.

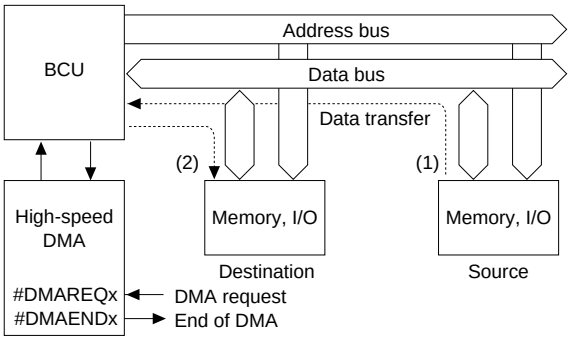


Figure V.2.1 Dual-Address Transfer Method

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.

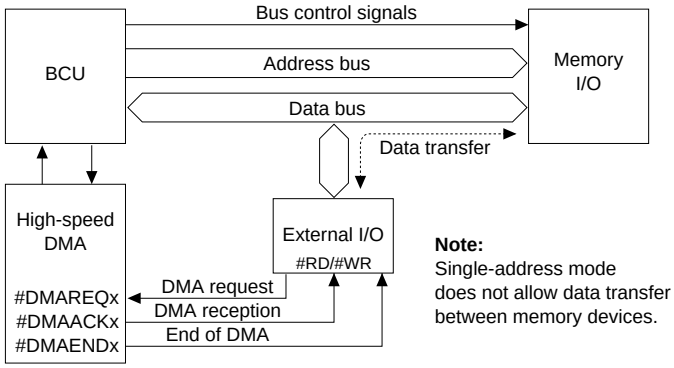


Figure V.2.2 Single-Address Transfer Method

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 DATSIZEx) should be set using the BLKLENx[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 DATSIZEx 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 DATSIZEx 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 DATSIZEx 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 seffing 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 bits7–0 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 DATSIZEx. 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.

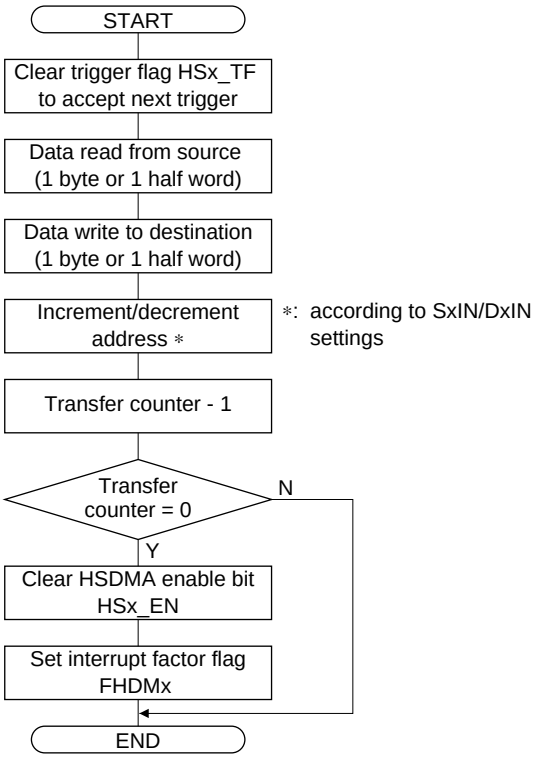


Figure V.2.3 Operation Flow in Single Transfer Mode

(1) When a trigger is accepted, the trigger flag 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 addresses are incremented or decremented according to the SxIN/DxIN settings.
(4) The transfer counter is decremented.
(5) 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").

Successive transfer mode

The channel for which DxMOD 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.

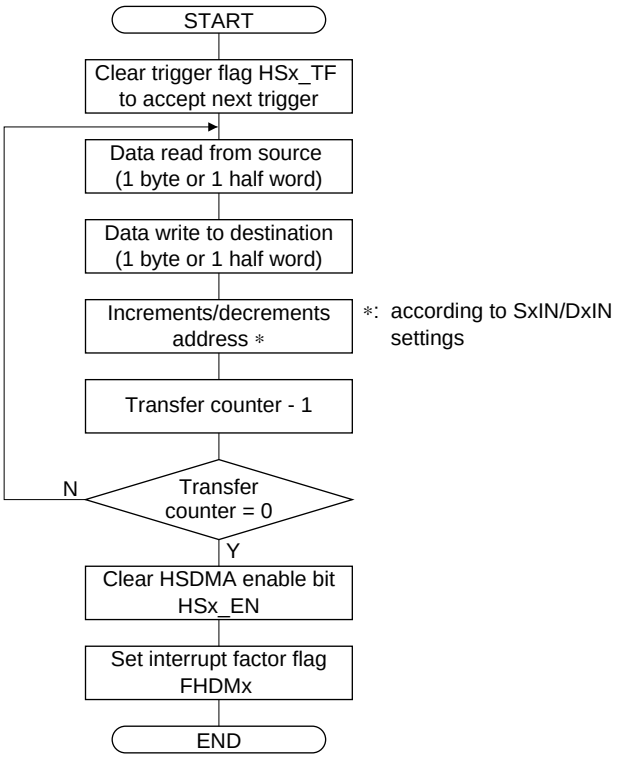


Figure V.2.4 Operation Flow in Successive Transfer Mode

(1) When a trigger is accepted, the trigger flag 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 addresses are incremented or decremented according to the SxIN/DxIN 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 HSx_EN is cleared and HSDMA interrupt factor flag in ITC is set when the transfer counter reaches 0 (when DINTENx = "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.

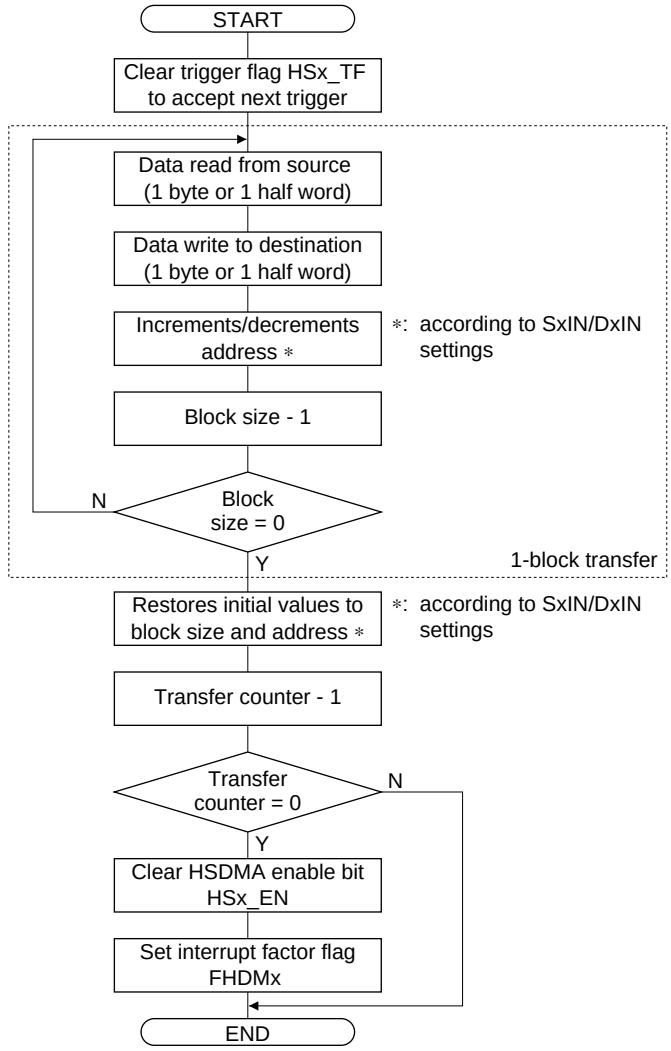


Figure V.2.5 Operation Flow in Block Transfer Mode

(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 BLKLENx 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

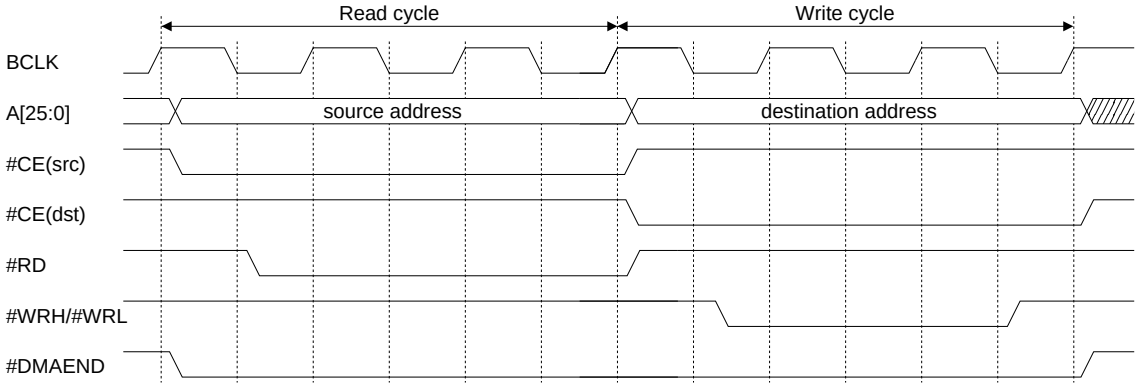


Figure V.2.6 #DMAEND Signal Output Timing (SRAM)

Single-address mode

(1) SRAM

Example: When 2 (RD)/1 (WR) wait cycles are inserted

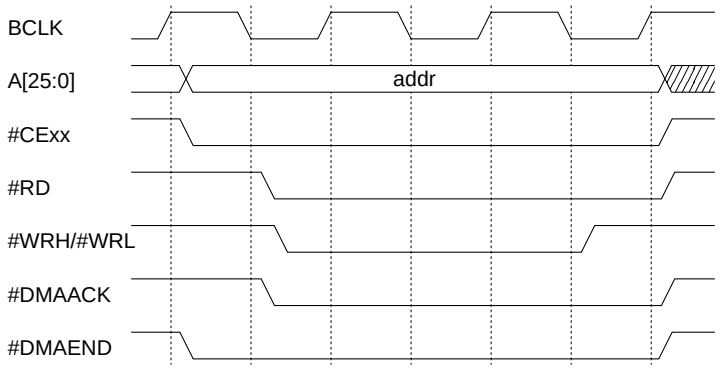


Figure V.2.7 #DMAACK/#DMAEND Signal Output Timing (SRAM)

(2) Burst ROM

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

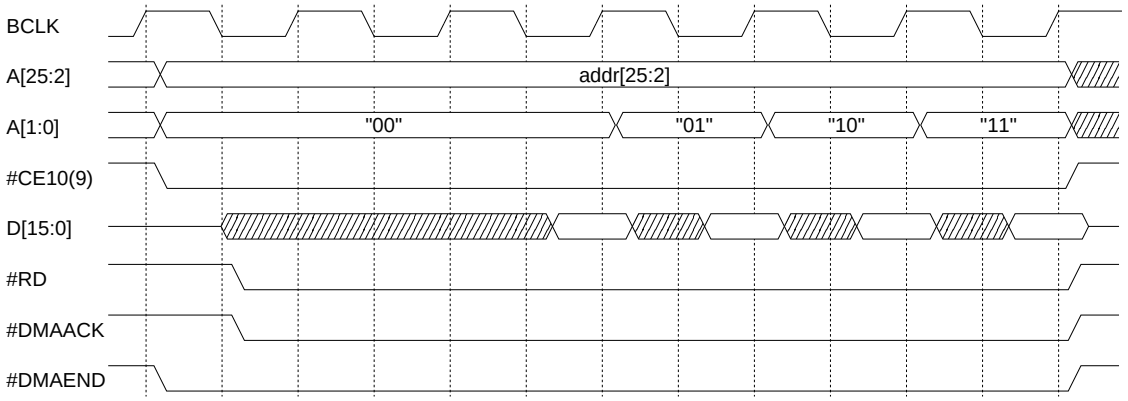


Figure V.2.8 #DMAACK/#DMAEND Signal Output Timing (Burst ROM)

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	0 to 7	X	R/W	
		D5	**PHSD1L1**	interrupt level		X		
		D4	**PHSD1L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PHSD0L2**	High-speed DMA Ch.0	0 to 7	X	R/W	
		D1	**PHSD0L1**	interrupt level		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	0 to 7	X	R/W	
		D5	**PHSD3L1**	interrupt level		X		
		D4	**PHSD3L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PHSD2L2**	High-speed DMA Ch.2	0 to 7	X	R/W	
		D1	**PHSD2L1**	interrupt level		X		
		D0	**PHSD2L0**			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	0	Software trigger	0	R/W	
		D6	**HSD1S2**	trigger set-up	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	0	Software trigger	0	R/W	
		D2	**HSD0S2**	trigger set-up	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			
High-speed DMA Ch.2/3 trigger set-up register	**0040299 (B)**	D7	**HSD3S3**	High-speed DMA Ch.3	0	Software trigger	0	R/W	
		D6	**HSD3S2**	trigger set-up	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			
		D3	**HSD2S3**	High-speed DMA Ch.2	0	Software trigger	0	R/W	
		D2	**HSD2S2**	trigger set-up	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	

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	–				–	–	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	
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) Ch.0 transfer counter[15:8] (single/successive 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) Ch.0 transfer counter[7:0] (single/successive transfer mode)					X	R/W	
		D6	BLKLEN06						X		
		D5	BLKLEN05						X		
		D4	BLKLEN04						X		
		D3	BLKLEN03						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 Note: D) Dual address mode S) Single address mode	**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 Note: D) Dual address mode S) Single address mode	**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 Note: D) Dual address mode S) Single address mode	**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 Note: D) Dual address mode S) Single address mode	**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		

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]	0	R/W	
		DE	**D0MOD0**		1 1 Invalid 1 0 Block 0 1 Successive 0 0 Single	0		
		DD	**D0IN1**	D) Ch.0 destination address control S) Invalid	D0IN[1:0]	0	R/W	
		DC	**D0IN0**		1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0		
		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)	1 Clear 0 No operation	0	R/W	
				Ch.0 trigger flag status (reading)	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) Ch.1 transfer counter[15:8] (single/successive transfer mode)		X	R/W	
		DE	**TC1_L6**			X		
		DD	**TC1_L5**			X		
		DC	**TC1_L4**			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) Ch.1 transfer counter[7:0] (single/successive transfer mode)		X	R/W	
		D6	**BLKLEN16**			X		
		D5	**BLKLEN15**			X		
		D4	**BLKLEN14**			X		
		D3	**BLKLEN13**			X		
		D2	**BLKLEN12**			X		
		D1	**BLKLEN11**			X		
		D0	**BLKLEN10**			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) Ch.1 transfer counter[23:16] (single/successive transfer mode)		X	R/W	
		D6	**TC1_H6**			X		
		D5	**TC1_H5**			X		
		D4	**TC1_H4**			X		
		D3	**TC1_H3**			X		
		D2	**TC1_H2**			X		
		D1	**TC1_H1**			X		
		D0	**TC1_H0**			X		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
High-speed DMA Ch.1 low-order source address set-up register Note: D) Dual address mode S) Single address mode	**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 Note: D) Dual address mode S) Single address mode	**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 S) Ch.1 memory address control	S1IN[1:0] Inc/dec	0	R/W	
		DC	**S1IN0**		1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0		
		DB	**S1ADRH11**	D) Ch.1 source address[27:16]		X	R/W	
		DA	**S1ADRH10**	S) Ch.1 memory address[27:16]		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		
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**			X		
		DD	**D1ADRL13**			X		
		DC	**D1ADRL12**			X		
		DB	**D1ADRL11**			X		
		DA	**D1ADRL10**			X		
		D9	**D1ADRL9**			X		
		D8	**D1ADRL8**			X		
		D7	**D1ADRL7**			X		
		D6	**D1ADRL6**			X		
		D5	**D1ADRL5**			X		
		D4	**D1ADRL4**			X		
		D3	**D1ADRL3**			X		
		D2	**D1ADRL2**			X		
		D1	**D1ADRL1**			X		
		D0	**D1ADRL0**			X		

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	**D1MOD1**	Ch.1 transfer mode	D1MOD[1:0]	0	R/W	
		DE	**D1MOD0**		1 1 Invalid 1 0 Block 0 1 Successive 0 0 Single	0		
		DD	**D1IN1**	D) Ch.1 destination address control S) Invalid	D1IN[1:0]	0	R/W	
		DC	**D1IN0**		1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0		
		DB	**D1ADRH11**	D) Ch.1 destination address[27:16] S) Invalid		X	R/W	
		DA	**D1ADRH10**			X		
		D9	**D1ADRH9**			X		
		D8	**D1ADRH8**			X		
		D7	**D1ADRH7**			X		
		D6	**D1ADRH6**			X		
		D5	**D1ADRH5**			X		
		D4	**D1ADRH4**			X		
		D3	**D1ADRH3**			X		
		D2	**D1ADRH2**			X		
		D1	**D1ADRH1**			X		
		D0	**D1ADRH0**			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	**TC2_L7**	Ch.2 transfer counter[7:0] (block transfer mode) Ch.2 transfer counter[15:8] (single/successive transfer mode)		X	R/W	
		DE	**TC2_L6**			X		
		DD	**TC2_L5**			X		
		DC	**TC2_L4**			X		
		DB	**TC2_L3**			X		
		DA	**TC2_L2**			X		
		D9	**TC2_L1**			X		
		D8	**TC2_L0**			X		
		D7	**BLKLEN27**	Ch.2 block length (block transfer mode) Ch.2 transfer counter[7:0] (single/successive transfer mode)		X	R/W	
		D6	**BLKLEN26**			X		
		D5	**BLKLEN25**			X		
		D4	**BLKLEN24**			X		
		D3	**BLKLEN23**			X		
		D2	**BLKLEN22**			X		
		D1	**BLKLEN21**			X		
		D0	**BLKLEN20**			X		
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) Ch.2 transfer counter[23:16] (single/successive transfer mode)		X	R/W	
		D6	**TC2_H6**			X		
		D5	**TC2_H5**			X		
		D4	**TC2_H4**			X		
		D3	**TC2_H3**			X		
		D2	**TC2_H2**			X		
		D1	**TC2_H1**			X		
		D0	**TC2_H0**			X		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
High-speed DMA Ch.2 low-order source address set-up register Note: D) Dual address mode S) Single address mode	**0048244 (HW)**	DF	**S2ADRL15**	D) Ch.2 source address[15:0] S) Ch.2 memory address[15:0]		X	R/W	
		DE	**S2ADRL14**			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 Note: D) Dual address mode S) Single address mode	**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 S) Ch.2 memory address control	S2IN[1:0] Inc/dec 1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0 0	R/W	
		DC	**S2IN0**					
		DB	**S2ADRH11**	D) Ch.2 source address[27:16] S) Ch.2 memory address[27:16]		X	R/W	
		DA	**S2ADRH10**			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 Note: D) Dual address mode S) Single address mode	**0048248 (HW)**	DF	**D2ADRL15**	D) Ch.2 destination address[15:0] S) Invalid		X	R/W	
		DE	**D2ADRL14**			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		

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	**D2MOD1**	Ch.2 transfer mode	D2MOD[1:0]: 1 1 Invalid; 1 0 Block; 0 1 Successive; 0 0 Single	0	R/W	
		DE	**D2MOD0**			0		
		DD	**D2IN1**	D) Ch.2 destination address control S) Invalid	D2IN[1:0]: 1 1 Inc.(no init); 1 0 Inc.(init); 0 1 Dec.(no init); 0 0 Fixed	0	R/W	
		DC	**D2IN0**			0		
		DB	**D2ADRH11**	D) Ch.2 destination address[27:16] S) Invalid		X	R/W	
		DA	**D2ADRH10**			X		
		D9	**D2ADRH9**			X		
		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		
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	**TC3_L7**	Ch.3 transfer counter[7:0] (block transfer mode) Ch.3 transfer counter[15:8] (single/successive transfer mode)		X	R/W	
		DE	**TC3_L6**			X		
		DD	**TC3_L5**			X		
		DC	**TC3_L4**			X		
		DB	**TC3_L3**			X		
		DA	**TC3_L2**			X		
		D9	**TC3_L1**			X		
		D8	**TC3_L0**			X		
		D7	**BLKLEN37**	Ch.3 block length (block transfer mode) Ch.3 transfer counter[7:0] (single/successive transfer mode)		X	R/W	
		D6	**BLKLEN36**			X		
		D5	**BLKLEN35**			X		
		D4	**BLKLEN34**			X		
		D3	**BLKLEN33**			X		
		D2	**BLKLEN32**			X		
		D1	**BLKLEN31**			X		
		D0	**BLKLEN30**			X		
High-speed DMA Ch.3 control register Note: D) Dual address mode S) Single address mode	**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) Ch.3 transfer counter[23:16] (single/successive transfer mode)		X	R/W	
		D6	**TC3_H6**			X		
		D5	**TC3_H5**			X		
		D4	**TC3_H4**			X		
		D3	**TC3_H3**			X		
		D2	**TC3_H2**			X		
		D1	**TC3_H1**			X		
		D0	**TC3_H0**			X		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
High-speed DMA Ch.3 low-order source address set-up register Note: D) Dual address mode S) Single address mode	**0048254 (HW)**	DF	**S3ADRL15**	D) Ch.3 source address[15:0] S) Ch.3 memory address[15:0]		X	R/W	
		DE	**S3ADRL14**			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 Note: D) Dual address mode S) Single address mode	**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 S) Ch.3 memory address control	S3IN[1:0] Inc/dec: 1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0	R/W	
		DC	**S3IN0**			0		
		DB	**S3ADRH11**	D) Ch.3 source address[27:16] S) Ch.3 memory address[27:16]		X	R/W	
		DA	**S3ADRH10**			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 Note: D) Dual address mode S) Single address mode	**0048258 (HW)**	DF	**D3ADRL15**	D) Ch.3 destination address[15:0] S) Invalid		X	R/W	
		DE	**D3ADRL14**			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		

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	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**			X		
		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			

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.

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.

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 DATSIZEx 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 BLKLENx.
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, TCx_L[7:0] is bits[7:0] of the transfer counter, and TCx_H[7:0] is bits[15:8] of the transfer counter.
In single or successive transfer mode, TCx_L[7:0] is bits[15:8] of the transfer counter, and TCx_H[7:0] is bits[23:16] of the transfer counter. The 8 low-order bits are specified by BLKLENx[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 #DMAENDx pin at the same time.
Even when the counter is 0, a DMA request is accepted and the counter is decremented to "0xFFFF" (or "0xFFFFFF").
Be sure to disable DMA transfers (HSx_EN = "0") before writing and reading to and from the counter.
At initial reset, these bits are not initialized.

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.

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 performed. Reading the control information can always be done.

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 LNKEN = "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)
	Timer 5 comparison B	17	R16TU5 (D0/0x40292)	DE16TU5 (D0/0x40296)
	Timer 5 comparison A	18	R16TC5 (D1/0x40292)	DE16TC5 (D1/0x40296)
8-bit programmable timer 0–3	Timer 0 underflow	19	R8TU0 (D2/0x40292)	DE8TU0 (D2/0x40296)
	Timer 1 underflow	20	R8TU1 (D3/0x40292)	DE8TU1 (D3/0x40296)
	Timer 2 underflow	21	R8TU2 (D4/0x40292)	DE8TU2 (D4/0x40296)
	Timer 3 underflow	22	R8TU3 (D5/0x40292)	DE8TU3 (D5/0x40296)
Serial interface Ch.0/1	Ch.0 receive buffer full	23	RSRX0 (D6/0x40292)	DESRX0 (D6/0x40296)
	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)
A/D converter	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	RP4 (D6/0x40293)	DEP4 (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)
LCD controller	Camera, JPEG, etc.	43	RLCDCI (D5/0x402AC)	DELCDCI (D5/0x402AE)
USB function controller	SIE, EPr, DMA, FIFO, etc.	44	RUSBI (D6/0x402AC)	DEUSBI (D6/0x402AE)
SPI	End of transfer, etc.	45	RSPII (D7/0x402AC)	DESPII (D7/0x402AE)
FIFO serial interface Ch.0	Ch.0 receive buffer full	56	RFSRX0 (D2/0x402B3)	DEFSRX0 (D2/0x402B4)
	Ch.0 transmit buffer empty	57	RFSTX0 (D3/0x402B3)	DEFSTX0 (D3/0x402B4)

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.

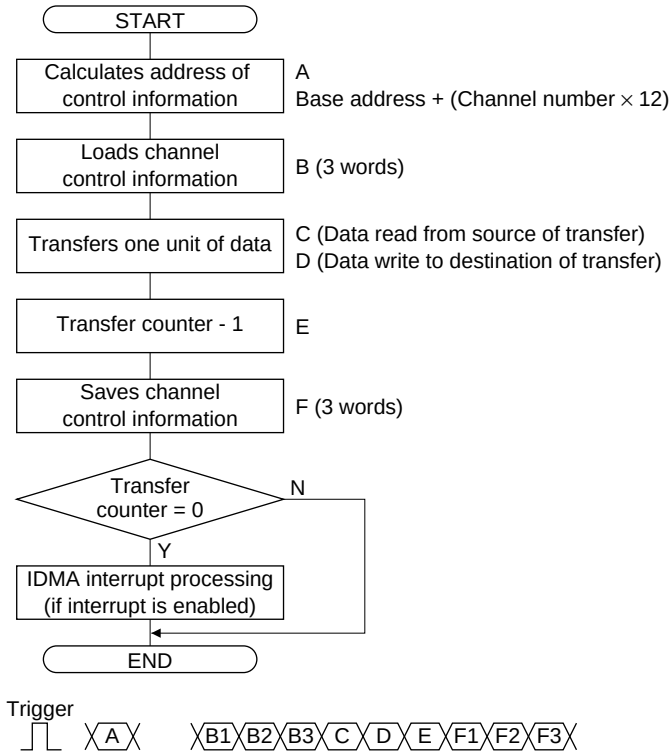


Figure V.3.1 Operation Flow in Single Transfer Mode

(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.

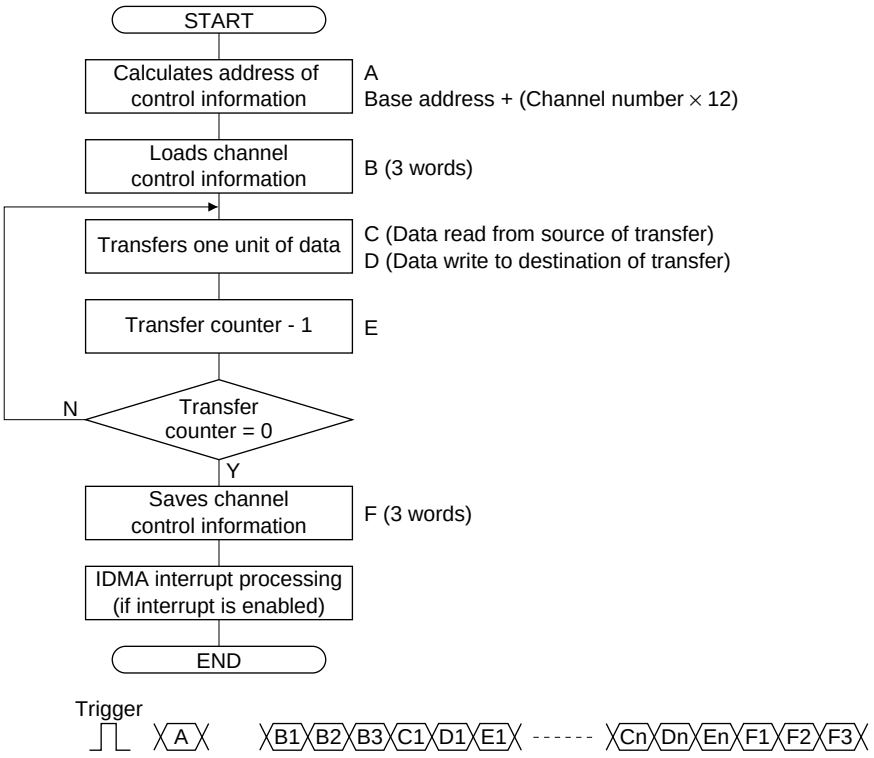


Figure V.3.2 Operation Flow in Successive Transfer Mode

(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.

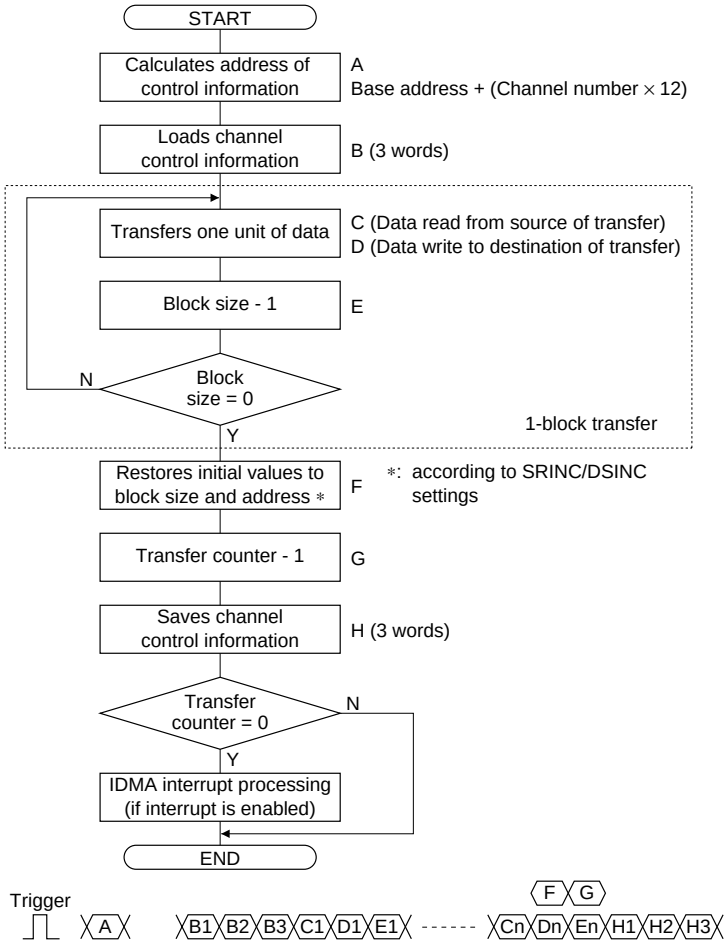


Figure V.3.3 Operation Flow in Block Transfer Mode

(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.

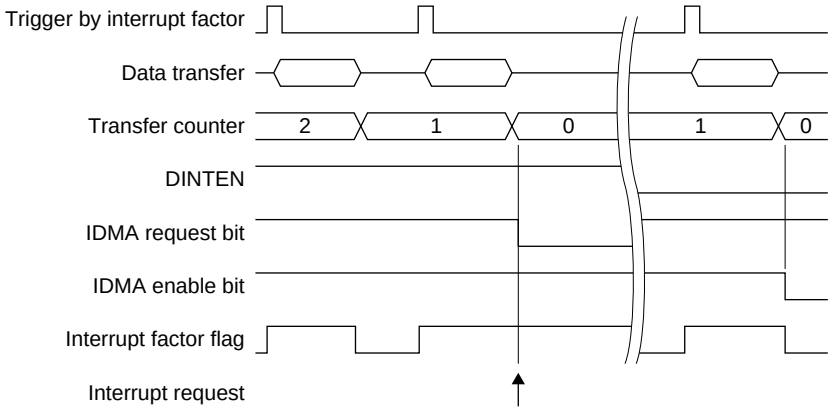


Figure V.3.4 Operation when Invoked by Interrupt Factor

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.

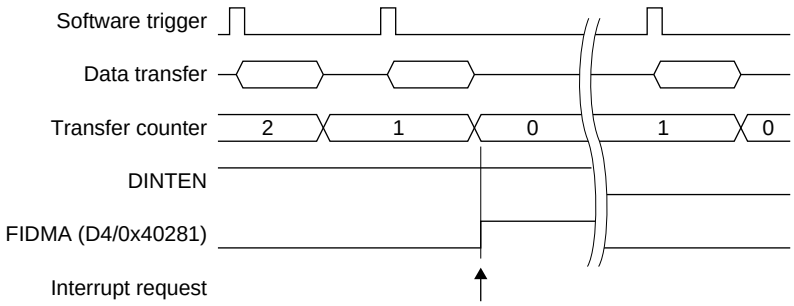


Figure V.3.5 Operation when Invoked by Software Trigger

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.

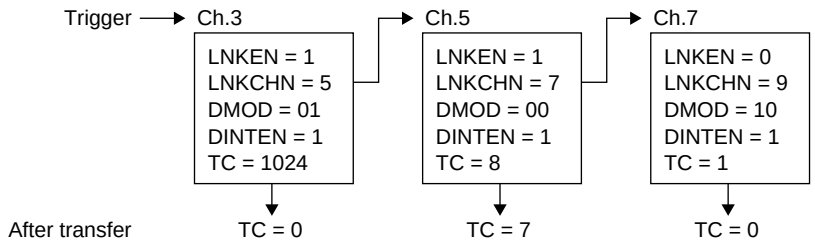


Figure V.3.6 Example of Link Setting

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**	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	
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	

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: IDMA enable (D0) / IDMA 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 <IDMA control information read cycles> + <IDMA transfer read/write cycles> + <IDMA control information write cycles>.

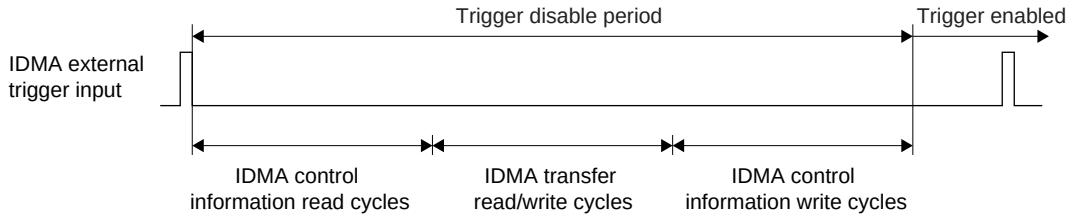


Figure V.3.7 External Trigger Input Interval

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.

VI USB BLOCK

VI-1 INTRODUCTION

The USB block consists of a USB function controller.

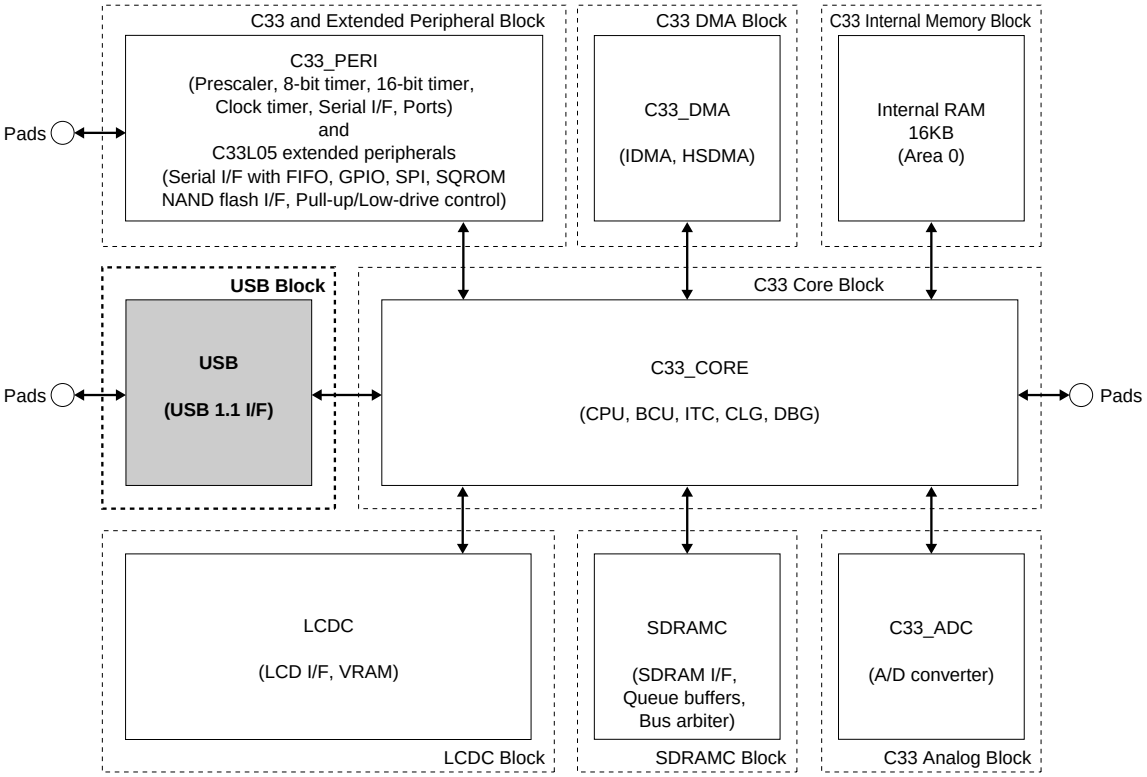


Figure VI.1.1 USB Block

THIS PAGE IS BLANK.

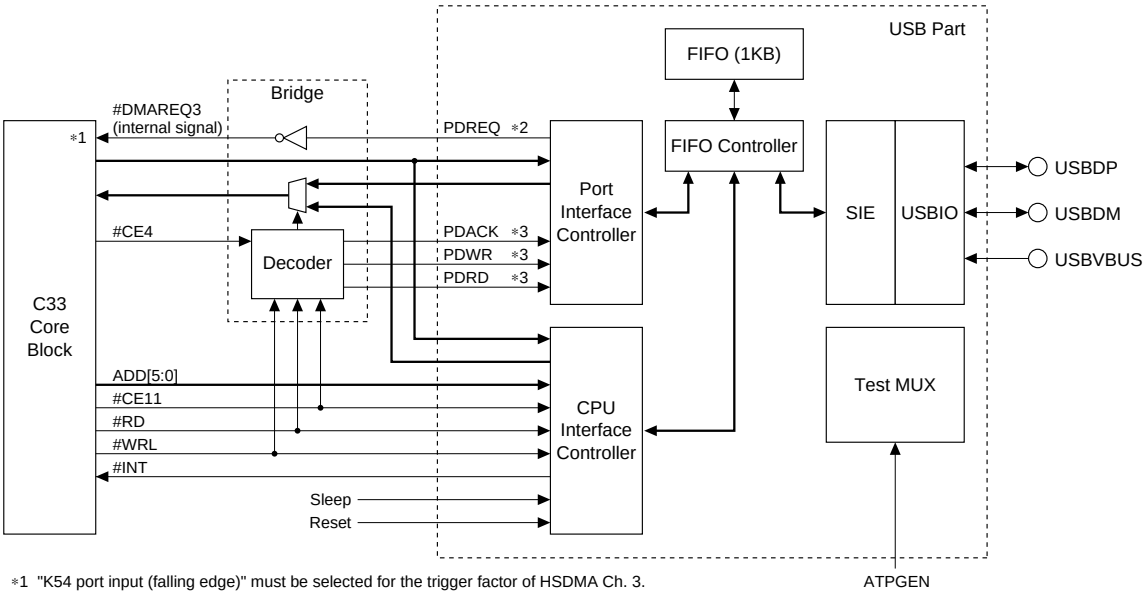
VI-2 USB FUNCTION CONTROLLER

Outline of the USB Function Controller

The S1C33L05 has a built-in USB function controller that supports the Full-Speed mode defined in the USB1.1 Specification. The features are shown below.

- Supports transfer at FS (12 Mbps).
- Supports control, bulk, isochronous and interrupt transfers.
- Supports four general-purpose endpoints and endpoint 0.
- Incorporate 1KB programmable FIFO for endpoints.
- Equipped with a general-purpose DMA port.
- Supports asynchronous procedures.
  Supports a slave configuration.
  Can be used with a bus width of 8 bits.
- Inputs 48 MHz clock.
- Supports snooze mode.

Figure VI.2.1 shows the block diagram of the USB function controller.



∗1 "K54 port input (falling edge)" must be selected for the trigger factor of HSDMA Ch. 3.
∗2 The PDREQ (→#DMAREQ3) signal level must be configured as "Active High".
∗3 The PDACK (←#CE4), PDWR (←#WRL) and PDRD (←#RD) signal levels must be configured as "Active Low".

Figure VI.2.1 USB Function Controller Block Diagram

Serial Interface Engine (SIE)

The SIE manages transactions and generates packets. It also controls bus events such as suspend, resume and reset operations.

FIFO

This is a 1KB buffer for endpoints.

FIFO Controller

This controller performs FIFO SRAM address management (user-programmable), timing generation, arbitration and more.

Port Interface Controller

This controller performs asynchronous handshakes.

CPU Interface Controller

This controller controls timings of the CPU interface and enables register access.

Test Mux

Switches the operational mode (test mode) using the Input signal.

Pins for the USB Interface

Table VI.2.1 list the pins used for the USB interface.

Table VI.2.1 USB Interface Pins

Pin name	I/O	Pull-up	Function
USBDP	I/O	–	USB D+ pin
USBDM	I/O	–	USB D- pin
USBVBUS	I	–	USB VBUS pin. Allows input of 5 V.

Setting Up the BCU, Clock Control, and DMA Registers

Setting the Access Conditions for Areas 11 and 4

The USB control registers and the USB DMA area are mapped into Area 11 and Area 4, respectively. Therefore, in order for the USB function controller to be accessed, the BCU register for Areas 11 and 4 must be set up in accordance with the procedure described below.

1. A12IO (DC) / Access control register (0x48132) = "1"
   A5IO (D8) / Access control register (0x48132) = "1"
   Areas 12–11 and 5–4 are configured for internal devices.
2. A12EC (D4) / Access control register (0x48132) = "0"
   A5EC (D0) / Access control register (0x48132) = "0"
   Areas 12–11 and 5–4 are configured for little endian access.
3. A12SZ (D6) / Areas 12–11 setup register (0x48124) = "1"
   The device size of Areas 12–11 is set to 8 bits.
4. A12WT[2:0] (D[2:0]) / Areas 12–11 setup register (0x48124) = "010" or "001"
   The number of wait cycles for accessing Areas 12–11 (USB control registers) is set to 2 (when the CPU is running with a 40-MHz or higher clock) or 1 (when the CPU is running with a less than 40-MHz clock and in x1 speed mode).
5. A5SZ (D6) / Areas 6–4 setup register (0x4812A) = "1"
   The device size of Areas 5–4 is set to 8 bits.
6. A5WT[2:0] (D[2:0]) / Areas 6–4 setup register (0x4812A) = "001"
   The number of wait cycles for accessing Areas 5–4 (USB DMA area) is set to 1.
7. SWAITE (D0) / Bus control register (0x4812E) = "1"
   The wait control using the #WAIT signal is enabled.
8. USBWT[2:0] (D[2:0]) / Macro control register (0x300F20)
   The USB function controller in Area 11 needs a different wait control from other internal blocks and the USB bus wait control bits (USBWT[2:0]) are provided in the macro control register. Wait states selected by USBWT[2:0] will be inserted into the USB register access cycle along with the wait states selected by A12WT[2:0].
   USBWT[2:0] should be set as follows according to the CPU operating clock frequency:

Table VI.2.2 Setting Wait State for USB Bus

USBWT2	USBWT1	USBWT0	CPU clock	Wait cycle
1	1	1	40 MHz or higher	BCU wait setting + 140 ns
1	1	0		BCU wait setting + 120 ns
1	0	1	24 MHz	BCU wait setting + 100 ns
1	0	0		BCU wait setting + 80 ns
0	1	1		BCU wait setting + 60 ns
0	1	0	12 MHz	BCU wait setting + 40 ns
0	0	1		BCU wait setting + 20 ns
0	0	0	6 MHz	BCU wait setting + 0 ns

Note: The bus speed is limited, up to 40 MHz. When running the CPU with a 40 MHz or higher operating clock, set the #X2SPD pin to "0", A5BS (D0/0x4813E) to "0", A12BS (D4/0x4813E) to "0", and A1X1MD (D3/0x4813A) to "1".

Controlling the USB Clock

The USB clock must be supplied to the USB function controller using the macro control register mapped into Area 6. Set up the access condition for Area 6 and the macro control register 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.
4. USBCKOF (D7) / Module clock control register 0 (0x300F35) = "0"
   This turns the USB clock on.
5. USBCLKEN (D4) / Macro control register (0x300F20) = "1"
   The OSC3 clock is supplied to the USB function controller.
6. USBSNZ (D5) / Macro control register (0x300F20) = "0"
   This disables the USB snooze control.

The OSC3 oscillation circuit must be turned on before the USB function controller can be used. Refer to "II-6 CLG (Clock Generator)" and "III-17 S1C33L05 Clock System and Misc. Registers" for details of the clock control.

Setting the DMA Controller

The USB function controller supports data transfer using the HSDMA (High-Speed DMA) function provided by the C33 DMA block. Therefore, HSDMA conditions must be set up before starting USB data transfer. This section describes how the HSDMA control registers are set up. Refer to "V-2 HSDMA (High-Speed DMA)" for details of HSDMA.

1. Setting the signals used for DMA transfer

The USB controller block has the following pads used for DMA data transfer and they are connected to the corresponding C33 Core pads (see Figure VI.2.1).

USB signal		Corresponding C33 Core signal
PDREQ *	(DMA request signal)	#DMAREQ3 (internal signal)
PDACK	(DMA acknowledge signal)	#CE4
PDRD	(Data read signal)	#RD
PDWR	(Data write signal)	#WRL

∗ The PDREQ signal is inverted before it is input to the C33 Core block.

The active level of the USB signals can be selected using the USB register. Set the respective control bits as follows so that the active levels match between the USB function controller and the C33 Core.

PDREQ: PDREQ_Level (D3/0x1000094) = "0" (Active High)
PDACK: PDACK_Level (D2/0x1000094) = "1" (Active Low)
PDRD, PDWR: PDRDWR_Level (D1/0x1000094) = "1" (Active Low)

Note: The USB function controller uses the internal #DMAREQ3 signal (internal K54 signal). The CFK54 (D4) / K5 function select register (0x402C0) must be set to "0". Furthermore, do not enable an interrupt function using the K54 port.

2. Setting the address mode

The HSDMA circuit provides two data transfer mode: dual address transfer and single address transfer modes. The USB function controller supports only the dual address mode. In this mode, 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.

To configure HSDMA Ch. 3 in this mode, set the DUALM3 (DF) / HSDMA Ch. 3 control register (0x48252) to "1".

Note: Do not set the HSDMA Ch.3 to single address mode when using it for USB data transfer.

3. Setting the transfer mode

The USB function controller supports two transfer modes, asynchronous multi-word DMA transfer (slave) mode and asynchronous single-word DMA transfer (slave) mode.

The asynchronous multi-word DMA transfer (slave) mode asserts the PDREQ (#DMAREQ3) signal while the USB FIFO contains data. The CPU cannot determine the amount of data in the FIFO to be transferred in a DMA transfer while the USB is sending/receiving data dynamically (since data in the FIFO is increased/decreased dynamically according to the circumstances of the USB data transfer).

Therefore, set the USB function controller to asynchronous single-word DMA transfer (slave) mode and the C33 DMA controller to single transfer mode with one byte transfer per trigger, and manage the DMA transfer count with the total amount of USB transfer data.

To set the HSDMA into single transfer mode, set the transfer mode select bits D3MOD[1:0] (D[F:E]) / HSDMA Ch.3 high-order destination address set-up register (0x4825A) to "00". In this mode, a transfer operation invoked by one trigger is completed after transferring one unit of data of the size set by DATSIZE3. If data transfer need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

4. Setting the transfer data size

The DATSIZE3 bit (DE) / HSDMA Ch. 3 high-order source address set-up register (0x48256) is used to set the unit size of data to be transferred. Set this bit to "0" (8 bits).

5. Setting the transfer counter

In the single transfer mode, up to 24 bits of transfer count can be specified using the registers below. Set the desired transfer count to these registers.

BLKLEN3[7:0]: Ch. 3 transfer counter [7:0] (D[7:0]) / HSDMA Ch.3 transfer counter register (0x48250)
TC3_L[7:0]: Ch. 3 transfer counter [15:8] (D[F:8]) / HSDMA Ch. 3 transfer counter register (0x48250)
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.

6. Setting the source and destination addresses

In dual-address mode, a source address and a destination address for DMA transfer can be specified using the registers below.

S3ADRL[15:0]: Ch. 3 source address [15:0] (D[F:0]) / Ch. 3 low-order source address set-up register (0x48254)
S3ADRH[11:0]: Ch. 3 source address [27:16] (D[B:0]) / Ch. 3 high-order source address set-up register (0x48256)
D3ADRL[15:0]: Ch. 3 destination address [15:0] (D[F:0]) / Ch. 3 low-order destination address set-up register (0x48258)
D3ADRH[11:0]: Ch. 3 destination address [27:16] (D[B:0]) / Ch. 3 high-order destination address set-up register (0x4825A)

Note: The DMA transfer address for the USB function controller must be located in Area 4 (0x100000 to 0x1FFFFF, 1MB).

7. Setting the address increment/decrement conditions

The source and/or destination addresses can be incremented or decremented when one data transfer is completed. The S3IN[1:0] bits (D[D:C]) / Ch. 3 high-order source address set-up register (0x48256) (for source address) and the D3IN[1:0] bits (D[D:C]) / Ch. 3 high-order destination address set-up register (0x4825A) (for destination address) are used to set this condition.

S3IN/D3IN = "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.

S3IN/D3IN = "01": address decremented
The address is decremented by an amount equal to the data size set by DATSIZE3 when one data transfer is completed. The address that has been decremented during transfer does not return to the initial value.

S3IN/D3IN = "10" or "11": address incremented
The address is incremented by an amount equal to the data size set by DATSIZE3 when one data transfer is completed. The address that has been incremented during transfer does not return to the initial value.
In the single transfer mode, "10" and "11" set the same condition.

8. Selecting the DMA trigger factor

The HSDMA trigger factor for the USB function controller is K54 port input (falling edge). The HSD3S[3:0] bits (D[7:4]) / HSDMA Ch. 2/3 trigger set-up register (0x40299) must be set to "0001".
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 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 HS3_TF bit (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.

9. Enabling/Disabling DMA transfer

The HSDMA transfer is enabled by writing "1" to the enable bit HS3_EN (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 HS3_EN = "1". When HS3_EN is set to "0", HSDMA requests are no longer accepted. When a DMA transfer is completed (transfer counter = 0), HS3_EN is reset to "0" to disable the following trigger inputs.

Functional Description

This section describes the functionality of the USB function controller.
In the subsequent sections, the register names follow the notational convention below:

* When a register for one address is referred to:
  Register name + register.
  Example: "MainInt register"

* When a discrete bit is referred to:
  Register name. bit name + bit, or bit name + bit.
  Example: "MainIntStat.RcvEP0SETUP bit", or "ForceNAK bit of the EP0ControlOUT register"

* When a register present for a specific end-point is referred to:
  EPx{x=0,a,b,c,d}register name + register, EPx{x=a,b,c,d}register name + register, and so forth.
  Example: "EPx{x=0,a,b,c,d}IntStat register", "EPx{x=a,b,c,d}Control register"

USB Control

End points

This macro has an endpoint (EP0) for control transfer and four general purpose-endpoints (EPa, EPb, EPc, EPd). Endpoints, EPa, EPb, EPc and EPd can be used as endpoints for bulk- or interrupt- or isochronous-type transfer, respectively. There is no difference between bulk and interrupt transfers in terns of hardware.
The macro hardware provides endpoints and manages transactions. However, it does not provide a management function in the interface defined for the USB (hereinafter referred to as USB-defined interface). The USB-defined interface should be implemented in your firmware. According to the device-specific descriptor definition, set endpoints as required and configure the USB-defined interface using an appropriate endpoint combination.
Besides variable control items and statuses that are controlled for each transfer operation, each endpoint has fixed basic setting items determined by the USB-defined interface. The basic setting items should be set up when initializing the chip or when the USB-defined interface is switched in response to a SetInterface() request. Table VI.2.3 lists the basic setting items for the EP0 endpoint (default control pipe).
The EP0 endpoint shares the register set and FIFO region between the In and OUT directions. For data and status stages at the EP0 endpoint, set the data transaction direction in your firmware before executing such stages.

Table VI.2.3 Basic Setting Items for Endpoint EP0

Item	Register/bit	Description
Max. packet size	EP0MaxSize	Sets the maximum packet size to 8, 16, 32 or 64 for the FS-mode operation. The EP0 endpoint is assigned a region of the size that is set in the EP0MaxSize register, starting with FIFO address 0.

Table VI.2.4 lists the basic setting items for the general-purpose endpoints (EPa, EPb, EPc, and EPd). The EPa, EPb, EPc, and EPd endpoints allow optional settings for the transaction directions and the endpoint numbers, which allows up to three discrete endpoints to be used. Set up and/or enable these endpoints as appropriate according to the definitions for the USB-defined interface.

Table VI.2.4 Setting Items for Endpoints EPa, EPb, EPc and EPd

Item	Register/bit	Description
Transaction direction	EPx{x=a,b,c,d}Config.INxOUT	Sets the transfer direction for each endpoint.
Max. packet size	EPx{x=a,b,c,d}MaxSize_H, EPx{x=a,b,c,d}MaxSize_L	Sets the maximum packet size of each endpoint to any desired value between 1 and 1024 bytes. For endpoints that perform bulk transfers, set them to 8, 16, 32 or 64 bytes in FS mode.
Endpoint number	EPx{x=a,b,c,d}Config.EndPointNumber	Sets each endpoint number to any desired value between 0x1 and 0xF.
Toggle mode	EPx{x=a,b,c,d}Config.ToggleMode	Sets a mode for a toggle sequence. Set it to "0" for an endpoint that performs bulk transfer. 0: Toggles only in successful transactions. 1: Toggles for every transaction.
Enable endpoint	EPx{x=a,b,c,d}Config.EnEndPoint	Enables each endpoint. Set it up when the USB-defined interface that uses the relevant endpoint is enabled.
FIFO region	EPx{x=a,b,c,d}StartAdrs_H, EPx{x=a,b,c,d}StartAdrs_L	Sets a region to be assigned to each endpoint using FIFO addresses. For a FIFO region, assign a region equivalent to the maximum packet size set for the relevant endpoint or greater. Note that the size of the FIFO region affects data transfer throughput. For details of FIFO region assignment, see the "FIFO Management" section.

Transaction

This macro hardware executes transactions while its interface provides the firmware with utilities for executing transactions. The interface to the firmware is implemented through control and status registers as well as the interrupt signal which is asserted depending on the status. For settings that enable asserting interruption according to the status, see the section on register description.

The macro issues a status to the firmware for each transaction. However, the firmware does not always have to control respective transactions. The macro references the FIFO when responding to a transaction and determines if data transfer is possible based on the number of data or vacancies to automatically handle the transaction. For example, for an OUT endpoint, the firmware can smoothly and sequentially process OUT transactions by reading data from the FIFO region via either the Port interface (EPa, EPb, EPc, EPd) or the CPU interface (EP0, EPa, EPb, EPc, EPd) to create a space in the FIFO region. On the other hand, for an IN endpoint, the firmware can smoothly and sequentially process IN transactions by writing data in the FIFO region via either the Port interface (EPa, EPb, EPc, EPd) or the CPU interface (EP0, EPa, EPb, EPc, EPd) to create valid data.

Table VI.2.5 lists control items and statuses related to transaction control on the EP0 endpoint.

Table VI.2.5 Control Items and Statuses for Endpoint EP0

Item	Register/bit	Description
Transaction direction	EP0Control.INxOUT	Sets the transfer direction at the data and status stages.
Enable descriptor return	EP0Control.ReplyDescriptor	Activates automatic descriptor return.
Enable short packet transmission	EP0ControlIN.EnShortPkt	Enables transmission of short packets that are under the maximum packet size. This setting is cleared after the IN transaction that has transmitted a short packet is completed.
Toggle sequence bit	EP0ControlIN.ToggleStat, EP0ControlOUT.ToggleStat	Indicates the state of the toggle sequence bit. This setting is automatically initialized by the SETUP stage.
Set toggle	EP0ControlIN.ToggleSet, EP0ControlOUT.ToggleSet	Sets the toggle sequence bit.
Clear toggle	EP0ControlIN.ToggleClr, EP0ControlOUT.ToggleClr	Clears the toggle sequence bit.
Forced NAK response	EP0ControlIN.ForceNAK, EP0ControlOUT.ForceNAK	Returns a NAK response to IN or OUT transactions regardless of the number of data or vacancies in the FIFO region.
STALL response	EP0ControlIN.ForceSTALL, EP0ControlOUT.ForceSTALL	Returns a STALL response to IN or OUT transactions.
Set automatic ForceNAK	EP0ControlOUT.AutoForceNAK	Sets the EP0Control.ForceNAK bit whenever an OUT transaction is completed.
SETUP reception status	MainIntStat.RcvEP0SETUP	Indicates that a SETUP transaction is executed.
Transaction status	EP0IntStat.IN_TranACK, EP0IntStat.OUT_TranACK, EP0IntStat.IN_TranNAK, EP0IntStat.OUT_TranNAK, EP0IntStat.IN_TranErr, EP0IntStat.OUT_TranErr	Indicates the result of the transaction.

Table VI.2.6 lists control items and statuses related to transaction processing on the EPa, EPb, EPc, and EPd endpoints.

Table VI.2.6 Control Items and Statuses for Endpoints EPa, EPb, EPc, and EPd

Item	Register/bit	Description
Set automatic ForceNAK	EPx{x=a,b,c,d}Control.AutoForceNAK	Sets the endpoint's EPx{x=a,b,c,d}Control. ForceNAK bit whenever an OUT transaction is completed.
Enable short packet transmission	EPx{x=a,b,c,d}Control.EnShortPkt	Enables transmission of short packets that are under the maximum packet size for IN transactions. This setting is cleared after the IN transaction that has transmitted a short packet is completed.
Disable automatic ForceNAK setting upon short packet reception	EPx{x=a,b,c,d}Control. DisAF_NAK_Short	In OUT transactions, reception of a short packet automatically disables the function that sets the endpoint's EPx{x=a,b,c,d}Control.ForceNAK bit.
Toggle sequence bit	EPx{x=a,b,c,d}Control.ToggleStat	Indicates the state of the toggle sequence bit.
Set toggle	EPx{x=a,b,c,d}Control.ToggleSet	Sets the toggle sequence bit.
Clear toggle	EPx{x=a,b,c,d}Control.ToggleClr	Clears the toggle sequence bit.
Forced NAK response	EPx{x=a,b,c,d}Control.ForceNAK	Returns a NAK response to a transaction regardless of the number of data or vacancies in the FIFO region.
STALL response	EPx{x=a,b,c,d}Control.ForceSTALL	Returns a STALL response to the transaction.
Transaction status	EPx{x=a,b,c,d}IntStat.OUT_ShortACK, EPx{x=a,b,c,d}IntStat.IN_TranACK, EPx{x=a,b,c,d}IntStat.OUT_TranACK, EPx{x=a,b,c,d}IntStat.IN_TranNAK, EPx{x=a,b,c,d}IntStat.OUT_TranNAK, EPx{x=a,b,c,d}IntStat.IN_TranErr, EPx{x=a,b,c,d}IntStat.OUT_TranErr	Indicates the result of the transaction.

SETUP transaction

The SETUP transaction addressed to the EP0 endpoint of the macro's own node is automatically executed. (The USB function must be enabled for this to happen.)

When a SETUP transaction is issued, all the contents of the data packet (8 bytes) are stored in the registers EP0Setup_0 through EP0Setup_7, followed by an ACK response. Meanwhile, a RcvEP0SETUP status is issued to the firmware.

If an error occurs during a SETUP transaction, no response or status is issued.

When the SETUP transaction is completed, the ForceNAK bit of the EP0ControlIN and EP0ControlOUT registers are set and the ForceSTALL bit is cleared. The ToggleStat bit is also set. After the firmware completes setting the EP0 endpoint and becomes ready to proceed to the next stage, clear the ForceNAK bit of the relevant direction in the EP0ControlIN or EP0ControlOUT register.

Figure VI.2.2 illustrates how the SETUP transaction is executed.

(a) The host issues a SETUP token addressed to the EP0 endpoint of this node.
(b) Next, the host sends an 8-byte long data packet. The macro writes these data in the EP0Setup_0 through EP0Setup_7 registers.
(c) The macro automatically returns an ACK response. In addition, it sets registers to be automatically set up and issues a status to the firmware.

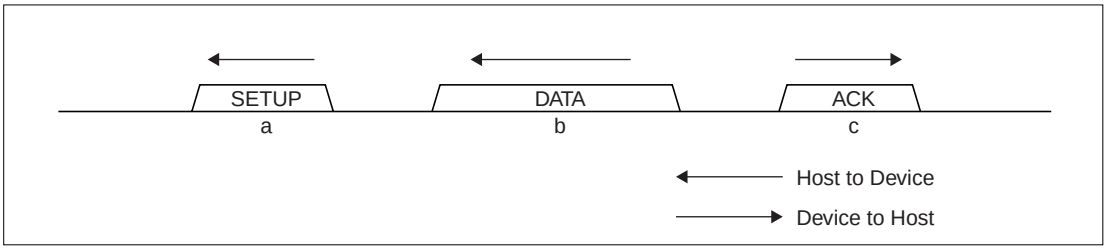


Figure VI.2.2 SETUP Transaction

OUT transaction

In OUT transactions, data reception is started regardless of the available space in the FIFO. Thus, this product provides satisfactory throughput by assigning a FIFO region about twice as large as the maximum packet size since it can read the FIFO data via the Port interface, for example, and receive data while creating an available space concurrently.

After all data are successfully received in an OUT transaction, the transaction is closed and an ACK response is returned. In addition, the firmware receives an OUT_TranACK status of the relevant endpoint (EPx{x=0,a,b,c,d}IntStat.OUT_TranACK bit). Furthermore, the FIFO is updated to acknowledge the data reception and to secure a space for the data.

In OUT transactions on the EPa, EPb, EPc, and EPd endpoints, reception of all short-packet data causes an OUT_ShortACK status (EPx{x=a,b,c,d}IntStat.OUT_ShortACK status) to be issued, in addition to executing the above closing process. If the EPx{x=a,b,c,d}Control.DisAF_NAK_Short bit is cleared, the relevant endpoint's EPx{x=a,b,c,d}ForceNAK bit is set.

If a toggle miss-match has occurred in an OUT transaction, an ACK response is returned to the transaction but no status is issued. Accordingly, the FIFO is not updated.

In the event of an error in an OUT transaction, no response is returned to the transaction. And an OUT_TranErr status (EPx{x=0,a,b,c,d}IntStat.OUT_TranErr bit) is issued. Accordingly, the FIFO is not updated.

If not all data are received in an OUT transaction, a NAK response is returned to the transaction and the OUT_TranNAK status (EPx{x=a,b,c,d}IntStat.OUT_TranNAK bit) is issued. Accordingly, the FIFO is not updated.

Figure VI.2.3 illustrates how a successful OUT transaction is executed and closed.

(a) The host issues an OUT token addressed to an OUT endpoint present on this node.
(b) Next, the host sends a data packet under the maximum packet size. The macro writes these data in the relevant endpoint's FIFO.
(c) Upon data reception, the macro automatically returns an ACK response. In addition, it sets registers to be automatically set up and issues a status to the firmware.

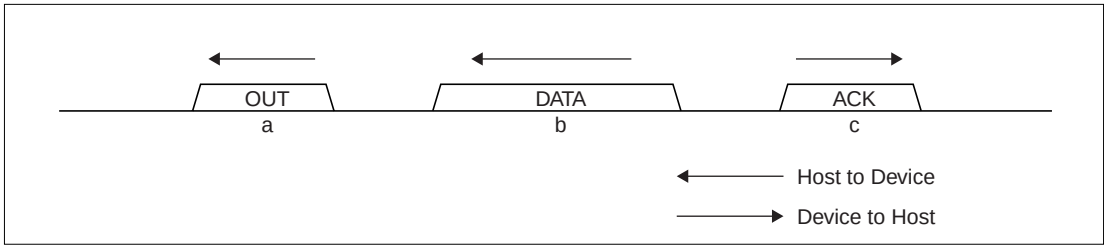


Figure VI.2.3 OUT Transaction

IN transaction

On an IN endpoint, if maximum packet size data exist in the FIFO or if the firmware has granted permission for short-packet transmission, the macro responds to the IN transaction, returning the data packet.
A permission for short-packet data transmission (including zero-length packets) is granted by setting the EP0ControlIN.EnShortPkt bit and the EPx{x=a,b,c,d}Control.EnShortPkt bit. When transmitting a short-packet data, make sure that no attempt is made to write any new data into the endpoint's FIFO after the transmission permission is granted and until the transaction is closed.
On the EP0 endpoint, the EP0ControlIN.ForceNAK bit is set after the IN transaction that transmits the short-packet data is closed.
After an ACK response is received in the IN transaction that has returned the data, the transaction is closed, followed by issuance of an IN_TranACK status (EPx{x=0,a,b,c,d}IntStat.IN_TranACK bit). Also, the FIFO is updated to acknowledge completion of the data transmission and to free the space.
If an ACK response is not received in the IN transaction that has returned the data, the transaction is considered as a failure, followed by issuance of an IN_TranErr status (EPx{x=0,a,b,c,d}IntStat.IN_TranErr bit). Accordingly, the FIFO is not updated, or no space is freed.
In on an IN endpoint, if no maximum packet size data exist in the FIFO and no permission is granted for short-packet transmission, the IN transaction receives a NAK response and an IN_TranNAK status (EPx{x=0,a,b,c,d}IntStat.IN_TranNAK bit) is issued to the firmware. Accordingly, the FIFO is not updated, or no space is freed.
Figure VI.2.4 illustrates how a successful IN transaction is executed and closed.

(a) The host issues an IN token addressed to an IN endpoint present on this node.
(b) If response is possible for this IN transaction, the macro transmits a data packet under the maximum packet size.
(c) The host returns an ACK response. After receiving an ACK response, the macro sets registers to be automatically set up and issues a status to the firmware.

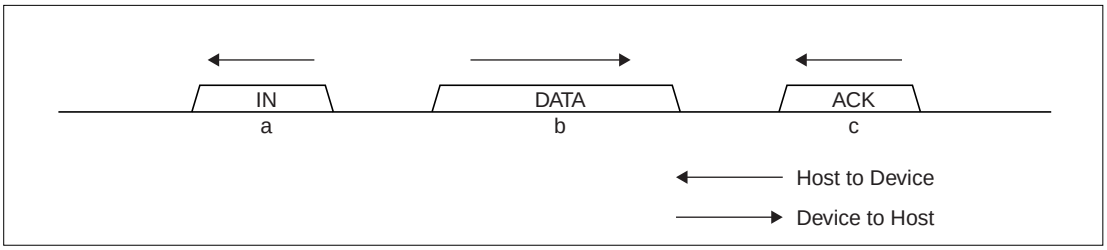


Figure VI.2.4 IN Transaction

Control transfer

Control transfer on the EP0 endpoint is controlled as a combination of a number of discrete transactions. Figure VI.2.5 illustrates how control transfer is executed for an OUT data stage.

(a) The host starts control transfer in a SETUP transaction. The device's firmware analyzes the request contents to prepare for responding to a data stage.
(b) The host issues an OUT transaction and executes a data stage, and the device receives data.
(c) The host issues an IN transaction and executes a status stage, and the device returns a zero-length data packet.

Control transfer without a data stage is executed as in this example but without the data stage.
Transition to a status stage is triggered by an issuance of a transaction by the host whose direction is opposite to that of the data stage. Have your firmware monitor an IN_TranNAK status (EP0IntStat.IN_TranNAK bit) as a trigger to transit to a status stage from a data stage.

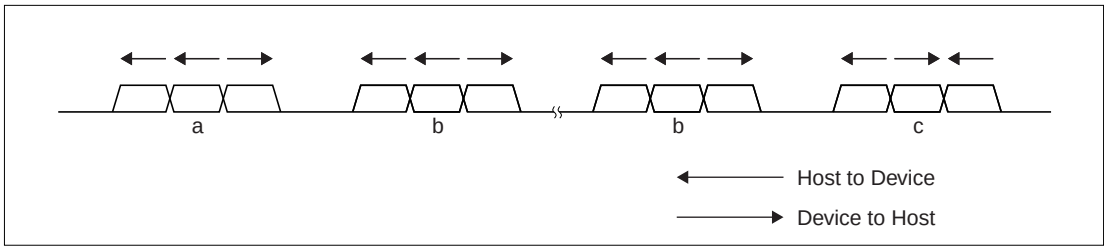


Figure VI.2.5 Control Transfer Having an OUT Data Stage

Figure VI.2.6 illustrates how control transfer is executed for an IN data stage.

(a) The host starts control transfer in a SETUP transaction. The device's firmware analyzes the request contents to prepare for responding to a data stage.
(b) The host issues an IN transaction and executes a data stage, and the device transmits data.
(c) The host issues an OUT transaction and executes a status stage, and the device returns an ACK response.

Transition to a status stage is triggered by an issuance of a transaction by the host whose direction is opposite to that of the data stage. Have your firmware monitor an OUT_TranNAK status (EP0IntStat.OUT_TranNAK bit) as a trigger to transit to a status stage from a data stage.

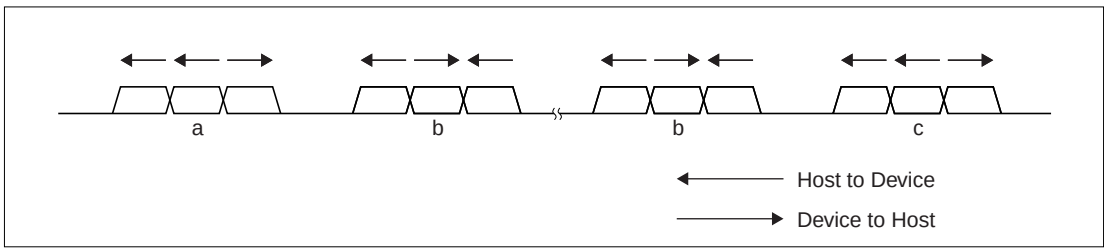


Figure VI.2.6 Control Transfer Having an IN Data Stage

Since status and data stages in control transfer execute ordinary OUT and IN transactions, flow control using NAK responses works effectively. The device is allowed to prepare for returning responses within a specified time frame.

SETUP stage

The macro automatically executes a SETUP transaction upon reception of a SETUP token addressed to its own node. Have your firmware monitor a RcvEP0SETUP status and analyze the request referring to the EP0Setup_0 through EP0Setup_7 registers to control "control transfer".
If the host has received a request that involves an OUT data stage, clear the INxOUT bit of the EP0Control register to set the EP0 endpoint direction to OUT.
If the host has received a request that involves an IN data stage, set the INxOUT bit of the EP0Control register to set the EP0 endpoint direction to IN.
If the host has received a request that involves no data stage, set the INxOUT bit of the EP0Control register to set the EP0 endpoint direction to IN in order to transit to a status stage.

Data stage/status stage

Transit to the next stage according to the result of request analysis executed by reading the EP0Setup_0 through EP0Setup_7 registers.
If it is an OUT stage, clear the INxOUT of the EP0Control register to set the direction to OUT and control the stage by setting the EP0ControlOUT accordingly. When the SETUP stage is completed, the ForceNAK bit is set.
If it is an IN stage, set the INxOUT of the EP0Control register to set the direction to IN and control the stage by setting the EP0ControlIN accordingly. When the SETUP stage is completed, the ForceNAK bit is set.

Automatic address setting function

This macro provides an automatic address setting function when processing a SetAddress() request in a control transfer at the EP0 endpoint.
This function is available for the firmware when the EP0Setup_0 through EP0Setup_7 registers are checked to confirm the contents and it is proven to be a valid SetAddress() request.
If it is determined to be a valid SetAddress() request, clear or set the EP0ControlIN.ForceNAK and EP0ControlIN.EnShortPkt bits accordingly and set the USB_Address.AutoSetAddress bit before responding to the status stage.
After this function is enabled and the IN transaction at the EP0 endpoint is completed, the macro extracts the address from the data in the SetAddress() request and sets it on the USB_Address.USB_Address bit. Meanwhile, a SetAddressCmp status (MainIntStat.SetAddressCmp bit) is issued to the firmware.
After this function is enabled, if any other transaction is invoked at the EP0 endpoint before an IN transaction is executed, this function is cancelled and the USB_Address.AutoSetAddress bit is cleared. Accordingly, a SetAddressCmp status is not issued to the firmware.

Descriptor return function

This macro provides a descriptor return function that is useful for an request that requires data and is issued more than once during control transfer at the EP0 endpoint (for example, during a GetDescriptor() request). The firmware can use this function for a request that involves an IN data stage.
Clear the EP0ControlIN.ForceNAK bit, and before starting responding to the data stage, set the top address of the data to be returned that is within the FIFO's descriptor region on the DescAdrs_H,L register as well as the total number of bytes in the return data on the DescSize_H,L register and set the EP0Control.ReplyDescriptor bit.
The descriptor return function executes IN transactions by returning data packets in response to IN transactions until it finishes sending all of a specified number of data. If a fractional number of data exist against the maximum packet size, the descriptor return function sets EP0ControlIN.EnShortPkt, enabling response to IN transactions until the entire data return is completed.
After returning all the specified number of data, the macro clears the EP0Control.ReplyDescriptor bit and issues a DescriptorCmp status (EPnIntStat.DescriptorCmp bit) to the firmware.
For details of the descriptor region, see the section on the FIFO in the functional description.

Bulk transfer/interrupt transfer

Bulk and interrupt transfers at the general-purpose endpoints, EPa, EPb, EPc, and EPd, can be controlled either as a data flow or as a series of discrete transactions (see the "Transaction" section).

Data flow control

This section describes controlling standard data flows in OUT and IN transfers.

OUT transfer

Data received from an OUT transfer are placed on the FIFO region at the respective endpoints. The FIFO data can be read via either the CPU interface (EP0, EPa, EPb, EPc, EPd) or the Port interface (EPa, EPb, EPc, EPd).
To read the FIFO data via the CPU interface, select one and only one endpoint using the CPU_JoinRd register. The FIFO data of the selected endpoint can be read sequentially with the EPnFIFOforCPU, according to the order of reception. Also, you can refer to the EPnRdRemain_H and EPnRdRemain_L registers to check the number of remaining data. Reading from an blank FIFO causes dummy reading to be performed.
To read the FIFO data via the Port interface, select one and only one OUT endpoint using the DMA_Join register. Perform the Port interface procedure to read the FIFO data of the selected endpoint; they are read sequentially in the order of reception. Also, you can refer to the DMA_Remain_H,DMA_Remain_L register to check the number of remaining data. After the FIFO is emptied, the Port interface automatically pauses to perform flow control.
Do not set the CPU and Port interfaces with the CPU_JoinRd and DMA_Join registers for reading from the same endpoint. Additionally, be sure to start reading data after ensuring that no data return responses are returned to IN transactions by setting the ForceNAK bit, for example, if you want to set an IN endpoint for data reading using the CPU_JoinRd register.
Data cannot be read from the IN endpoint via the Port interface.
If the FIFO has available space for receiving data packets, the macro automatically responds to OUT transactions to receive data. This enables the firmware to perform OUT transfer without individual transaction control. Note, however, that the EPx{x=a,b,c,d}Control.ForceNAK bit of the endpoint is set if short packets are received (including zero-length data packet) when the EPx{x=a,b,c,d}Control.DisAF_NAK_Short bit is cleared. Clear this bit when the next data transfer is ready.
Figure VI.2.7 illustrates the data flow in OUT transfer. The FIFO region for an OUT endpoint is connected to the Port interface. Also, the FIFO region assigned to this endpoint is assumed to be twice as large as the maximum packet size.

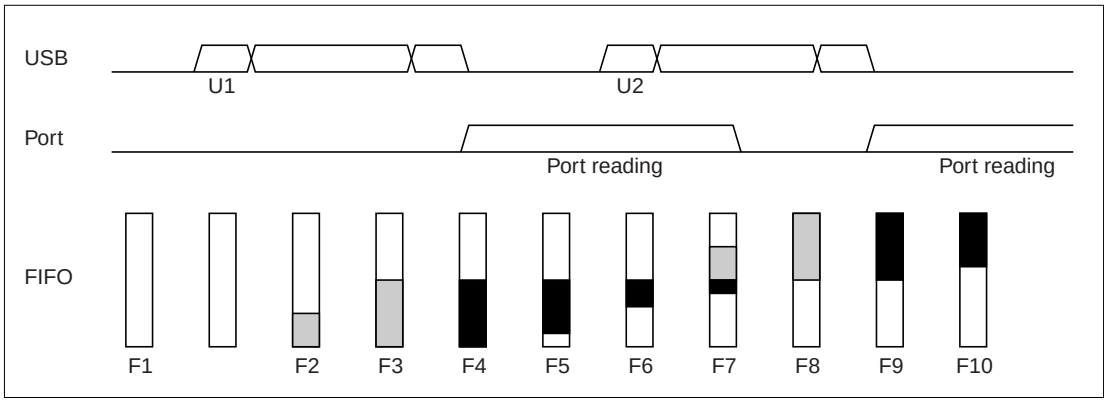


Figure VI.2.7 Example of Data Flow in OUT Transfer

(U1) Data transfer of the maximum packet size is performed in the first OUT transaction.
(U2) Data transfer of the maximum packet size is performed in the second OUT transaction.
(F1) The FIFO is blank. Although the Port interface is invoked, no transfer is performed since the FIFO is blank. (The #DMAREQ3 signal is negated.)
(F2) An OUT transaction is developing, and data reception has started in the FIFO. At this point, the FIFO data is not considered to be valid since the transaction is not closed.
(F3) Although data packet reception is completed from the OUT transaction, the FIFO data is not considered to be valid since the transaction is not closed.
(F4) The OUT transaction is closed and the received data are considered to be valid.
(F5) The presence of valid data in the FIFO triggers data transfer via the Port interface. (The #DMAREQ3 signal is asserted.)
(F6) As Port transfer develops, the amount of the remaining valid data in the FIFO is reduced.
(F7) Starting the next transaction starts writing data. Port transfer continues as long as any valid data remains.
(F8) Port transfer has stopped as there is no valid data left. The second OUT transaction is not closed yet.
(F9) The second OUT transaction is closed, causing the FIFO data to become valid.
(F10) The presence of valid data in the FIFO restarts Port transfer.

IN transfer

Place data transmitted thorough IN transfer on each endpoint's FIFO. The FIFO data can be written via either the CPU interface (EP0, EPa, EPb, EPc, EPd) or the Port interface (EPa, EPb, EPc, EPd).
To write data into the FIFO via the CPU interface, select one and only one endpoint using the CPU_JoinWr register. Data can be written in the selected endpoint's FIFO by using the EPnFIFOforCPU register, which are transmitted in data packets in the order of writing. Also, you can refer to the EPnWrRemain_H, EPnWrRemain_L register to check the available space in the FIFO. An attempt to write in a full FIFO causes dummy writing to be performed.
To write data into the FIFO via the Port interface, select one and only one IN endpoint using the DMA_Join register. Perform the Port interface procedure to write data into the selected endpoint's FIFO. These data are transmitted in data packets in the order of writing. After the FIFO becomes full, the Port interface automatically pauses to perform flow control.
Do not set the CPU and Port interfaces with the CPU_JoinWr and DMA_Join registers for writing data into the same endpoint. Additionally, be sure to start writing data after ensuring that no data are received from the OUT transactions by setting the ForceNAK bit, for example, if you want to set an OUT endpoint for data writing using the CPU_JoinWr register.
Data cannot be written into an OUT endpoint via the Port interface.
If the FIFO contains data exceeding the maximum packet size, the macro automatically responds to IN transactions to perform data transmission. This enables the firmware to perform IN transfer without individual transaction control. Note, however, that you should set the EnShortPkt bit if you need to transmit a short packet at the end of the data transfer. Since this bit is cleared when the IN transaction which has transmitted the short packet is closed, you can set it after data is completely written into the FIFO.
When the DMA_Join.AutoEnShort bit is set, the EPx{x=a,b,c,d}Control.EnShortPkt bit of the relevant endpoint is automatically set if the FIFO still contains any fractional amount of data under the maximum packet size after writing via the Port interface is completed. Using this function provides automatic control to the end that only a non-zero-length short packet is returned, eliminating return of a zero-length data packet.
Figure VI.2.8 illustrates the data flow in IN transfer. The FIFO region for an IN endpoint is connected to the Port interface. Also, the FIFO region assigned to this endpoint is assumed to be twice as large as the maximum packet size.

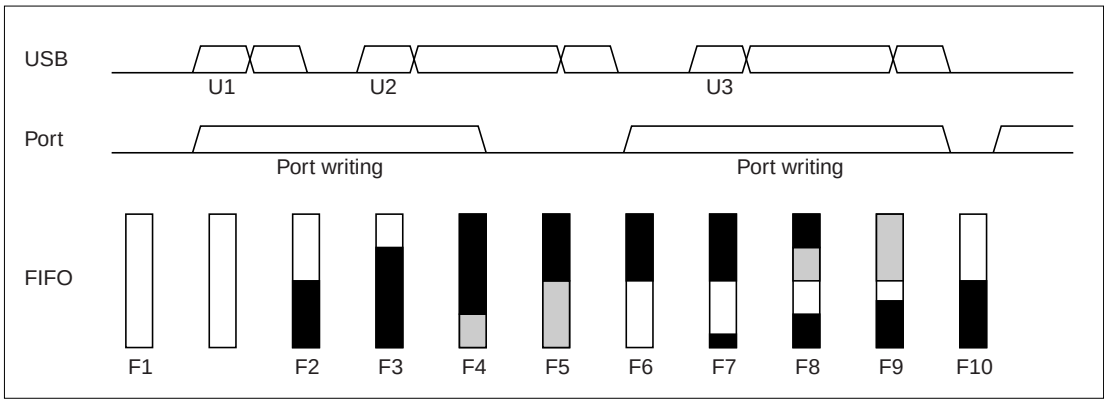


Figure VI.2.8 Example of Data Flow in IN Transfer

(U1) In the first IN transaction, an NAK response is returned since the FIFO has no valid maximum packet size data.
(U2) Data transfer of the maximum packet size is performed in the second IN transaction.
(U3) Data transfer of the maximum packet size is performed in the third IN transaction.
(F1) The FIFO is blank.
(F2) Port transfer is started and valid data is written into the FIFO. (The #DMAREQ3 signal is asserted.)
(F3) As the FIFO still has an available space, Port transfer is continuing.
(F4) Since the FIFO contains valid maximum packet size data, the macro responds to the IN transaction with data packet transmission. As the transaction is not closed yet, the region from which data are transmitted is not freed. The FIFO is full, causing Port transfer to stop. (The #DMAREQ3 signal is negated.)
(F5) Although data packet transmission in the IN transaction has been completed, the FIFO region is not freed since the transaction is not closed. Port transfer remains discontinued.
(F6) The FIFO region is freed as the transaction is closed upon reception of an ACK handshake packet.
(F7) As the FIFO now has some available space, Port transfer is resumed. (The #DMAREQ3 signal is asserted.)
(F8) The macro responds to an IN transaction and transmits a data packet. Since the FIFO has some available space, Port transfer continues.
(F9) Although data packet transmission in the IN transaction has been completed, the FIFO region is not freed since the transaction is not closed. Since the FIFO has some available space, Port transfer continues.
(F10) The FIFO region is freed when the transaction is closed upon reception of an ACK handshake packet. Although Port transfer pauses as all the available space has been consumed, it is resumed upon closing of the IN transaction that creates available space.

Auto-negotiation function

This function automatically performs Suspend detection, Reset detection, Resume detection and Restore execution, checking the state of the USB bus for each operation. You can check each interruption (DetectRESET, DetectSUSPEND, and RestoreCmp) to confirm what has been actually performed.

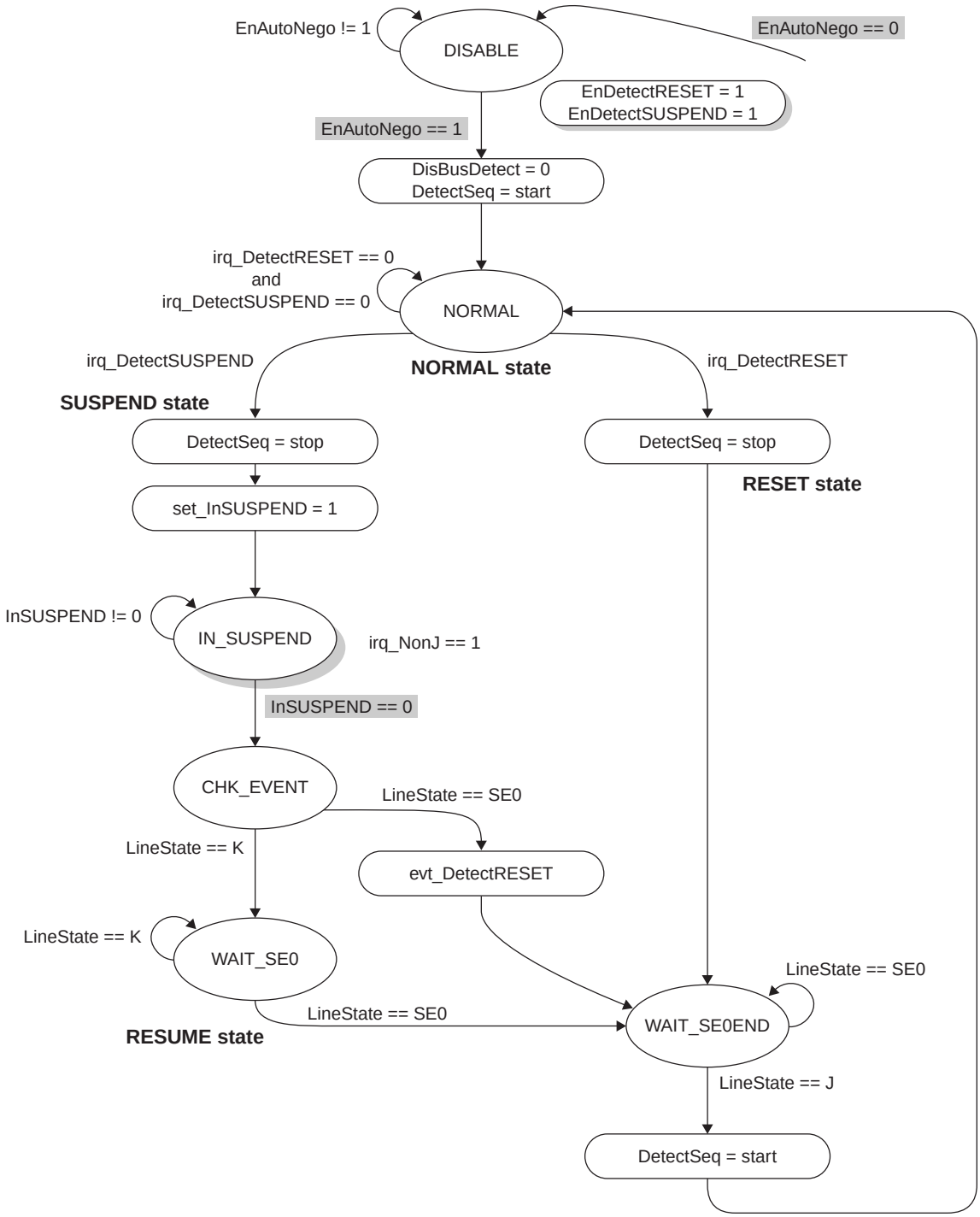


Figure VI.2.9 Auto-negotiator

(1) DISABLE

The macro enters this state when the USB_Control.EnAutoNego bit is cleared.
To enable the auto-negotiation function, set interruptions for Reset detection (SIE_IntEnb.EnDetectRESET) and Suspend detection (SIE_IntEnb.EnDetectSUSPEND) before setting the USB_Control.EnAutoNego bit and give permission to event detection interruption.
Enabling the auto-negotiation function automatically clears the USB_Control.DisBusDetect bit and enables the event detection function. While the auto-negotiation function is enabled, never set the USB_Control.DisBusDetect bit.

(2) NORMAL

This is a state of waiting for Reset or Suspend detection.
The state is determined to be Reset if SE0 of 2.5 µs or greater, and it is determined to be Suspend if no activities are detected beyond 3 ms. Concurrently with judgment as described above, an interruption for Reset detection or Suspend detection is generated, and the SIE_IntStat.DetectRESET bit and the SIE_IntStat.DetectSUSPEND bit are set.
If the state is determined to be Suspend, suspend the event detection function once and enter the IN_SUSPEND state.

(3) IN_SUSPEND

When the state is determined to be SUSPEND, H/W automatically sets the USB_Control.InSUSPEND bit and the macro enters the IN_SUSPEND state. This USB_Control.InSUSPEND bit enables the function of detecting changes of buses from FS-J, only enabling detection of Resume or Reset from the host.
The ability to reduce current consumption during Suspend depends on the application. This macro provides SNOOZE function for reducing current consumption. To use the function of reducing current consumption when the auto-negotiation function is enabled, be sure to check that the USB_Control.InSUSPEND bit is set before staring the current consumption reducing function.
At this time, in order to detect Resume (FS-K) that indicates the end of Suspend, set the SIE_IntEnb.EnNonJ bit in the firmware when the macro enters this state to give permission to NonJ interruption.
When NonJ interruption status (SIE_IntStat.NonJ) is set, it is interpreted as an indication of return from Suspend, and the macro enters the CHK_EVENT state after the USB_Control.InSUSPEND bit is cleared in the firmware.
In an application with a remote wake-up function enabled, if it is determined that the macro must return from Suspend, set the USB_Control.SendWakeup bit in this state and output FS-K at least for 1 ms but do not exceed 15 ms.

(4) CHK_EVENT

In this state, the macro checks the USB cable and determines that the state is Resume if FS-K is detected, and that it is Reset if SE0 is detected. When determined to be Reset, set the SIE_IntStat.DetectRESET bit.

Note that you should terminate this auto-negotiation function as soon as the USB cable is unplugged; in none of the above states, the macro does not consider the implication of USB cable disconnection.

Description by negotiation function

Suspend detection

When the USB_Control.DisBusDetect bit is set to "0", the macro hardware automatically performs the following Suspend detection sequence.

(1) The internal timer checks that there is no data transmission/reception (continues to detect "J" in USB_Status.LineState[1:0]) for 3 ms or longer (T_1).
(2) At T_2, if "J" is detected in USB_Status.LineState[1:0], set the SIE_IntStat.DetectSUSPEND bit.
(3) If the SIE_IntEnb.EnDetectSUSPEND and MainIntEnb.EnSIE_IntStat bits are set, the macro asserts the #INT signal.

If the SIE_IntStat.DetectSUSPEND bit is set, on the firmware that controls this macro, set the USB_Control.DisBusDetect bit to "1" and the USBSNZ bit (D5/0x300F20) to "1" to start processing Snooze before reaching T_4. As for self-powered products, however, the firmware does not have to perform Snooze. (Figure VI.2.10 shows the operation when Snooze is performed.)

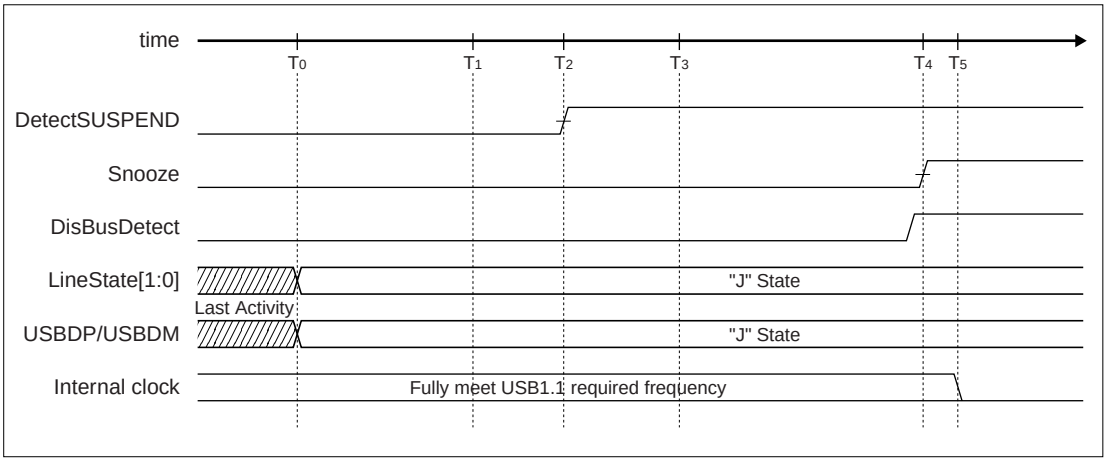


Figure VI.2.10 Suspend Timing (FS mode)

Reset detection

When the USB_Control.DisBusDetect bit is set to "0", the macro hardware automatically performs the following Reset detection sequence.

(1) The internal timer checks that it has continued to detect "SE0" in USB_Status.LineState[1:0]) for 2.5 µs or longer (T_1).
(2) At T_2, if "SE0" is detected in USB_Status.LineState[1:0], the macro sets the SIE_IntStat.DetectRESET bit.
(3) If the SIE_IntEnb.EnDetectRESET and MainIntEnb.EnSIE_IntStat bits are set, the macro asserts the #INT signal.

If the SIE_IntStat.DetectRESET bit is set, on the firmware that controls this macro, set the USB_Control.DisBusDetect bit to "1".

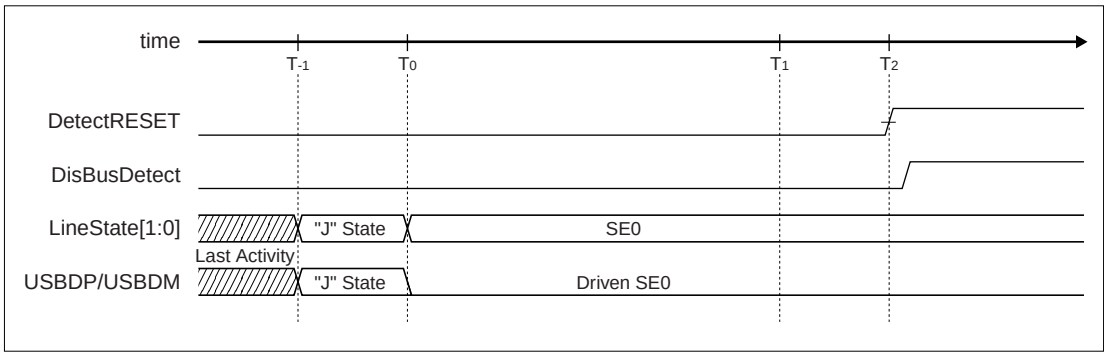


Figure VI.2.11 Reset Timing (FS mode)

Issuing resume

This section describes how to enable automatic resume to be triggered by some cause when remote wake-up is supported and the remote wakeup function is enabled from the host.
Remote wakeup can only be enabled 5 ms after the bus enters the Idle state. Furthermore, the current used before the USB Suspend state cannot be pulled from the VBUS until 10 ms has elapsed after the Resume signal output.

The S1C33L05 supports Snooze state. This section describes the operation for issuing Resume when the oscillation circuit is in operation (USBCLKEN (D4/0x300F20) = "1", not in Sleep). Steps (3), (4), (8) and (9) below are handled by the macro hardware automatically. Perform steps (1), (2) and (6) on the firmware that controls this macro.

(1) Clear the SIE_IntEnbEnNonJ and USBSNZ (D5/0x300F20) bits. This is to cause this macro return from Snooze for automatic wakeup.
(2) Set the USB_Control.SendWakeup bit and send out the Resume signal.
(3) The macro sets XcvrControl.OpMode[1:0] to "Disable Bit Stuffing and NRZI encoding" and prepares for transmission of "All 0" data.
(4) The macro starts data transmission and sends out "FS K" (the Resume signal) to a downstream port.
(5) The downstream port detects this Resume signal and returns "FS K" (the Resume signal) onto the bus.
(6) Clear the USB_Control.SendWakeup bit and suspend Resume signal send-out. After that, set the USB_Control.RestoreUSB bit.
(7) The downstream port suspends Resume signal send-out. Here, note that the Resume signal from downstream port (host) has EOP of LS at the end.
(8) The macro clears the USB_Control.RestoreUSB bit and sets the SIE_IntStat.RestoreCmp bit.
(9) If the SIE_IntEnb.EnRestoreCmp and MainIntEnb.EnSIE_IntStat bits are set, the macro asserts the #INT signal.

This section describes the operation of the oscillation circuit by assuming that it is in operation (USBCLKEN (D4/0x300F20) = "1", not in Sleep). If the oscillation circuit is in the Sleep state (deactivated), OSC power-up time is needed before returning from the Snooze state (with the USBSNZ bit (D5/0x300F20) reset from "1" to "0").

Detecting resume

When the USB is suspended, "J" is observed on the bus (USB_Status.LineState[1:0] is "J"). If "K" is observed on the bus, it means the instruction for wakeup (Resume) is received from the downstream port. This section describes the operation when Resume is detected, assuming that this macro is in the Snooze state when the USB is suspended. Use the firmware that controls this macro to perform steps (4) and (5). The other steps are handled by the macro hardware automatically.

(1) The bus transits from "J" to "K".
(2) The macro sets the SIE_IntStat.NonJ bit.
(3) If the SIE_IntEnb.EnNonJ and MainIntEnb.EnSIE_IntStat bits are set, the macro asserts the #INT signal.
(4) Clear the USBSNZ bit (D5/0x300F20).
(5) Set USB_Control.RestoreUSB.
(6) The downstream port suspends "K" send-out.
(7) The macro clears the USB_Control.RestoreUSB bit and sets the SIE_IntStat.RestoreCmp bit.
(8) If the SIE_IntEnb.EnRestoreCmp and MainIntEnb.EnSIE_IntStat bits are set, the macro asserts the #INT signal.

This section describes the operation of the oscillation circuit by assuming that it is in operation (USBCLKEN (D4/0x300F20) = "1", not in Sleep). If the oscillation circuit is in the Sleep state (deactivated), OSC power-up time is needed before returning from the Snooze state (with the USBSNZ bit (D5/0x300F20) reset from "1" to "0").

Cable plug-in

This section describes the operation that is carried out when the macro is connected to the hub or the host (via cable plug-in). Use the firmware that controls this macro to perform steps (3) and (4). Steps (1) and (2) are handled by the macro hardware automatically.

(1) When the cable is connected, VBUS turns to HIGH and the macro sets the USB_Status.VBUS and SIE_IntStat.VBUS_Changed bit (T_0).
(2) If the SIE_IntEnb.EnVBUS_Changed and MainIntEnb.EnSIE_IntStat bits are set, the macro asserts the #INT signal.
(3) Set the USBCLKEN bit (D4/0x300F20) to start supplying the USB clock (T_1).
(4) Clear the USBSNZ bit (D5/0x300F20) (T_2).
(5) The downstream port sends out Reset (T_4).

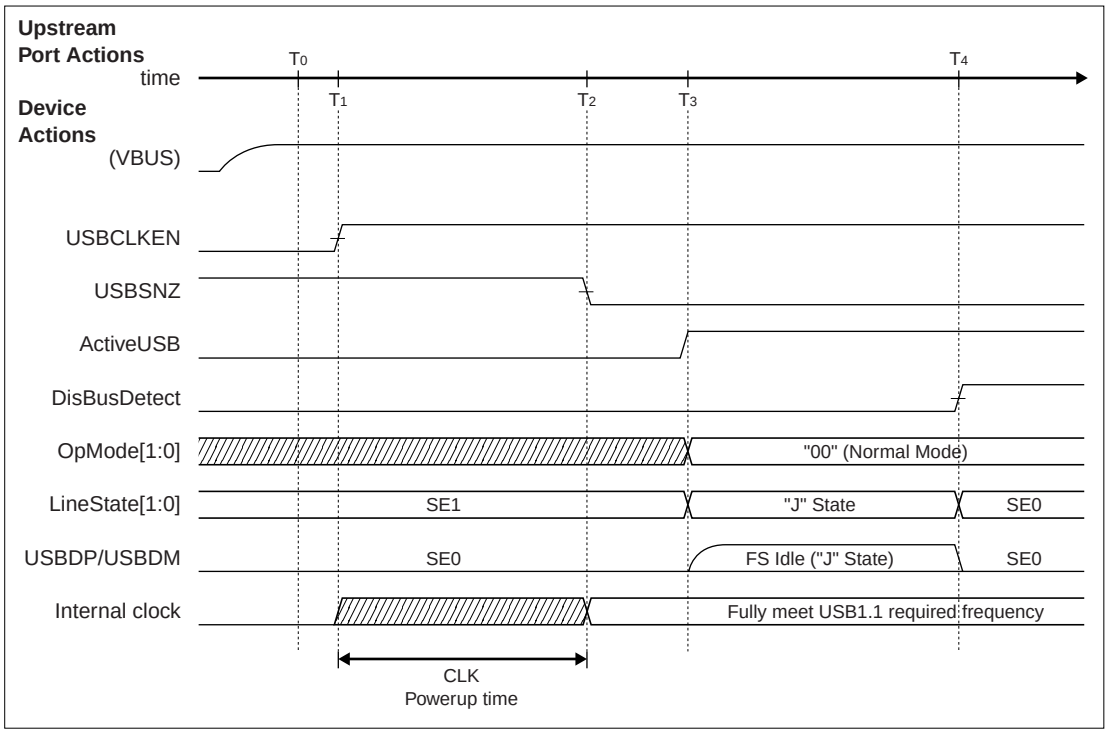


Figure VI.2.12 Device Attach Timing

Table VI.2.7 Device Attach Timing Values

Timing parameter	Description	Value
T_0	VBUS is enabled.	0 (Reference)
T_1	Set USBCLKEN to "1" (on the firmware). The clock input starts.	T_1
T_2	Clear USBSNZ to "0" (on the firmware).	T_1 + 250 ms < T_2
T_3	Set ActiveUSB to "1". Set OpMode[1:0] to "00" (on the firmware).	T_0 + 100 ms < T_3
T_4	The downstream port sends out Reset. Set DisBusDetect to "1" (on the firmware).	T_3 + 100 ms < T_4

FIFO Management

FIFO memory map

This section describes the memory map for the FIFO region.

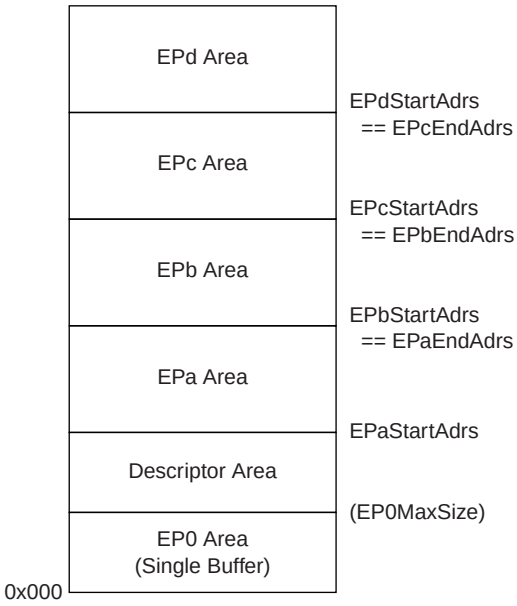


Figure VI.2.13 FIFO Memory Map

The FIFO memory is roughly divided into six areas: EP0 area, descriptor area, EPa area, EPb area, EPc area, and EPd area, and each of these areas can be divided according to the settings for the EP0MaxSize register, EPaStartAdrs register, EPbStartAdrs register, EPcStartAdrs register, and EPdStartAdrs register.

The EP0 area is used for the required USB endpoint 0, and can be used both for IN and OUT directions. This area is uniquely determined to be the maximum packet size of endpoint 0 that is set up in the EP0MaxSize register. This means that it can only receive/transmit one packet (Single Buffer) at a time.

EPa, EPb, EPc, and EPd areas are for the general-purpose endpoint that can take an endpoint number and an IN/OUT setting. The EPa area extends from the address set in the EPaStartAdrs register up to the point before the address set in the EPbStartAdrs register. The EPb area extends from the address set in the EPbStartAdrs register up to the point before the address set in the EPcStartAdrs register. The EPd area extends from the address set in the EPdStartAdrs register up to the end of FIFO RAM. The addresses available in the area setup registers must be written in the unit of four bytes (meaning that the lowest two bits cannot be written). Additionally, a space exceeding the maximum packet size must be assigned to these areas. Although there should be no problem as far a value larger than the maximum packet size is assigned, we recommend that you use its integral multiple to set them up.

The descriptor area extends from the address set in the EP0MaxSize register up to the point before the address set in the EPaMaxAdrs. (Actually, the entire FIFO region can be used as the descriptor area. We recommend, however, that the area described here be used in order to avoid operational contentions.)The practical use is described later.

Set the EPnControl.AllFIFO_Clr bit for the initial setting or re-setting of an area set-up register. Once the initial setting for an area is established, the EPnControl.AllFIFO_Clr bit is cleared. This bit will never cause the RAM data to be cleared. Therefore, unless you have changed the descriptor area, there is no need to re-set the information recorded within the area since will never be cleared otherwise.

Using the descriptor area

The descriptor area provides high-speed, straightforward execution of part of operations for packets received/transmitted via EP0, or a standard request. Among contents of standard requests, write those in this area that are uniquely determined by the device during the initial setup stage following power-on to automatically execute the data stage included in the request simply by setting the top address and the data size in response to a request from the host. Accordingly, this technique eliminates the need of writing data in the EP0 area, enabling very quick response to a request.

Writing data in the descriptor area

To write data in the descriptor area, first set the write start address in the DescAdrs_H and DescAdrs_L registers, and then write data in the DescDoor register (RegWindowSel == 0x2). After completing writing data, the DescAdrs_H and DescAdrs_L registers are automatically incremented by one, enabling sequential writing in the DescDoor register (RegWindowSel == 0x2) when writing data at a series of adjacent addresses. Note that this incrementing function does not mean that written data can be read when writing and reading are executed sequentially; it only increments by one for both writing and reading.

Reading data from descriptor area

To read data from the descriptor area, first set the read start address in the DescAdrs_H and DescAdrs_L registers, and then read data from the DescDoor register (RegWindowSel == 0x2). After completing reading data, the DescAdrs_H and DescAdrs_L registers are automatically incremented by one, enabling sequential reading in the DescDoor register (RegWindowSel == 0x2) when reading data from a series of adjacent addresses. Note that this incrementing function does not mean that written data can be read when writing and reading are executed sequentially; it only increments by one for both writing and reading.

Executing data stage (IN) in the descriptor area

To use written data in response to a request from EP0, set the top address of the data to be transmitted to the data stage, set the data size specified in the request in the DescSize_H and DescSize_L registeres, and then set the EP0Control.ReplyDescriptor bit to "1".

After receiving the IN token from the host, the macro start transmitting data to the host, automatically splitting them into the maximum packet size (set in the EP0MaxSize). In addition, if the value in the DescSize_H or DescSize_L register is under the maximum packet size, or if the remaining number of data after splitting, the macro automatically transmits such data as short packets. When the specified number of data are completely transmitted, the EP0Control.ReplyDescriptor is cleared and the EPnIntStat.DescriptorCmp is set. At this stage, the EPnIntEnb.EnDescriptorCmp bit is set and the MainIntEnb.EnEPnIntStat bit is set as well, the #INT signal is asserted at the same time.

If the process enters a status stage before the transmitted amount reaches the specified number of data (that is, if an OUT token is received), the EP0Control.ReplyDescriptor is automatically cleared to suspend this function. At the same time, the EP0IntStat.OUT_TranNAK status and the EPnIntStat.DescriptorCmp status are set. If either of the following sets of bits are set, the #INT signal is asserted at the same time:
(1) The EP0IntEnb.EnOUT_TranNAK, PnIntEnb.EnEP0IntStat and MainIntEnb.EnEPnIntStat bits, or
(2) The EPnIntEnb.EnDescriptorCmp and MainIntEnb.EnEPnIntStat bits.

Accessing to FIFO by CPU

To enable the CPU to access the FIFO, set the bit of the relevant endpoint of the CPU_JoinRd and CPU_JoinWr registsters to "1" and execute reading and writing via the EPnFIFOforCPU register. For each of the CPU_JoinRd and CPU_JoinWr registers, you can only set one bit out of the four bits. If you attempt to set more than one bit at a time, only the highest bit is set.
The EPnRdRemain_H and EPnRdRemain_L registers indicate the remaining number of data that can be read at the endpoint set in the CPU_JoinRd register. The EPnWrRemain_H and EPnWrRemain_L registers indicate the remaining area space available for writing at the endpoint set in the CPU_JoinWr register.

Note that, if the CPU_JoinRd register is set when register dumping is planned for debugging of a CPU using ICE, data will be read from the FIFO upon dumping the register.

Limiting access to FIFO

The FIFO of this macro allows concurrent execution of data reception/transmission between the macro and the USB and/or the Port and writing/reading to and from the CPU. Because of this, there are two limitations for accessing the FIFO (for writing and reading) from the CPU (the firmware):

(1) From the CPU, no writing is allowed to the same endpoint while the USB or the Port is writing data to the FIFO.
(2) No reading from the CPU is allowed from the same endpoint while the USB or the Port is reading from the FIFO.

Never execute these operations; they may destroy data continuity.

Port Interface

Functional description

The Port interface is a DMA interface designed for fast data transfer between this macro and the FIFO for its built-in endpoints. It provides Asynchronous DMA Transfer mode for transfer triggered by the Red/Write-strobe signal.

Basic operations

This section describes the basic operations of the Port interface.

Register setting

Table VI.2.8 lists the registers used for setting basic items of the Port interface. Set desired values for the respective registers. To enable the DMA to write, set the DMA_Join register to connect the Port interface to the endpoint set to the IN direction of the USB. To enable the DMA to read, connect to the endpoint set to the OUT direction.

Do not modify the basic setting registers while the DMA is transferring data (when DMA_Control.DMA_Running is set to "1"). We do not guarantee normal operations if the basic setting registers are modified while the DMA in transferring data.

Table VI.2.8 Port Interface's Registers for Basic Setting Items

Item	Register/bit	Description
Endpoint connection	DMA_Join.JoinEPr{r=a,b,c,d}DMA	Connects the Port interface to the endpoint of the bit set to "1". Writing/reading is enabled to/from the connected endpoint.
Counter setting	DMA_Count_r{r=HH,HL,LH,LL}	Sets the number of bytes to be down-counted in Countdown mode.
Active port	DMA_Config_0.ActivePort	Enables the port for the Port interface.
Active level	DMA_Config_0.PDREQ_Level DMA_Config_0.PDACK_Level DMA_Config_0.PRDWR_Level	Sets the active level of the Port interface signal. 0: High-active. 1: Low-active.
RcvLimit mode	DMA_Config_1.RcvLimitMode	Only enabled while writing in Asynchronous transfer mode. If this bit is set to "1", up to 16 bytes of data can be received even after negating #DMAREQ3.
Single-/multi-word	DMA_Config_1.SingleWord	Sets the transfer mode for operation in Asynchronous transfer mode. 0: Multi-word transfer. 1: Single-word transfer.
Count mode	DMA_Config_1.CountMode	Sets Countdown/Free-run mode. 0: Free-run mode. 1: Countdown mode.

DMA transfer

After setting the basic setting registers, write "1" to the DMA_Control.DMA_Go bit to cause the Port interface to start running the DMA. After the DMA starts running, the DMA_Control.DMA_Running bit is set to "1", indicating that the DMA is running.

If the DMA is set to the Countdown mode with DMA_Config_1.CountMode=1, the DMA completes data transfer when the DMA_Count_HH, HL, LH and LL registers reach 0000_0000h. To force the DMA to terminate data transfer, provide "1" to the DMA_Control.DMA_Stop bit. After the DMA completes data transfer, the DMA_Control.DMA_Running bit attains "0" and the MainIntStat.DMA_Cmp bit "1". At this time, if the MainIntEnb.EnDMA_Cmp bit is set, the #INT signal is asserted to the CPU.

Asynchronous DMA transfer

This macro provides an 8-bit asynchronous DMA transfer function that outputs/inputs data, triggered by the Data Transfer Request signal PDREQ (#DMAREQ3), Data Transfer Permit signal PDACK (#CE4) and Read-strobe PDRD (#RD)/Write-strobe PDWR (#WRL). This mode only supports the slave functionality, and enables data transfer either in Multi-word or Single-word mode.

Asynchronous multi-word DMA transfer mode - slave

1) Writing operation

The Port interface starts writing operation in Asynchronous multi-word DMA transfer mode when the following register settings are established:

• DMA_Config_1.SingleWord bit = "0"
• Direction of the target endpoint = IN

The Port interface starts data transfer on the DMA when "1" is written on the DMA_Control.DMA_Go bit. After data transfer starts on the DMA, it requests data transfer by asserting PDREQ (#DMAREQ3) to the C33 Core block (master) if any available space is found at the connected endpoint. The DMA loads the data and writes them to the endpoint when PDWR (#WRL) is rising (when the DMA_Config_0.PRDWR_Level bit is set to "1"). When available space is entirely consumed at the endpoint, the interface negates PDREQ (#DMAREQ3) to the C33 Core block (master) to reject data transfer.

If any data is set to the DMA_Latency.DMA_Latency[3:0] bit other than "0h", this mode negates PDREQ (#DMAREQ3) once after completing transfer of 4-byte data, and does not assert PDREQ (#DMAREQ3) as long as 130 ns × N (N = DMA_ Latency.DMA_Latency[3:0]).
If the DMA is set to the Countdown mode with DMA_Config_1.CountMode = "1", the DMA completes data transfer when the DMA_Count_HH, HL, LH and LL registers reach 0000_0000h. To force the DMA to terminate data transfer, set the DMA_Control.DMA_Stop bit to "1". Note that forced termination of DMA transfer by writing to this bit may cause loss of data from those being transferred. To avoid it, first terminate the C33 Core block (master) and then terminate the macro's DMA transfer.

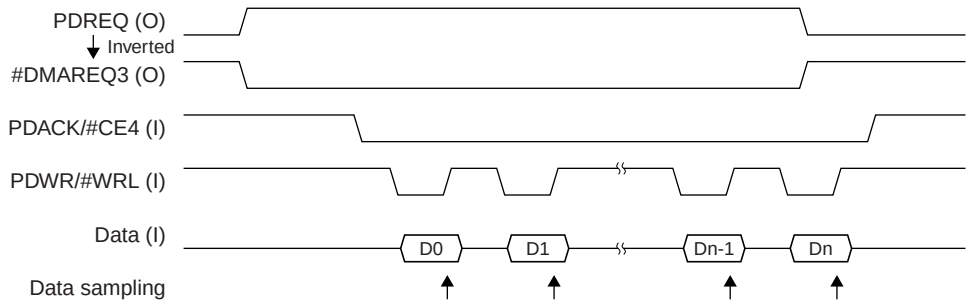


Figure VI.2.14 Transfer Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Writing

Providing "1" to the DMA_Config_1.DMA_RcvLimit bit enables the RcvLimit mode. The RcvLimit mode is not available in Countdown mode.
When the DAM is writing asynchronously in RcvLimit mode, up to 16 bytes of data can be received even after this macro negates PDREQ (#DMAREQ3).
In this mode, PDREQ (#DMAREQ3) is negated when the available space is less than 32 bytes at the relevant endpoint as a result of the DMA's writing operation. However, when PDREQ (#DMAREQ3) is negated, 16 bytes of data that have not been written to the endpoint may exist within the internal circuit. Therefore, up to 16 bytes of data can be received after PDREQ (#DMAREQ3) is negated.
In this mode, PDREQ (#DMAREQ3) is negated before the endpoint becomes completely full. If the region set with the EP{a,b,c,d}StartAddress register is the same as that set with the EP{a,b,c,d}MaxSize register (Single Buffer), the endpoint never becomes full, and data cannot be transmitted through USB IN transfer.

Therefore, you should set up an area exceeding the EP{a,b,c,d}MaxSize value + 32 bytes to use the RcvLimit mode, using the EP{a,b,c,d}StartAddress register. Therefore, you should set up an area exceeding the EP{a,b,c,d}MaxSize value + 32 bytes to use the RcvLimit mode, using the EP{a,b,c,d}StartAddress register.

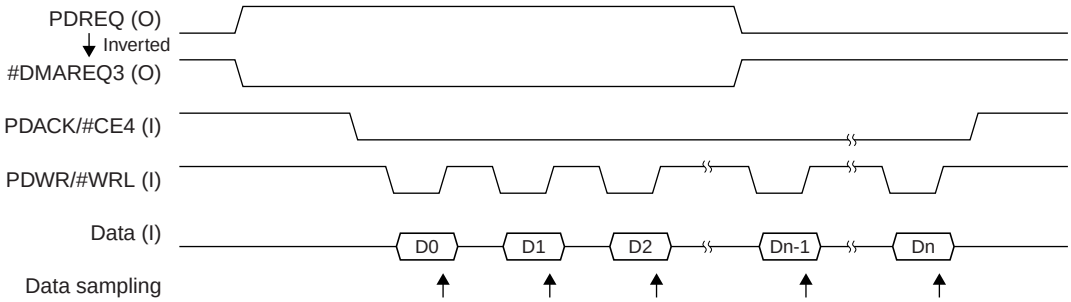


Figure VI.2.15 Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Writing (RcvLimit mode)

2) Reading operation

The Port interface starts reading operation in the Asynchronous Multi-word DMA transfer mode when the following register settings are established:

- DMA_Config_1.SingleWord bit = "0"
- Direction of the target endpoint = OUT

The Port interface starts data transfer on the DMA when "1" is written on the DMA_Control.DMA_Go bit. After data transfer starts on the DMA, it requests data transfer by asserting PDREQ (#DMAREQ3) to the C33 Core block (master) if any data exist at the connected endpoint. Turning PDACK (#CE4) to active starts outputting transferred data to the data bus. Have the C33 Core block (master) load the data while PDRD (#RD) is rising (when the DMA_Config_0.PRDWR_Level bit is set to "1"). When no data remains at the endpoint, the interface negates PDREQ (#DMAREQ3) to the C33 Core block (master) to reject data transfer.
If any data is set to the DMA_Latency.DMA_Latency[3:0] bit other than "0h", this mode negates PDREQ (#DMAREQ3) once after completing transfer of 4-byte data, and does not assert PDREQ (#DMAREQ3) as long as 130 ns × N (N= DMA_ Latency.DMA_Latency[3:0]).
If the DMA is set to the Countdown mode with DMA_Config_1.CountMode="1", the DMA completes data transfer when the DMA_Count_HH, HL, LH and LL registers reach 0000_0000h. To force the DMA to terminate data transfer, set the DMA_Control.DMA_Stop bit to "1". Note that forced termination of DMA transfer by writing to this bit may cause loss of data from those being transferred. To avoid it, first terminate the C33 Core block (the master) and then terminate the macro's DMA transfer.

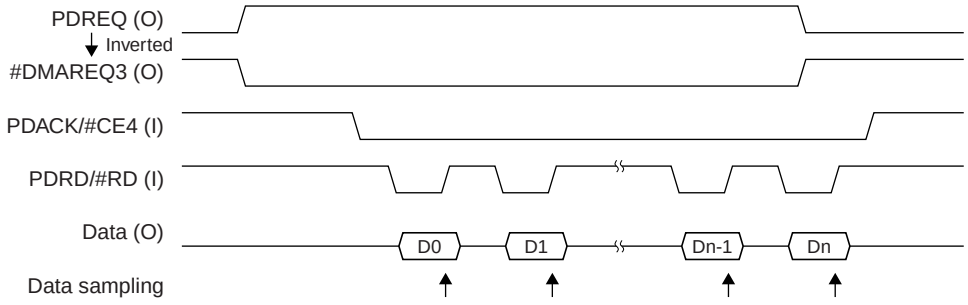


Figure VI.2.16 Transfer Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Reading

Asynchronous single-word DMA transfer mode - slave

1) Writing operation

The Port interface starts writing operation in Asynchronous single-word DMA transfer mode when the following register settings are established:

- DMA_Config_1.SingleWord bit = "1"
- Direction of the target endpoint = IN

The Port interface starts data transfer on the DMA when "1" is written on the DMA_Control.DMA_Go bit. After data transfer starts on the DMA, it requests data transfer by asserting PDREQ (#DMAREQ3) to the C33 Core block (master) if any available space is found at the connected endpoint. The DMA loads the data and writes them to the endpoint when PDWR (#WRL) is rising (when the DMA_Config_0.PRDWR_Level bit is set to "1"). This mode negates PDREQ (#DMAREQ3) after transferring 1-byte data (PDWR (#WRL) becomes active). At this point, if any space is still available at the endpoint, it requests data transfer by asserting PDREQ (#DMAREQ3) to the C33 Core block (the master). If there is no available space left at the endpoint, PDREQ (#DMAREQ3) is not asserted and data transfer is rejected.
If any data is set to the DMA_Latency.DMA_Latency[3:0] bit other than "0h", this mode negates PDREQ (#DMAREQ3) once after completing transfer of 4-byte data, and does not assert PDREQ (#DMAREQ3) as long as 130 ns × N (N=DMA_Latency.DMA_Latency[3:0]).
If the DMA is set to the Countdown mode with DMA_Config_1.CountMode="1", the DMA completes data transfer when the DMA_Count_HH, HL, LH and LL registers reach 0000_0000h. To force the DMA to terminate data transfer, set the DMA_Control.DMA_Stop bit to "1". Note that forced termination of DMA transfer by writing to this bit may cause loss of data from those being transferred. To avoid it, first terminate the C33 Core block (master) and then terminate the macro's DMA transfer.

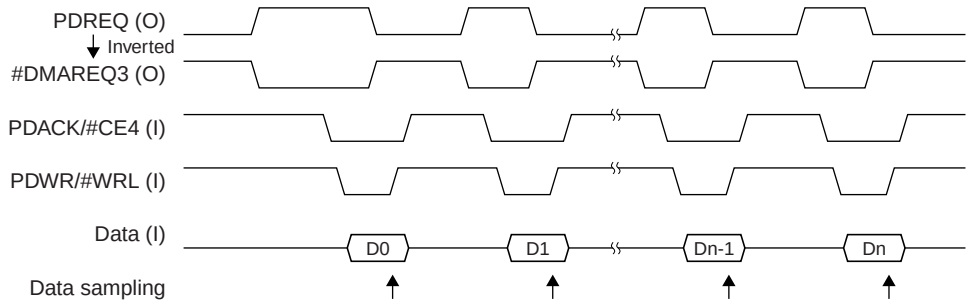


Figure VI.2.17 Transfer Waveforms in Asynchronous Single-Word DMA Transfer Mode - Writing

2) Reading operation

The Port interface starts reading operation in the Asynchronous single-word DMA transfer mode when the following register settings are established:

- DMA_Config_1.SingleWord bit = "1"
- Direction of the target endpoint = OUT

The Port interface starts data transfer on the DMA when "1" is written on the DMA_Control.DMA_Go bit. After data transfer starts on the DMA, it requests data transfer by asserting PDREQ (#DMAREQ3) to the C33 Core block (master) if any data exist at the connected endpoint. Turning PDACK (#CE4) to active starts outputting transferred data to the data bus. Have the C33 Core block (master) load the data while PDRD (#RD) is rising (when the DMA_Config_0.PRDWR_Level bit is set to "1"). This mode negates PDREQ (#DMAREQ3) after transferring 1-byte data (PDRD (#RD) becomes active). At this point, if any data still remain at the endpoint, it requests data transfer by asserting PDREQ (#DMAREQ3) to the C33 Core block (master). If there are no data left at the endpoint, PDREQ (#DMAREQ3) is not asserted and data transfer is rejected.
If any data is set to the DMA_Latency.DMA_Latency[3:0] bit other than "0h", this mode negates PDREQ (#DMAREQ3) once after completing transfer of 4-byte data, and does not assert PDREQ (#DMAREQ3) as long as 130 ns × N (N=DMA_Latency.DMA_Latency[3:0]).

If the DMA is set to the Countdown mode with DMA_Config_1.CountMode="1", the DMA completes data transfer when the DMA_Count_HH, HL, LH and LL registers reach 0000_0000h. To force the DMA to terminate data transfer, set the DMA_Control.DMA_Stop bit to "1". Note that forced termination of DMA transfer by writing to this bit may cause loss of data from those being transferred. To avoid it, first terminate the C33 Core block (master) and then terminate the macro's DMA transfer.

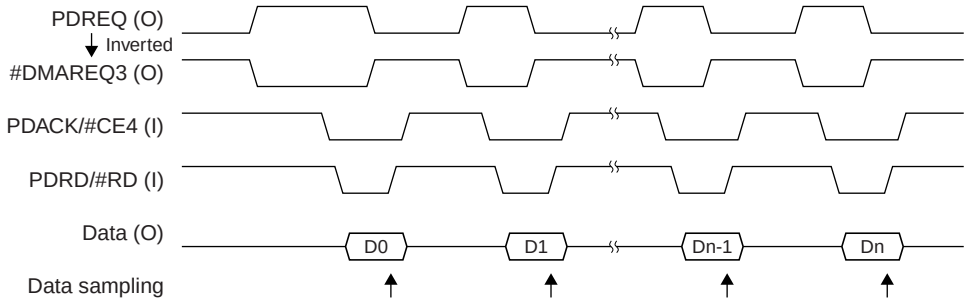


Figure VI.2.18 Transfer Waveforms in Asynchronous Single-Word DMA Transfer Mode - Reading

Snooze

This macro has Snooze function which enables very low power operation when USB is not active.
When the SNOOZE signal is asserted by writing "1" to USBSNZ (D5/0x300F20), the following procedure will be performed and allows to stop feeding 48 MHz clock after 5 clocks inputs.

- Disable USB differential comparator
- Allow asynchronous access for VBUS_Changed and NonJ registers. (Monitor the USB interface input pins)
- Mask Read/Write for synchronous registers
- Mask synchronous interrupt

This macro will resume after 5 clocks (oscillation must be stable) when the SNOOZE signal is negated.

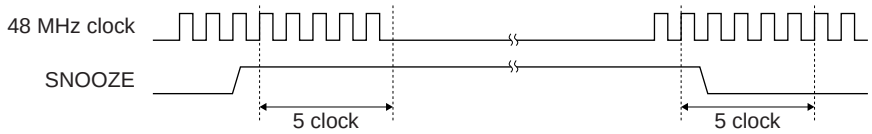


Figure VI.2.19 Snooze Sequence

Registers

List of Registers

• ***Italic & bold*** represents readable registers in suspend mode.

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000000	MainIntStat	R/(W)	0x00	***SIE_IntStat***	EPrIntStat	DMA_IntStat	FIFO_IntStat			EP0IntStat	RcvEP0SETUP
1000001	SIE_IntStat	R/(W)	0x00	***VBUS_Changed***	NonJ	DetectReset	DetectSuspend	RcvSOF	DetectJ		SetAddressCmp
1000002	EPrIntStat	R	0x00					EPdIntStat	EPcIntStat	EPbIntStat	EPaIntStat
1000003	DMA_IntStat	R/(W)	0x00							DMA_CountUp	DMA_Cmp
1000004	FIFO_IntStat	R/(W)	0x00	DescriptorCmp						FIFO_IN_Cmp	FIFO_OUT_Cmp
1000005											
1000006											
1000007	EP0IntStat	R/(W)	0x00			IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
1000008	EPaIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
1000009	EPbIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
100000A	EPcIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
100000B	EPdIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
100000C											
100000D											
100000E											
100000F											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000010	MainIntEnb	R/W	0x00	***EnSIE_IntStat***	EnEPrIntStat	EnDMA_IntStat	EnFIFO_IntStat			EnEP0IntStat	EnRcvEP0SETUP
1000011	SIE_IntEnb	R/W	0x00	***EnVBUS_Changed***	EnNonJ	EnDetectReset	EnDetectSuspend	EnRcvSOF	EnDetectJ		EnSetAddressCmp
1000012	EPrIntEnb	R/W	0x00					EnEPdIntStat	EnEPcIntStat	EnEPbIntStat	EnEPaIntStat
1000013	DMA_IntEnb	R/W	0x00							EnDMA_CountUp	EnDMA_Cmp
1000014	FIFO_IntEnb	R/W	0x00	EnDescriptorCmp						EnFIFO_IN_Cmp	EnFIFO_OUT_Cmp
1000015											
1000016											
1000017	EP0IntEnb	R/W	0x00			EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
1000018	EPaIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
1000019	EPbIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
100001A	EPcIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
100001B	EPdIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
100001C											
100001D											
100001E											
100001F											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000020	RevisionNum	R	0x12	***Revision Number[7:0]***							
1000021	USB_Control	R/W	0x00	DisBusDetect	EnAutoNego	InSUSPEND	StartDetectJ	SendWakeup			ActiveUSB
1000022	USB_Status	R	0xXX	***VBUS***	***1(FS)***					***LineState[1:0]***	
1000023	XcvrControl	R/W	0x81	RpuEnb						OpMode[1:0]	
1000024	USB_Test	R/W	0x00	EnUSB_Test				Test_SE0_NAK	Test_J	Test_K	Test_Packet
1000025	EPnControl	W	0x00	AllForceNAK	EPrForceSTALL	AllFIFO_Clr					EP0FIFO_Clr
1000026	EPrFIFO_Clr	W	0x00					EPdFIFO_Clr	EPcFIFO_Clr	EPbFIFO_Clr	EPaFIFO_Clr
1000027											
1000028											
1000029											
100002A											
100002B											
100002C											
100002D											
100002E	FrameNumber_H	R	0x80	FnInvalid					FrameNumber[10:8]		
100002F	FrameNumber_L	R	0x00	FrameNumber[7:0]							

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000030	EP0Setup_0	R	0x00								
1000031	EP0Setup_1	R	0x00								
1000032	EP0Setup_2	R	0x00								
1000033	EP0Setup_3	R	0x00								
1000034	EP0Setup_4	R	0x00								
1000035	EP0Setup_5	R	0x00								
1000036	EP0Setup_6	R	0x00								
1000037	EP0Setup_7	R	0x00								
1000038	USB_Address	R/W	0x00	AutoSetAddress	USB_Address[6:0]						
1000039	EP0Control	R/W	0x00	INxOUT							ReplyDescriptor
100003A	EP0ControlIN	R/W	0x00		EnShortPkt		ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
100003B	EP0ControlOUT	R/W	0x00	AutoForceNAK			ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
100003C											
100003D											
100003E											
100003F	EP0MaxSize	R/W	0x08		EP0MaxSize[6:3]						

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000040	EPaControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
1000041	EPbControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
1000042	EPcControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
1000043	EPdControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
1000044											
1000045											
1000046											
1000047											
1000048											
1000049											
100004A											
100004B											
100004C											
100004D											
100004E											
100004F											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000050	EPaMaxSize_H	R/W	0x00							EPaMaxSize[9:8]	
1000051	EPaMaxSize_L	R/W	0x00	EPaMaxSize[7:0]							
1000052	EPaConfig_0	R/W	0x00	INxOUT	ToggleMode	EnEndPoint		EndPointNumber[3:0]			
1000053	EPaConfig_1	R/W	0x00	ISO	ISO_CRCmode						
1000054	EPbMaxSize_H	R/W	0x00							EPbMaxSize[9:8]	
1000055	EPbMaxSize_L	R/W	0x00	EPbMaxSize[7:0]							
1000056	EPbConfig_0	R/W	0x00	INxOUT	ToggleMode	EnEndPoint		EndPointNumber[3:0]			
1000057	EPbConfig_1	R/W	0x00	ISO	ISO_CRCmode						
1000058	EPcMaxSize_H	R/W	0x00							EPcMaxSize[9:8]	
1000059	EPcMaxSize_L	R/W	0x00	EPcMaxSize[7:0]							
100005A	EPcConfig_0	R/W	0x00	INxOUT	ToggleMode	EnEndPoint		EndPointNumber[3:0]			
100005B	EPcConfig_1	R/W	0x00	ISO	ISO_CRCmode						
100005C	EPdMaxSize_H	R/W	0x00							EPdMaxSize[9:8]	
100005D	EPdMaxSize_L	R/W	0x00	EPdMaxSize[7:0]							
100005E	EPdConfig_0	R/W	0x00	INxOUT	ToggleMode	EnEndPoint		EndPointNumber[3:0]			
100005F	EPdConfig_1	R/W	0x00	ISO	ISO_CRCmode						

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000070	EPaStartAdrs_H	R/W	0x00					EPaStartAdrs[11:8]			
1000071	EPaStartAdrs_L	R/W	0x00	EPaStartAdrs[7:2]							
1000072	EPbStartAdrs_H	R/W	0x00					EPbStartAdrs[11:8]			
1000073	EPbStartAdrs_L	R/W	0x00	EPbStartAdrs[7:2]							
1000074	EPcStartAdrs_H	R/W	0x00					EPcStartAdrs[11:8]			
1000075	EPcStartAdrs_L	R/W	0x00	EPcStartAdrs[7:2]							
1000076	EPdStartAdrs_H	R/W	0x00					EPdStartAdrs[11:8]			
1000077	EPdStartAdrs_L	R/W	0x00	EPdStartAdrs[7:2]							
1000078											
1000079											
100007A											
100007B											
100007C											
100007D											
100007E											
100007F											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000080	CPU_JoinRd	R/W	0x00					JoinEPdRd	JoinEPcRd	JoinEPbRd	JoinEPaRd
1000081	CPU_JoinWr	R/W	0x00					JoinEPdWr	JoinEPcWr	JoinEPbWr	JoinEPaWr
1000082	EnEPnFIFO_Access	R/W	0x00							EnEPnFIFO_Wr	EnEPnFIFO_Rd
1000083	EPnFIFOforCPU	R/W	0xXX	EPnFIFOData[7:0]							
1000084	EPnRdRemain_H	R	0x00					EPnRdRemain[11:8]			
1000085	EPnRdRemain_L	R	0x00	EPnRdRemain[7:0]							
1000086	EPnWrRemain_H	R	0x00					EPnWrRemain[11:8]			
1000087	EPnWrRemain_L	R	0x00	EPnWrRemain[7:0]							
1000088	DescAdrs_H	R/W	0x00					DescAdrs[11:8]			
1000089	DescAdrs_L	R/W	0x00	DescAdrs[7:0]							
100008A	DescSize_H	R/W	0x00							DescSize[9:8]	
100008B	DescSize_L	R/W	0x00	DescSize[7:0]							
100008C											
100008D											
100008E											
100008F	DescDoor	R/W	0x00	DescMode[7:0]							

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
1000090	DMA_FIFO_Control	R/W	0x00	FIFO_Running	AutoEnShort						
1000091	DMA_Join	R/W	0x00					JoinEPdDMA	JoinEPcDMA	JoinEPbDMA	JoinEPaDMA
1000092	DMA_Control	R/W	0xX0	DMA_Running	PDREQ	PDACK		CounterClr		DMA_Stop	DMA_Go
1000093											
1000094	DMA_Config_0	R/W	0x00	ActivePort				PDREQ_Level	PDACK_Level	PDRDWR_Level	
1000095	DMA_Config_1	R/W	0x00	RcvLimitMode				SingleWord			CountMode
1000096											
1000097	DMA_Latency	R/W	0x00					DMA_Latency[3:0]			
1000098	DMA_Remain_H	R	0x00					DMA_Remain[11:8]			
1000099	DMA_Remain_L	R	0x00	DMA_Remain[7:0]							
100009A											
100009B											
100009C	DMA_Count_HH	R/W	0x00	DMA_Count[31:24]							
100009D	DMA_Count_HL	R/W	0x00	DMA_Count[23:16]							
100009E	DMA_Count_LH	R/W	0x00	DMA_Count[15:8]							
100009F	DMA_Count_LL	R/W	0x00	DMA_Count[7:0]							

Detailed Description of Registers

0x1000000 MainIntStat (Main Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
MainIntStat (Main interrupt status)	1000000 (B)	D7	**SIE_IntStat**	1	SIE interrupts	0	None	0	R	
		D6	**EPrIntStat**	1	EPr interrupts	0	None	0	R	
		D5	**DMA_IntStat**	1	DMA interrupts	0	None	0	R	
		D4	**FIFO_IntStat**	1	FIFO interrupts	0	None	0	R	
		D3–2	**–**	–				–	–	0 when being read.
		D1	**EP0IntStat**	1	EP0 interrupts	0	None	0	R	
		D0	**RcvEP0SETUP**	1	Receive EP0 SETUP	0	None	0	R(W)	

This register displays factors of interrupts having occurred in the USB function controller. This register has the bit indirectly showing interrupt factors and the bit directly showing interrupt factors.
The bit indirectly showing interrupt factors can be traced to the bit directly showing interrupt factors by reading the relevant status registers. The bit showing interrupt factors is read-only, and is automatically cleared by clearing the bit directly showing interrupt factors at the main source. The bits showing interrupt factors are writable, and the interrupt factors can be cleared by setting the relevant bits to "1". When the corresponding bits are enabled by the MainIntEnb register, setting the interrupt factor to "1" asserts the #INT signal, and causes an interruption of the CPU. Clearing all relevant interrupt factors negates the #INT signal.

D7 SIE_IntStat

Shows an interrupt factor indirectly.
When the SIE_IntStat register has an interrupt factor and the SIE_IntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1". Reading this bit is valid during snooze as well.

D6 EPrIntStat

Shows an interrupt factor indirectly.
When the EPrIntStat register has an interrupt factor and the EPrIntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1".

D5 DMA_IntStat

Shows an interrupt factor indirectly.
When the DMA_IntStat register has an interrupt factor and the DMA_IntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1".

D4 FIFO_IntStat

Shows an interrupt factor indirectly.
When the FIFO_IntStat register has an interrupt factor and the FIFO_IntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1".

D3–D2 Reserved

D1 EP0IntStat

Shows an interrupt factor indirectly.
When the EP0IntStat register has an interrupt factor and the EP0IntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1".

D0 RcvEP0SETUP

Shows an interrupt factor directly.
Set to "1" when the received data are set to the EP0Setup_0 to EP0Setup_7 after the set up stage has been completed. At the same time, the ForceSTALL bit, the ForceNAK bit and the ToggleStat bit of the EP0ControlIN and EP0ControlOUT registers are automatically set to "0", "1" and "1", respectively.

0x1000001 SIE_IntStat (SIE Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
SIE_IntStat (SIE interrupt status)	1000001 (B)	D7	VBUS_Changed	1	VBUS is changed	0	None	0	R(W)	
		D6	NonJ	1	Detect non J state	0	None	0	R(W)	
		D5	DetectReset	1	Detect USB reset	0	None	0	R(W)	
		D4	DetectSuspend	1	Detect USB suspend	0	None	0	R(W)	
		D3	RcvSOF	1	Receive SOF token	0	None	0	R(W)	
		D2	DetectJ	1	Detect J state	0	None	0	R(W)	
		D1	–	–				–	–	0 when being read.
		D0	SetAddressCmp	1	AutoSetAddress complete	0	None	0	R(W)	

This register displays the interrupts related to SIE.

D7 VBUS_Changed

Shows an interrupt factor directly.
When the condition of the VBUS terminal changes, this bit is set to "1".
Check the condition of the VBUS by the VBUS bit in the USB Status register. If the VBUS is "0", it shows that the cable is pulled off. This bit is valid during snooze as well.

D6 NonJ

Shows an interrupt factor directly.
Set to "1" when the status other than the J state is detected in the USB bus. This bit is valid when the InSUSPEND bit of the USB_Control register is set to "1".

D5 DetectReset

Shows an interrupt factor directly.
Set to "1" when the reset state of the USB is detected. This reset detection is valid when the ActiveUSB bit of the USB_Control register is set to "1".
When the AutoNegotiation function is not used, if this bit is set to "1", set to the DisBusDetect bit of the USB_Control register to "1", not to detect the succeeding reset wrongly by disabling detection of the reset/suspend state. Set the DisBusDetect bit to "0" (to be cleared) after completing the process for reset, to enable the reset/suspend state detection.
Refer to the item on the EnAutoNego bit of the USB_Control register, for the AutoNegotiation function.

D4 DetectSuspend

Shows an interrupt factor directly.
Set to "1" when the suspend state of the USB is detected.
After detecting the USB suspend state, setting the Snooze bit of the Macro Control register to "1" enables the IC to enter the snooze mode (to stop the built-in PLL oscillation).

D3 RcvSOF

Shows an interrupt factor directly.
Set to "1" when the SOF token is received.

D2 DetectJ

Shows an interrupt factor directly.
Set to "1" when the J-state is detected.

D1 Reserved

D0 SetAddressCmp

Shows an interrupt factor directly.
When the AutoSetAddress function (refer to the USB_Address register) ends normally, this bit is set to "1". The case when AutoSetAddress function ends normally is that when ACK is received during IN Transaction.

0x1000002 EPrIntStat (EPr Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPrIntStat (EPr interrupt status)	**1000002 (B)**	D7–4	**–**	–				–	–	0 when being read.
		D3	**EPdIntStat**	1	EPd interrupt	0	None	0	R	
		D2	**EPcIntStat**	1	EPc interrupt	0	None	0	R	
		D1	**EPbIntStat**	1	EPb interrupt	0	None	0	R	
		D0	**EPaIntStat**	1	EPa interrupt	0	None	0	R	

D7–D4 Reserved

D3 EPdIntStat

Shows an interrupt factor indirectly.
When the EPdIntStat register has an interrupt factor and the EPdIntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1".

D2 EPcIntStat

Shows an interrupt factor indirectly.
When the EPcIntStat register has an interrupt factor and the EPcIntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1".

D1 EPbIntStat

Shows an interrupt factor indirectly.
When the EPbIntStat register has an interrupt factor and the EPbIntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1".

D0 EPaIntStat

Shows an interrupt factor indirectly.
When the EPaIntStat register has an interrupt factor and the EPaIntEnb register bit corresponding to the interrupt factor is enabled, this bit is set to "1".

0x1000003 DMA_IntStat (DMA Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
DMA_IntStat	1000003	D7–2	–	–				–	–	0 when being read.
(DMA interrupt	(B)	D1	DMA_CountUp	1	DMA counter overflow	0	None	0	R(W)	
status)		D0	DMA_Cmp	1	DMA complete	0	None	0	R(W)	

This register displays the interrupt status of the DMA.

D7–D2 Reserved

D1 DMA_CountUp

Shows an interrupt factor directly.
Set to "1" when values of DMA_Count_HH, HL, LH and LL overflow while the DMA operates in the free run mode. Then values of DMA_Count_HH, HL, LH and LL return to "0", and the DMA operation continues.

D0 DMA_Cmp

Shows an interrupt factor directly.
Set to "1" when the DMA is stopped or completes the specified number of transfer operations and the end processing.

0x1000004 FIFO_IntStat (FIFO Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
FIFO_IntStat (FIFO interrupt status)	1000004 (B)	D7	DescriptorCmp	1	Descriptor complete	0	None	0	R(W)	
		D6–2	–	–				–	–	0 when being read.
		D1	FIFO_IN_Cmp	1	IN FIFO Complete	0	None	0	R(W)	
		D0	FIFO_OUT_Cmp	1	OUT FIFO complete	0	None	0	R(W)	

This register displays the interrupt status of the FIFO.

D7 DescriptorCmp

Shows an interrupt factor directly.
Set to "1" when as many data as specified in the DescriptorSize register have been replied in the Description Reply function.
And the OUT_TranNAK bit of the EP0IntStat register is set to "1" as well as this bit, when changing to the status stage takes place (the OUT token is received) before sending data up to the quantity specified in the DescriptorSize register.

D6–D2 Reserved

D1 FIFO_IN_Cmp

Shows an interrupt factor directly.
If the transfer direction of the endpoint bound to DMA (refer to the DMA_Join register) is the IN direction, this bit is set to "1" when the FIFO becomes empty after completion of the DMA transfer.

D0 FIFO_OUT_Cmp

Shows an interrupt factor directly.
If the transfer direction of the endpoint bound to DMA (refer to the DMA_Join register) is the OUT direction, this bit is set to "1" when the DMA transfer is completed.

0x1000007 EP0IntStat (EP0 Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EP0IntStat (EP0 interrupt status)	1000007 (B)	D7–6	–	–				–	–	0 when being read.
		D5	IN_TranACK	1	In transaction ACK	0	None	0	R(W)	
		D4	OUT_TranACK	1	Out transaction ACK	0	None	0	R(W)	
		D3	IN_TranNAK	1	In transaction NAK	0	None	0	R(W)	
		D2	OUT_TranNAK	1	Out transaction NAK	0	None	0	R(W)	
		D1	IN_TranErr	1	In transaction error	0	None	0	R(W)	
		D0	OUT_TranErr	1	Out transaction error	0	None	0	R(W)	

This register displays the interrupt status of the endpoint EP0.

D7–D6 Reserved

D5 IN_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the IN transaction.

D4 OUT_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the OUT transaction.

D3 IN_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the IN transaction.

D2 OUT_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the OUT transaction.

D1 IN_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the IN transaction, when an error occurred in the packet or when the handshake is failed in Time-Out.

D0 OUT_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x1000008 EPaIntStat (EPa Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPaIntStat (EPa interrupt status)	1000008 (B)	D7	–	–				–	–	0 when being read.
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)	
		D5	IN_TranACK	1	In transaction ACK	0	None	0	R(W)	
		D4	OUT_TranACK	1	Out transaction ACK	0	None	0	R(W)	
		D3	IN_TranNAK	1	In transaction NAK	0	None	0	R(W)	
		D2	OUT_TranNAK	1	Out transaction NAK	0	None	0	R(W)	
		D1	IN_TranErr	1	In transaction error	0	None	0	R(W)	
		D0	OUT_TranErr	1	Out transaction error	0	None	0	R(W)	

This register displays the interrupt status of the endpoint EPa.

D7 Reserved

D6 OUT_ShortACK

Shows an interrupt factor directly.
Set to "1" when a short packet is received and ACK is replied in the OUT transaction, OUT_TranACK and this bits are set to "1" at the same time.

D5 IN_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the IN transaction.

D4 OUT_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the OUT transaction.

D3 IN_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the IN transaction.

D2 OUT_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the OUT transaction.

D1 IN_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the IN transaction, when an error occurred in the packet or when the handshake is failed in Time-Out.

D0 OUT_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x1000009 EPbIntStat (EPb Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPbIntStat (EPb interrupt status)	1000009 (B)	D7	–	–				–	–	0 when being read.
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)	
		D5	IN_TranACK	1	In transaction ACK	0	None	0	R(W)	
		D4	OUT_TranACK	1	Out transaction ACK	0	None	0	R(W)	
		D3	IN_TranNAK	1	In transaction NAK	0	None	0	R(W)	
		D2	OUT_TranNAK	1	Out transaction NAK	0	None	0	R(W)	
		D1	IN_TranErr	1	In transaction error	0	None	0	R(W)	
		D0	OUT_TranErr	1	Out transaction error	0	None	0	R(W)	

This register displays the interrupt status of the endpoint EPb.

D7 Reserved

D6 OUT_ShortACK

Shows an interrupt factor directly.
Set to "1" when a short packet is received and ACK is replied in the OUT transaction, OUT_TranACK and this bits are set to "1" at the same time.

D5 IN_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the IN transaction.

D4 OUT_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the OUT transaction.

D3 IN_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the IN transaction.

D2 OUT_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the OUT transaction.

D1 IN_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the IN transaction, when an error occurred in the packet or when the handshake is failed in Time-Out.

D0 OUT_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x100000A EPcIntStat (EPc Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPcIntStat (EPc interrupt status)	100000A (B)	D7	–	–				–	–	0 when being read.
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)	
		D5	IN_TranACK	1	In transaction ACK	0	None	0	R(W)	
		D4	OUT_TranACK	1	Out transaction ACK	0	None	0	R(W)	
		D3	IN_TranNAK	1	In transaction NAK	0	None	0	R(W)	
		D2	OUT_TranNAK	1	Out transaction NAK	0	None	0	R(W)	
		D1	IN_TranErr	1	In transaction error	0	None	0	R(W)	
		D0	OUT_TranErr	1	Out transaction error	0	None	0	R(W)	

This register displays the interrupt status of the endpoint EPc.

D7 Reserved

D6 OUT_ShortACK

Shows an interrupt factor directly.
Set to "1" when a short packet is received and ACK is replied in the OUT transaction, OUT_TranACK and this bits are set to "1" at the same time.

D5 IN_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the IN transaction.

D4 OUT_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the OUT transaction.

D3 IN_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the IN transaction.

D2 OUT_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the OUT transaction.

D1 IN_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the IN transaction, when an error occurred in the packet or when the handshake is failed in Time-Out.

D0 OUT_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x100000B EPdIntStat (EPd Interrupt Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPdIntStat (EPd interrupt status)	100000B (B)	D7	–	–				–	–	0 when being read.
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)	
		D5	IN_TranACK	1	In transaction ACK	0	None	0	R(W)	
		D4	OUT_TranACK	1	Out transaction ACK	0	None	0	R(W)	
		D3	IN_TranNAK	1	In transaction NAK	0	None	0	R(W)	
		D2	OUT_TranNAK	1	Out transaction NAK	0	None	0	R(W)	
		D1	IN_TranErr	1	In transaction error	0	None	0	R(W)	
		D0	OUT_TranErr	1	Out transaction error	0	None	0	R(W)	

This register displays the interrupt status of the endpoint EPd.

D7 Reserved

D6 OUT_ShortACK

Shows an interrupt factor directly.
Set to "1" when a short packet is received and ACK is replied in the OUT transaction, OUT_TranACK and this bits are set to "1" at the same time.

D5 IN_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the IN transaction.

D4 OUT_TranACK

Shows an interrupt factor directly.
Set to "1" when ACK is received in the OUT transaction.

D3 IN_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the IN transaction.

D2 OUT_TranNAK

Shows an interrupt factor directly.
Set to "1" when NAK is replied in the OUT transaction.

D1 IN_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the IN transaction, when an error occurred in the packet or when the handshake is failed in Time-Out.

D0 OUT_TranErr

Shows an interrupt factor directly.
Set to "1" when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x1000010 MainIntEnb (Main Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
MainIntEnb (Main interrupt enable)	1000010 (B)	D7	En**SIE_IntStat**	1	Enabled	0	Disabled	0	R/W	
		D6	En**EPrIntStat**					0	R/W	
		D5	**EnDMA_IntStat**					0	R/W	
		D4	**EnFIFO_IntStat**					0	R/W	
		D3–2	**–**	–				–	–	0 when being read.
		D1	En**EP0IntStat**	1	Enabled	0	Disabled	0	R/W	
		D0	En**RcvEP0SETUP**					0	R/W	

This register enables/disables assertion of the interrupt signal (#INT) with the interrupt factor of the MainIntStat register. Setting the corresponding bit to "1" enables interrupt. EnSIE_IntStat bit is valid during snooze as well.

0x1000011 SIE_IntEnb (SIE Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
SIE_IntEnb (SIE interrupt enable)	1000011 (B)	D7	**EnVBUS_Changed**	1	Enabled	0	Disabled	0	R/W	
		D6	**EnNonJ**					0	R/W	
		D5	**EnDetectReset**					0	R/W	
		D4	**EnDetectSuspend**					0	R/W	
		D3	**EnRcvSOF**					0	R/W	
		D2	**EnDetectJ**					0	R/W	
		D1	**–**	–				–	–	0 when being read.
		D0	**EnSetAddressCmp**	1	Enabled	0	Disabled	0	R/W	

This register enables/disables assertion of the SIE_IntStat bit of the MainIntStat register with the interrupt factor of the SIE_IntStat register. EnVBUS_Changed and EnNonJ bits are valid during snooze as well.

0x1000012 EPrIntEnb (EPr Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPrIntEnb (EPr interrupt enable)	1000012 (B)	D7–4	**–**	–				–	–	0 when being read.
		D3	**EnEPdIntStat**	1	Enabled	0	Disabled	0	R/W	
		D2	**EnEPcIntStat**					0	R/W	
		D1	**EnEPbIntStat**					0	R/W	
		D0	**EnEPaIntStat**					0	R/W	

This register enables/disables assertion of the EPrIntStat bit of the MainIntStat register with the interrupt factor of the EPrIntStat register.

0x1000013 DMA_IntEnb (DMA Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
DMA_IntEnb (DMA interrupt enable)	1000013 (B)	D7–2	**–**	–				–	–	0 when being read.
		D1	**EnDMA_CountUp**	1	Enabled	0	Disabled	0	R/W	
		D0	**EnDMA_Cmp**					0	R/W	

This register enables/disables assertion of the DMA_IntStat bit of the MainIntStat register with the interrupt factor of the DMA_IntStat register.

0x1000014 FIFO_IntEnb (FIFO Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
FIFO_IntEnb (FIFO interrupt enable)	1000014 (B)	D7	**EnDescriptorCmp**	1	Enabled	0	Disabled	0	R/W	
		D6–2	**–**	–				–	–	0 when being read.
		D1	**EnFIFO_IN_Cmp**	1	Enabled	0	Disabled	0	R/W	
		D0	**EnFIFO_OUT_Cmp**					0	R/W	

This register enables/disables assertion of the FIFO_IntStat bit of the MainIntStat register with the interrupt factor of the FIFO_IntStat register.

0x1000017 EP0IntEnb (EP0 Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EP0IntEnb (EP0 interrupt enable)	1000017 (B)	D7–6	–	–				–	–	0 when being read.
		D5	EnIN_TranACK	1	Enabled	0	Disabled	0	R/W	
		D4	EnOUT_TranACK					0	R/W	
		D3	EnIN_TranNAK					0	R/W	
		D2	EnOUT_TranNAK					0	R/W	
		D1	EnIN_TranErr					0	R/W	
		D0	EnOUT_TranErr					0	R/W	

This register enables/disables assertion of the EP0IntStat bit of the MainIntStat register with the interrupt factor of the EP0IntStat register.

0x1000018 EPaIntEnb (EPa Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPaIntEnb (EPa interrupt enable)	1000018 (B)	D7	–	–				–	–	0 when being read.
		D6	EnOUT_ShortACK	1	Enabled	0	Disabled	0	R/W	
		D5	EnIN_TranACK					0	R/W	
		D4	EnOUT_TranACK					0	R/W	
		D3	EnIN_TranNAK					0	R/W	
		D2	EnOUT_TranNAK					0	R/W	
		D1	EnIN_TranErr					0	R/W	
		D0	EnOUT_TranErr					0	R/W	

This register enables/disables assertion of the EPaIntStat bit of the EPrIntStat register with the interrupt factor of the EPaIntStat register.

0x1000019 EPbIntEnb (EPb Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPbIntEnb (EPb interrupt enable)	1000019 (B)	D7	–	–				–	–	0 when being read.
		D6	EnOUT_ShortACK	1	Enabled	0	Disabled	0	R/W	
		D5	EnIN_TranACK					0	R/W	
		D4	EnOUT_TranACK					0	R/W	
		D3	EnIN_TranNAK					0	R/W	
		D2	EnOUT_TranNAK					0	R/W	
		D1	EnIN_TranErr					0	R/W	
		D0	EnOUT_TranErr					0	R/W	

This register enables/disables assertion of the EPbIntStat bit of the EPrIntStat register with the interrupt factor of the EPbIntStat register.

0x100001A EPcIntEnb (EPc Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPcIntEnb (EPc interrupt enable)	100001A (B)	D7	–	–				–	–	0 when being read.
		D6	EnOUT_ShortACK	1	Enabled	0	Disabled	0	R/W	
		D5	EnIN_TranACK					0	R/W	
		D4	EnOUT_TranACK					0	R/W	
		D3	EnIN_TranNAK					0	R/W	
		D2	EnOUT_TranNAK					0	R/W	
		D1	EnIN_TranErr					0	R/W	
		D0	EnOUT_TranErr					0	R/W	

This register enables/disables assertion of the EPcIntStat bit of the EPrIntStat register with the interrupt factor of the EPcIntStat register.

0x100001B EPdIntEnb (EPd Interrupt Enable)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPdIntEnb (EPd interrupt enable)	100001B (B)	D7	–	–				–	–	0 when being read.
		D6	EnOUT_ShortACK	1	Enabled	0	Disabled	0	R/W	
		D5	EnIN_TranACK					0	R/W	
		D4	EnOUT_TranACK					0	R/W	
		D3	EnIN_TranNAK					0	R/W	
		D2	EnOUT_TranNAK					0	R/W	
		D1	EnIN_TranErr					0	R/W	
		D0	EnOUT_TranErr					0	R/W	

This register enables/disables assertion of the EPdIntStat bit of the EPrIntStat register with the interrupt factor of the EPdIntStat register.

0x1000020 RevisionNum (Revision Number)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
RevisionNum (Revision number)	**1000020 (B)**	D7	**RevisionNum[7]**	Revision number (0x12)	0	R	
		D6	**RevisionNum[6]**		0		
		D5	**RevisionNum[5]**		0		
		D4	**RevisionNum[4]**		1		
		D3	**RevisionNum[3]**		0		
		D2	**RevisionNum[2]**		0		
		D1	**RevisionNum[1]**		1		
		D0	**RevisionNum[0]**		0		

This register shows the revision number of the USB function controller. This register is valid during snooze as well.

0x1000021 USB_Control (USB Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
USB_Control (USB control register)	**1000021 (B)**	D7	**DisBusDetect**	1	Disable bus detect	0	Enable bus detect	0	R/W	
		D6	**EnAutoNego**	1	Enable auto negotiation	0	Disable auto negotiation	0	R/W	
		D5	**InSUSPEND**	1	Monitor NonJ	0	Do nothing	0	R/W	
		D4	**StartDetectJ**	1	Start J-state detection	0	Do nothing	0	R/W	
		D3	**SendWakeup**	1	Send remote wakeup signal	0	Do nothing	0	R/W	
		D2–1	**–**	–				–	–	0 when being read.
		D0	**ActiveUSB**	1	Activate USB	0	Disactivate USB	0	R/W	

The operation setting is done for the USB.

D7 DisBusDetect

Setting this bit to "1" disables the automatic detection of the USB reset/suspend state.
When this bit is set to "0" (to be cleared), activities on the USB bus is monitored to detect the reset/suspend state.
If the bus activities cannot be detected within 3 ms, the USB is determined to be suspend state. And if "SE0" longer than 2.5 microseconds is detected, the USB is determined to be reset state, and then the relevant interrupt factor (DetectReset, DetectSuspend) is set.
If the DetectReset or the DetectSuspend bit is set to "1", set the DisBusDetect bit to "1" to disable detection when the reset/suspend state is continued.
When using the Auto Negotiation function, do not set this bit to "1".

D6 EnAutoNego

This bit enables the Auto Negotiation function. The Auto Negotiation function automates the work sequence to be done after detecting the reset, from the end of the speed negotiation to determination of the speed mode. Refer to the section describing operations for details of the Auto Negotiation.

D5 InSUSPEND

This bit enables the detection of the NonJ state. If the USB suspend state is detected and f/w is prepared. set this bit to "1". To return from the suspended state, set this bit to "0" (to be cleared).
The NonJ state can be detected only when this bit is set. If the Snooze function is not be used when the USB goes into the suspend state, set this bit.
Refer to description on operations for how to use the Auto Negotiation function.

D4 StartDetectJ

This bit enables the detection of the J state. After setting this bit and J-state is coming, DetectJ interrupt is set when EnDetectJ is set.

D3 SendWakeup

Setting this bit to "1" outputs the RemoteWakeup signal (K) to the USB port.
Within the time between 1 ms and 15 ms after starting to send the RemoteWakeup signal, set this bit to "0" (to be cleared) to stop sending the signals.

D2–D1 Reserved

D0 ActiveUSB

Since this bit is set to "0" (to be cleared) after hardware reset, all USB functions are stopped. The operation as a USB will be enabled by setting this bit to "1" after completing the setting of this IC.

0x1000022 USB_Status (USB Status)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
USB_Status (USB status register)	**1000022 (B)**	D7	**VBUS**	1	VBUS=High	0	VBUS=Low	X	R	
		D6	**FS**	1	FS mode (fixed)	0	–	1	R	
		D5–2	**–**	–				–	–	
		D1	**LineState[1]**	LineState[1:0]		DP/DM		X	R	
		D0	**LineState[0]**	1	1	SE1		X		
				1	0	K				
				0	1	J				
				0	0	SE0				

This register displays the status related to the USB.
This register is valid during snooze as well.

D7 VBUS

This bit displays the status of the USBVBUS pin.

D6 FS

Returns always "1" (FS mode).

D5–D2 Reserved

D1–D0 LineState[1:0]

Shows the signal status on the USB cable.
Shows the value received by the FS receiver of the DP/DM.

LineState

LineState[1]	LineState[0]	DP/DM
1	1	SE1
1	0	K
0	1	J
0	0	SE0

0x1000023 XcvrControl (Xcvr Control)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks
XcvrControl (Xcvr control register)	**1000023 (B)**	D7	**RpuEnb**	1 Enable pull-up		0 Disable pull-up	1	R/W	
		D6–2	**–**	–			–	–	0 when being read.
		D1	**OpMode[1]**	OpMode[1:0]		Operation mode	0	R/W	
		D0	**OpMode[0]**	1	1	reserved	1		
				1	0	Disable bitstuffing and NRZI encoding			
				0	1	Non-driving			
				0	0	Normal operation			

The operation setting is done for the Transceiver macro.

D7 RpuEnb

This bit enables the D+ pull-up resistor.

D6–D2 Reserved

D1–D0 OpMode

This bit sets the operation mode of the Transceiver macro.
This bit needs not be set up normally, excluding when the USB cable is pulled off (*) and during the test mode.

OpMode

OpMode[1]	OpMode[0]	Operation mode
1	1	reserved
1	0	Disable bitstuffing and NRZI encoding
0	1	Non-driving
0	0	Normal operation

* When the USB cable is pulled off, it is recommended to set this register to "41h".

0x1000024 USB_Test (USB Test)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
USB_Test (USB test)	1000024 (B)	D7	**EnUSB_Test**	1	Enable USB test	0	Do nothing	0	R/W	
		D6–4	**–**	–				–	–	0 when being read.
		D3	**Test_SE0_NAK**	1	Test_SE0_NAK	0	Do nothing	0	R/W	
		D2	**Test_J**	1	Test_J	0	Do nothing	0	R/W	
		D1	**Test_K**	1	Test_K	0	Do nothing	0	R/W	
		D0	**Test_Packet**	1	Test_Packet	0	Do nothing	0	R/W	

The operation setting is done in this register for the USB test mode. Set the bit corresponding to the test mode specified by the SetFeature request, and after completing the status stage, set the EnUSB_Test bit to "1" and perform the test mode operation defined by the USB standard.

D7 EnUSB_Test

When this bit is set to "1", if one of the lower order 4 bits in the USB_Test register is set to "1", the IC will go into the test mode corresponding to the bit. When performing the test mode, the DisBusDetect bit of the USB_Control register must be set to "1" not to detect the USB suspend and the reset before performing the test. In addition, set the EnAutoNego bit of the USB_Control register to "0" (to be cleared) to disable the Auto Negotiation.
Note that the change to the test mode must be done after completing the status stage for the SetFeature request.

D6–D4 Reserved

D3 Test_SE0_NAK

By setting this bit to "1" and the EnUSB_Test bit to "1", the Test_SE0_NAK test mode can start.

D2 Test_J

By setting this bit to "1" and the EnUSB_Test bit to "1", the Test_J test mode can start. In this test mode, before EnUSB_Test bit is set to "1", set OpMode to 10 (Disable Bitstuffing and NRZI encoding).

D1 Test_K

By setting this bit to "1" and the EnUSB_Test bit to "1", the Test_K test mode can start. In this test mode, before EnUSB_Test bit is set to "1", set OpMode to 10 (Disable Bitstuffing and NRZI encoding).

D0 Test_Packet

By setting this bit to "1", the Test_Packet test mode can start.
Since this test mode uses the endpoint EPc, set the followings.

(1) Set the MaxPacketSize of the endpoint EPc to 64 or more, the direction of transfer to IN and the EndPointNumber to 0xF to make the endpoint be ready to use. And allocate the FIFO of the endpoint EPc for 64 bytes or more.
(2) Do not overlap the above setting with the settings of the endpoints EPa and EPb.
Or clear the EPaConfig.EnEndPoint bit and EPbConfig.EnEndPoint bit.
(3) Clear the FIFO of the EPc and write data for the following test packet into this FIFO.
(4) Set the EnIN_TranErr of the EPcIntEnb register to "0" (clear this bit).
IN_TranErr status is set to "1" at every time the Test Packet transmission completes.

The data to write into the FIFO in the packet transmission test mode are the following 53 bytes.

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA,
0xAA, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE, 0xEE,
0xEE, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0xBF, 0xDF,
0xEF, 0xF7, 0xFB, 0xFD, 0xFC, 0x7E, 0xBF, 0xDF,
0xEF, 0xF7, 0xFB, 0xFD, 0x7E
```

Since the SIE adds the PID and CRC to the test packet when sending it, the data to write into the FIFO are from "the data after the DATA 0 PID" to "the data before the CRC16" that are described as the test packet data in the USB standard Rev.2.0. (Note that Test Packet is defined only HS mode in USB specification.)

0x1000025 EPnControl (Endpoint Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPnControl (Endpoint control)	1000025 (B)	D7	AllForceNAK	1	Set all ForceNAK	0	Do nothing	0	W	0 when being read.
		D6	EPrForceSTALL	1	Set EP's ForceSTALL	0	Do nothing	0	W	
		D5	AllFIFO_Clr	1	Clear all FIFO	0	Do nothing	0	W	
		D4–1	–	–				–	–	
		D0	EP0FIFO_Clr	1	Clear EP0 FIFO	0	Do nothing	0	W	

This register sets operations of entire endpoints, and display them.

D7 AllForceNAK

Set the ForceNAK bit of all endpoints to "1".

D6 EPrForceSTALL

Set the ForceSTALL bit of EPa, EPb, EPc and EPd endpoints to "1".

D5 AllFIFO_Clr

Clear the FIFOs of all endpoints. After setting the area of the respective endpoints, be sure to set this bit to "1" to clear the FIFOs of all endpoints. This bit is automatically set "0" (to be cleared) after completing the FIFO clear operation.

Do not set this bit to "1" during start operation of the general port (when the DMA_Running bit of the DMA_Control register is 1). Otherwise, a malfunction may occur.

D4–D1 Reserved

D0 EP0FIFO_Clr

Clear the FIFO of the endpoint EP0. This bit is automatically set "0" (to be cleared) after completing the FIFO clear operation.

0x1000026 EPrFIFO_Clr (EPr FIFO Clear)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPrFIFO_Clr (EPr FIFO clear)	1000026 (B)	D7–4	–	–				–	–	0 when being read.
		D3	**EPdFIFO_Clr**	1	Clear EPd FIFO	0	Do nothing	0	W	
		D2	**EPcFIFO_Clr**	1	Clear EPc FIFO	0	Do nothing	0	W	
		D1	**EPbFIFO_Clr**	1	Clear EPb FIFO	0	Do nothing	0	W	
		D0	**EPaFIFO_Clr**	1	Clear EPa FIFO	0	Do nothing	0	W	

This register sets operations of entire endpoints, and display them.

D7–D4 Reserved

D3 EPdFIFO_Clr

Clear the FIFO of the endpoint EPd. This bit is automatically set "0" (to be cleared) after completing the FIFO clear operation.
Do not set this bit to "1" when the endpoint EPd is connected to the general port (the JoinEPdDMA bit of the DMA_Join register is set to "1") and the start operation of the general port is being done (when the DMA_Running bit of the DMA_Control register is "1"). Otherwise, a malfunction may occour.

D2 EPcFIFO_Clr

Clear the FIFO of the endpoint EPc. This bit is automatically set "0" (to be cleared) after completing the FIFO clear operation.
Do not set this bit to "1" when the endpoint EPc bis connected to the general port (the JoinEPcDMA bit of the DMA_Join register is set to "1") and the start operation of the general port is being done (when the DMA_Running bit of the DMA_Control register is "1"). Otherwise, a malfunction may occour.

D1 EPbFIFO_Clr

Clear the FIFO of the endpoint EPb. This bit is automatically set "0" (to be cleared) after completing the FIFO clear operation.
Do not set this bit to "1" when the endpoint EPb is connected to the general port (the JoinEPbDMA bit of the DMA_Join register is set to "1") and the start operation of the general port is being done (when the DMA_Running bit of the DMA_Control register is "1"). Otherwise, a malfunction may occour.

D0 EPaFIFO_Clr

Clear the FIFO of the endpoint EPa. This bit is automatically set "0" (to be cleared) after completing the FIFO clear operation.
Do not set this bit to "1" when the endpoint EPa is connected to the general port (the JoinEPaDMA bit of the DMA_Join register is set to "1") and the start operation of the general port is being done (when the DMA_Running bit of the DMA_Control register is "1"). Otherwise, a malfunction may occour.

0x100002E FrameNumber_H (Frame Number HIGH)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
FrameNumber_H (Frame number high)	100002E (B)	D7	FnInvalid	1 Invalid frame number 0 Valid frame number	1	R	
		D6–3	–	–	–	–	0 when being read.
		D2	FrameNumber[10]	Frame number high	0	R	
		D1	FrameNumber[9]		0		
		D0	FrameNumber[8]		0		

This register displays the USB frame number that is updated every time the SOF token is received. When frame numbers are acquired, the FrameNumber_H and the FrameNumber_L registers must be accessed as a pair. When accessing them, access the FrameNumber_H register first.

D7 FnInvalid

When an error occurs in the received SOF packet, this bit is set to "1".

D6–D3 Reserved

D2–D0 FrameNumber[10:8]

The upper order 3 bits in the FrameNumber field of the received SOF packet are stored in these bits.

0x100002F FrameNumber_L (Frame Number LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
FrameNumber_L (Frame number low)	100002F (B)	D7	FrameNumber[7]	Frame number low	0	R	
		D6	FrameNumber[6]		0		
		D5	FrameNumber[5]		0		
		D4	FrameNumber[4]		0		
		D3	FrameNumber[3]		0		
		D2	FrameNumber[2]		0		
		D1	FrameNumber[1]		0		
		D0	FrameNumber[0]		0		

D7–D0 FrameNumber[7:0]

The lower order 8 bits in the FrameNumber field of the received SOF packet are stored in these bits.

0x1000030 EP0Setup_0 (EP0 Setup 0)–0x1000037 EP0Setup_7 (EP0 Setup 7)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EP0Setup_0	1000030	D7	EP0Setup_n[7]	Endpoint 0 set-up data 0	0	R	
(EP0 set-up 0)	|	D6	EP0Setup_n[6]	|	0		
|	1000037	D5	EP0Setup_n[5]	Endpoint 0 set-up data 7	0		
EP0Setup_7	(B)	D4	EP0Setup_n[4]		0		
(EP0 set-up 7)		D3	EP0Setup_n[3]		0		
		D2	EP0Setup_n[2]		0		
		D1	EP0Setup_n[1]		0		
		D0	EP0Setup_n[0]		0		

Eight-byte data received at the endpoint EP0 setup stage are stored from the EP0Setup_0 sequentially.

0x1000030 EP0Setup_0

BmRequestType is set.

0x1000031 EP0Setup_1

BRequest is set.

0x1000032 EP0Setup_2

The lower order 8 bits in Wvalue are set.

0x1000033 EP0Setup_3

The upper order 8 bits in Wvalue are set.

0x1000034 EP0Setup_4

The lower order 8 bits in WIndex are set.

0x1000035 EP0Setup_5

The upper order 8 bits in WIndex are set.

0x1000036 EP0Setup_6

The lower order 8 bits in WLength are set.

0x1000037 EP0Setup_7

The upper order 8 bits in WLength are set.

0x1000038 USB_Address (USB Address)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
USB_Address (USB address)	1000038 (B)	D7	AutoSetAddress	1 Auto set address 0 Do nothing	0	R/W	
		D6	USB_Address[6]	USB address	0	R/W	
		D5	USB_Address[5]		0		
		D4	USB_Address[4]		0		
		D3	USB_Address[3]		0		
		D2	USB_Address[2]		0		
		D1	USB_Address[1]		0		
		D0	USB_Address[0]		0		

This register sets up the USB address.

D7 AutoSetAddress

Sets up the USB Address automatically. If this bit is set to "1" after receiving the SetAddress request and before implementing the status stage, the address received by the SetAddress request will be written into the USB_Address register when the status stage completes.

The processing procedure of the SetAddress request using this function is as follows.

(1) The SETUP transaction of the SetAddress request completes.
The RcvEP0SETUP bit of the MainIntStat register is set to "1". Read the EP0Setup_0–7 registers and interpret the request.
(2) Set the AuroSetAddress bit.
(3) Set the INxOUT bit of the EP0Control register.
(4) Clear the ForceNAK bit of the EP0ControlIN register, and set the EnShortPkt bit.
(5) Wait for the end of the status stage.

D6–D0 USB_Address

These bits set up the USB address.

The USB address is written automatically by the AutoSetAddress function. Or it can be written.

0x1000039 EP0Control (EP0 Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EP0Control (EP0 control)	1000039 (B)	D7	INxOUT	1	IN	0	OUT	0	R/W	
		D6–1	–	–				–	–	0 when being read.
		D0	ReplyDescriptor	1	Reply descriptor	0	Do nothing	0	W	

This register sets up the endpoint EP0.

D7 INxOUT

Set the transfer direction of the endpoint EP0.
Judging from the request received at the setup stage, set a value in this bit.
If the data stage exists, set the transfer direction at the data stage into this bit. As the setup of the ForceNAK bits of the EP0ControlIN and EP0ControlOUT registers completes when the setup stage completes, clear them during execution of the data stage or the status stage.
After the data stage is completed, set this bit again conforming to the direction of the status stage. When the transfer direction of the data stage is IN, the transfer direction of the status stage is OUT. Therefore, set this bit to "0". When the transfer direction of the data stage is OUT, or there is no data stage, the transfer direction of the status stage is IN. Therefore, clear the FIFO of the endpoint EP0, and set this bit to "1".
For the IN or OUT transactions which have a transfer direction different from that of this bit, NAK response is done. However, if the ForceSTALL bit of the EP0ControlIN or EP0ControlOUT register with the transaction direction corresponding to the above one, is set, the STALL response will be done.

D6–D1 Reserved

D0 ReplyDescriptor

Executes the Descriptor reply function.
If this bit is set to "1", this bit replies as much Descriptor data as specified as MaxPacketSize from the FIFO, responding to the IN transaction of the endpoint EP0. The Descriptor data start from the address specified in the DescAdrs_H, L register, and its data size is specified in the DescSize_H, L register. Since these setting values are updated during execution of the Descriptor reply function, set these setting values every time setting the ReplyDescriptor bit.
In every transaction, the DescAdrs_H, L register is incremented as many as the number of data that were sent, while the DescSize_H, L register is decremented as many as the number of data that were sent.
When the data transmission ends after sending as many data as specified in the DescSize_H, L or when a transaction other than the IN transaction is done, the Descriptor reply function ends, the ReplyDescriptor bit is set to "0" (to be cleared) and the IN_TranACK bit of the EPnIntStat register is set to "1".
Refer to the section describing operations, for details.

0x100003A EP0ControlIN (EP0 Control IN)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EP0ControlIN (EP0 control IN)	100003A (B)	D7	–	–				–	–	0 when being read.
		D6	EnShortPkt	1	Enable short packet	0	Do nothing	0	R/W	
		D5	–	–				–	–	0 when being read.
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	R/W	
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	R/W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	

This register sets the operations related to the IN transaction of the endpoint EP0 and displays their status.

D7 Reserved

D6 EnShortPkt

Setting this bit to "1" enables to send the data within the FIFO that is less than the quantity specified for the MaxPacketSize, as a short packet for the IN transaction of the endpoint EP0. When the IN transaction that transmitted short packets completes, this bit is automatically set to "0" (to be cleared). When a packet of the max packet size is transmitted, this bit is not cleared.
If this bit is set to "1" when the FIFO has no data, a zero-length packet can be transmitted for the IN token from the host. If the data is written into the FIFO that is in the transmission process with the packet to which this bit is set, that data may be included in transmission. Therefore, do not write into the FIFO until the packet transmission completes and this bit is cleared.

D5 Reserved

D4 ToggleStat

Shows the status of the toggle sequence bit in the IN transaction of the endpoint EP0.

D3 ToggleSet

Sets the toggle sequence bit in the IN transaction of the endpoint EP0, to "1".

D2 ToggleClr

Sets the toggle sequence bit in the IN transaction of the endpoint EP0, to "0" (clear).

D1 ForceNAK

If this bit is set to "1", the NAK response is done for the IN transaction of the endpoint EP0, regardless of the FIFO data quantity.
When the RcvEP0SETUP bit of the MainIntStat register is set to "1" after completion of the setup stage, this bit is set to "1", and this bit cannot be set to "0" (to be cleared) as long as the RcvEP0SETUP bit is "1". When the IN transaction that transmitted short packets completes, this bit is set to "1".
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

D0 ForceSTALL

If this bit is set to "1", the STALL response is done for the IN transaction of the endpoint EP0. This bit has a priority over the setting of the ForceNAK bit.
When the RcvEP0SETUP bit of the MainIntStat register is set to "1" after completion of the setup stage, this bit is set to "0" (to be cleared), and this bit cannot be set to "1" as long as the RcvEP0SETUP bit is "1".
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

0x100003B EP0ControlOUT (EP0 Control OUT)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EP0ControlOUT (EP0 control OUT)	100003B (B)	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	
		D6–5	–	–				–	–	0 when being read.
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	

This register sets the operations related to the OUT transaction of the endpoint EP0 and displays their status.

D7 AutoForceNAK

Sets the ForceNAK bit of this register to "1" when the OUT transaction of the endpoint EP0 completes normally.

D6–D5 Reserved

D4 ToggleStat

Shows the status of the toggle sequence bit in the OUT transaction of the endpoint EP0.

D3 ToggleSet

Sets the toggle sequence bit in the OUT transaction of the endpoint EP0, to "1".

D2 ToggleClr

Sets the toggle sequence bit in the OUT transaction of the endpoint EP0, to "0" (clear).

D1 ForceNAK

If this bit is set to "1", the NAK response is done for the OUT transaction of the endpoint EP0, regardless of the FIFO space capacity.
When the RcvEP0SETUP bit of the MainIntStat register is set to "1" after completion of the setup stage, this bit is set to "1", and this bit cannot be set to "0" (to be cleared) as long as the RcvEP0SETUP bit is "1". When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

D0 ForceSTALL

If this bit is set to "1", the STALL response is done for the OUT transaction of the endpoint EP0. This bit has a priority over the setting of the ForceNAK bit.
When the RcvEP0SETUP bit of the MainIntStat register is set to "1" after completion of the setup stage, this bit is set to "0" (to be cleared), and this bit cannot be set to "1" as long as the RcvEP0SETUP bit is "1".
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

0x100003F EP0MaxSize (EP0 Max Packet Size)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EP0MaxSize (EP0 max packet size)	**100003F (B)**	D7	**–**	–	–	–	0 when being read.
		D6	**EP0MaxSize[6]**	Endpoint EP0 max packet size	0	R/W	
		D5	**EP0MaxSize[5]**		0		
		D4	**EP0MaxSize[4]**		0		
		D3	**EP0MaxSize[3]**		1		
		D2–0	**–**	–	–	–	0 when being read.

EP0MaxSize[6:3]

This register sets the MaxPacketSize of the endpoint EP0.
The size of this endpoint can be set to 8, 16, 32 or 64 bytes.

0x1000040 EPaControl (EPa Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPaControl (EPa control)	1000040 (B)	D7	**AutoForceNAK**	1	Auto force NAK	0	Do nothing	0	R/W	
		D6	**EnShortPkt**	1	Enable short packet	0	Do nothing	0	R/W	
		D5	**DisAF_NAK_Short**	1	Disable auto force	0	Auto force NAK short	0	R/W	
		D4	**ToggleStat**	Toggle sequence bit				0	R	
		D3	**ToggleSet**	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	**ToggleClr**	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	**ForceNAK**	1	Force NAK	0	Do nothing	0	R/W	
		D0	**ForceSTALL**	1	Force STALL	0	Do nothing	0	R/W	

This register sets operations of the endpoint EPa.

D7 AutoForceNAK

Sets the ForceNAK bit of this register to "1" when the transaction of the endpoint EPa completes normally.

D6 EnShortPkt

Setting this bit to "1" enables to send the data within the FIFO that is less than the quantity specified for the MaxPacketSize, as a short packet for the IN transaction of the endpoint EPa. When the IN transaction that transmitted short packets completes, this bit is automatically set to "0" (to be cleared). When a packet of the max packet size is transmitted, this bit is not cleared.
If this bit is set to "1" when the FIFO has no data, a zero-length packet can be transmitted for the IN token from the host. If the data is written into the FIFO that is in the transmission process with the packet to which this bit is set, that data may be included in transmission. Therefore, do not write into the FIFO until the packet transmission completes and this bit is cleared.

D5 DisAF_NAK_Short

When this bit is set to "0" (default setting) and the packet that was received at normal completion time of the OUT transaction is a short packet, the ForceNAK bit is automatically set to "1". When this bit is set to "1", this function is disabled.
When the AutoForceNAK bit is set to "1", the AutoForceNAK bit has a priority.

D4 ToggleStat

Shows the status of the toggle sequence bit of the endpoint EPa.

D3 ToggleSet

Set the toggle sequence bit of the endpoint EPa to "1".

D2 ToggleClr

Set the toggle sequence bit of the endpoint EPa to "0" (to be cleared).

D1 ForceNAK

If this bit is set to "1", the NAK response is done for the transaction of the endpoint EPa regardless of the FIFO data quantity and space capacity.
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

D0 ForceSTALL

If this bit is set to "1", the STALL response is done for the transaction of the endpoint EPa. This bit has a priority over the setting of the ForceNAK bit.
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

0x1000041 EPbControl (EPb Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPbControl (EPb control)	1000041 (B)	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	
		D6	EnShortPkt	1	Enable short packet	0	Do nothing	0	R/W	
		D5	DisAF_NAK_Short	1	Disable auto force	0	Auto force NAK short	0	R/W	
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	

This register sets operations of the endpoint EPb.

D7 AutoForceNAK

Sets the ForceNAK bit of this register to "1" when the transaction of the endpoint EPb completes normally.

D6 EnShortPkt

Setting this bit to "1" enables to send the data within the FIFO that is less than the quantity specified for the MaxPacketSize, as a short packet for the IN transaction of the endpoint EPb. When the IN transaction that transmitted short packets completes, this bit is automatically set to "0" (to be cleared). When a packet of the max packet size is transmitted, this bit is not cleared.
If this bit is set to "1" when the FIFO has no data, a zero-length packet can be transmitted for the IN token from the host. If the data is written into the FIFO that is in the transmission process with the packet to which this bit is set, that data may be included in transmission. Therefore, do not write into the FIFO until the packet transmission completes and this bit is cleared.

D5 DisAF_NAK_Short

When this bit is set to "0" (default setting) and the packet that was received at normal completion time of the OUT transaction is a short packet, the ForceNAK bit is automatically set to "1". When this bit is set to "1", this function is disabled.
When the AutoForceNAK bit is set to "1", the AutoForceNAK bit has a priority.

D4 ToggleStat

Shows the status of the toggle sequence bit of the endpoint EPb.

D3 ToggleSet

Set the toggle sequence bit of the endpoint EPb to "1".

D2 ToggleClr

Set the toggle sequence bit of the endpoint EPb to "0" (to be cleared).

D1 ForceNAK

If this bit is set to "1", the NAK response is done for the transaction of the endpoint EPb regardless of the FIFO data quantity and space capacity.
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

D0 ForceSTALL

If this bit is set to "1", the STALL response is done for the transaction of the endpoint EPb. This bit has a priority over the setting of the ForceNAK bit.
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

0x1000042 EPcControl (EPc Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPcControl (EPc control)	1000042 (B)	D7	**AutoForceNAK**	1	Auto force NAK	0	Do nothing	0	R/W	
		D6	**EnShortPkt**	1	Enable short packet	0	Do nothing	0	R/W	
		D5	**DisAF_NAK_Short**	1	Disable auto force	0	Auto force NAK short	0	R/W	
		D4	**ToggleStat**	Toggle sequence bit				0	R	
		D3	**ToggleSet**	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	**ToggleClr**	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	**ForceNAK**	1	Force NAK	0	Do nothing	0	R/W	
		D0	**ForceSTALL**	1	Force STALL	0	Do nothing	0	R/W	

This register sets operations of the endpoint EPc.

D7 AutoForceNAK

Sets the ForceNAK bit of this register to "1" when the transaction of the endpoint EPc completes normally.

D6 EnShortPkt

Setting this bit to "1" enables to send the data within the FIFO that is less than the quantity specified for the MaxPacketSize, as a short packet for the IN transaction of the endpoint EPc. When the IN transaction that transmitted short packets completes, this bit is automatically set to "0" (to be cleared). When a packet of the max packet size is transmitted, this bit is not cleared.
If this bit is set to "1" when the FIFO has no data, a zero-length packet can be transmitted for the IN token from the host. If the data is written into the FIFO that is in the transmission process with the packet to which this bit is set, that data may be included in transmission. Therefore, do not write into the FIFO until the packet transmission completes and this bit is cleared.

D5 DisAF_NAK_Short

When this bit is set to "0" (default setting) and the packet that was received at normal completion time of the OUT transaction is a short packet, the ForceNAK bit is automatically set to "1". When this bit is set to "1", this function is disabled.
When the AutoForceNAK bit is set to "1", the AutoForceNAK bit has a priority.

D4 ToggleStat

Shows the status of the toggle sequence bit of the endpoint EPc.

D3 ToggleSet

Set the toggle sequence bit of the endpoint EPc to "1".

D2 ToggleClr

Set the toggle sequence bit of the endpoint EPc to "0" (to be cleared).

D1 ForceNAK

If this bit is set to "1", the NAK response is done for the transaction of the endpoint EPc regardless of the FIFO data quantity and space capacity.
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

D0 ForceSTALL

If this bit is set to "1", the STALL response is done for the transaction of the endpoint EPc. This bit has a priority over the setting of the ForceNAK bit.
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

0x1000043 EPdControl (EPd Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPdControl (EPd control)	1000043 (B)	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	
		D6	EnShortPkt	1	Enable short packet	0	Do nothing	0	R/W	
		D5	DisAF_NAK_Short	1	Disable auto force	0	Auto force NAK short	0	R/W	
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	

This register sets operations of the endpoint EPd.

D7 AutoForceNAK

Sets the ForceNAK bit of this register to "1" when the transaction of the endpoint EPd completes normally.

D6 EnShortPkt

Setting this bit to "1" enables to send the data within the FIFO that is less than the quantity specified for the MaxPacketSize, as a short packet for the IN transaction of the endpoint EPd. When the IN transaction that transmitted short packets completes, this bit is automatically set to "0" (to be cleared). When a packet of the max packet size is transmitted, this bit is not cleared.
If this bit is set to "1" when the FIFO has no data, a zero-length packet can be transmitted for the IN token from the host. If the data is written into the FIFO that is in the transmission process with the packet to which this bit is set, that data may be included in transmission. Therefore, do not write into the FIFO until the packet transmission completes and this bit is cleared.

D5 DisAF_NAK_Short

When this bit is set to "0" (default setting) and the packet that was received at normal completion time of the OUT transaction is a short packet, the ForceNAK bit is automatically set to "1". When this bit is set to "1", this function is disabled.
When the AutoForceNAK bit is set to "1", the AutoForceNAK bit has a priority.

D4 ToggleStat

Shows the status of the toggle sequence bit of the endpoint EPd.

D3 ToggleSet

Set the toggle sequence bit of the endpoint EPd to "1".

D2 ToggleClr

Set the toggle sequence bit of the endpoint EPd to "0" (to be cleared).

D1 ForceNAK

If this bit is set to "1", the NAK response is done for the transaction of the endpoint EPd regardless of the FIFO data quantity and space capacity.
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

D0 ForceSTALL

If this bit is set to "1", the STALL response is done for the transaction of the endpoint EPd. This bit has a priority over the setting of the ForceNAK bit.
When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

0x1000050 EPaMaxSize_H (EPa Max Packet Size HIGH)
0x1000051 EPaMaxSize_L (EPa Max Packet Size LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPaMaxSize_H (EPa max packet size high)	1000050 (B)	D7–2	–	–	–	–	0 when being read.
		D1	EPaMaxSize[9]	Endpoint EPa max packet size	0	R/W	
		D0	EPaMaxSize[8]		0		
EPaMaxSize_L (EPa max packet size low)	1000051 (B)	D7	EPaMaxSize[7]	Endpoint EPa max packet size	0	R/W	
		D6	EPaMaxSize[6]		0		
		D5	EPaMaxSize[5]		0		
		D4	EPaMaxSize[4]		0		
		D3	EPaMaxSize[3]		0		
		D2	EPaMaxSize[2]		0		
		D1	EPaMaxSize[1]		0		
		D0	EPaMaxSize[0]		0		

EPaMaxSize[9:0]

This register sets the MaxPacketSize of the endpoint EPa.
When using this endpoint for the bulk transfer, 8, 16, 32, or 64 bytes should be set.
When using this endpoint for the interrupt transfer, up to 64 bytes can be set.
If the area of the endpoint EPa is smaller than specified here, the macro does not operate normally.

0x1000052 EPaConfig_0 (EPa Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPaConfig_0 (EPa configuration 0)	1000052 (B)	D7	INxOUT	1 In / 0 Out	0	R/W	
		D6	ToggleMode	1 Always toggle / 0 Normal toggle	0	R/W	
		D5	EnEndPoint	1 Enable endpoint / 0 Disable endpoint	0	R/W	
		D4	–	–	–	–	0 when being read.
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)	0	R/W	
		D2	EndPointNumber[2]		0		
		D1	EndPointNumber[1]		0		
		D0	EndPointNumber[0]		0		

This register sets up the endpoint EPa.
Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 INxOUT

Set the transfer direction of the endpoint.

D6 ToggleMode

Set the operation mode of the toggle sequence bit. (Only for the IN transaction)
Normal toggle - Perform the toggle only when the transaction ends normally.
Always toggle - Always perform the toggle for every transaction.

D5 EnEndPoint

Setting this bit to "1" enables this endpoint.
When this bit is "0", access to an endpoint is neglected.
Perform the setup according to the SetConfiguration request from the host.

D4 Reserved

D3–D0 EndPointNumber

Set an endpoint number between 0x1 and 0xF.

0x1000053 EPaConfig_1 (EPa Configuration 1)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPaConfig_1 (EPa configuration 1)	**1000053 (B)**	D7	**ISO**	1	ISO	0	Non-ISO	0	R/W	
		D6	**ISO_CRCmode**	1	CRC mode	0	Normal ISO	0	R/W	
		D5–0	**–**	–				–	–	0 when being read.

This register sets up the endpoint EPa.
Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 ISO
Set the isochronous mode.

D6 ISO_CRCmode
According to USB spec, a packet must be discarded when CRC error occours in isochronous transaction. When this bit is set, a packet with CRC error is not discarded. This bit is valid when ISO bit (D7) is set.

D5–D0 Reserved

0x1000054 EPbMaxSize_H (EPb Max Packet Size HIGH)
0x1000055 EPbMaxSize_L (EPb Max Packet Size LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPbMaxSize_H (EPb max packet size high)	1000054 (B)	D7–2	–	–	–	–	0 when being read.
		D1	EPbMaxSize[9]	Endpoint EPb max packet size	0	R/W	
		D0	EPbMaxSize[8]		0		
EPbMaxSize_L (EPb max packet size low)	1000055 (B)	D7	EPbMaxSize[7]	Endpoint EPb max packet size	0	R/W	
		D6	EPbMaxSize[6]		0		
		D5	EPbMaxSize[5]		0		
		D4	EPbMaxSize[4]		0		
		D3	EPbMaxSize[3]		0		
		D2	EPbMaxSize[2]		0		
		D1	EPbMaxSize[1]		0		
		D0	EPbMaxSize[0]		0		

EPbMaxSize[9:0]

This register sets the MaxPacketSize of the endpoint EPb.
When using this endpoint for the bulk transfer, 8, 16, 32, or 64 bytes should be set.
When using this endpoint for the interrupt transfer, up to 64 bytes can be set.
If the area of the endpoint EPb is smaller than specified here, the macro does not operate normally.

0x1000056 EPbConfig_0 (EPb Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPbConfig_0 (EPb configuration 0)	1000056 (B)	D7	INxOUT	1 In / 0 Out	0	R/W	
		D6	ToggleMode	1 Always toggle / 0 Normal toggle	0	R/W	
		D5	EnEndPoint	1 Enable endpoint / 0 Disable endpoint	0	R/W	
		D4	–	–	–	–	0 when being read.
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)	0	R/W	
		D2	EndPointNumber[2]		0		
		D1	EndPointNumber[1]		0		
		D0	EndPointNumber[0]		0		

This register sets up the endpoint EPb.
Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 INxOUT

Set the transfer direction of the endpoint.

D6 ToggleMode

Set the operation mode of the toggle sequence bit. (Only for the IN transaction)
Normal toggle - Perform the toggle only when the transaction ends normally.
Always toggle - Always perform the toggle for every transaction.

D5 EnEndPoint

Setting this bit to "1" enables this endpoint.
When this bit is "0", access to an endpoint is neglected.
Perform the setup according to the SetConfiguration request from the host.

D4 Reserved

D3–D0 EndPointNumber

Set an endpoint number between 0x1 and 0xF.

0x1000057 EPbConfig_1 (EPb Configuration 1)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPbConfig_1 (EPb configuration 1)	**1000057 (B)**	D7	**ISO**	1	ISO	0	Non-ISO	0	R/W	
		D6	**ISO_CRCmode**	1	CRC mode	0	Normal ISO	0	R/W	
		D5–0	**–**	–				–	–	0 when being read.

This register sets up the endpoint EPb.
Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 ISO

Set the isochronous mode.

D6 ISO_CRCmode

According to USB spec, a packet must be discarded when CRC error occours in isochronous transaction. When this bit is set, a packet with CRC error is not discarded. This bit is valid when ISO bit (D7) is set.

D5–D0 Reserved

0x1000058 EPcMaxSize_H (EPc Max Packet Size HIGH)
0x1000059 EPcMaxSize_L (EPc Max Packet Size LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPcMaxSize_H (EPc max packet size high)	1000058 (B)	D7–2	–	–	–	–	0 when being read.
		D1	EPcMaxSize[9]	Endpoint EPc max packet size	0	R/W	
		D0	EPcMaxSize[8]		0		
EPcMaxSize_L (EPc max packet size low)	1000059 (B)	D7	EPcMaxSize[7]	Endpoint EPc max packet size	0	R/W	
		D6	EPcMaxSize[6]		0		
		D5	EPcMaxSize[5]		0		
		D4	EPcMaxSize[4]		0		
		D3	EPcMaxSize[3]		0		
		D2	EPcMaxSize[2]		0		
		D1	EPcMaxSize[1]		0		
		D0	EPcMaxSize[0]		0		

EPcMaxSize[9:0]

This register sets the MaxPacketSize of the endpoint EPc.
When using this endpoint for the bulk transfer, 8, 16, 32, or 64 bytes should be set.
When using this endpoint for the interrupt transfer, up to 64 bytes can be set.
If the area of the endpoint EPc is smaller than specified here, the macro does not operate normally.

0x100005A EPcConfig_0 (EPc Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPcConfig_0 (EPc configuration 0)	100005A (B)	D7	INxOUT	1 In / 0 Out	0	R/W	
		D6	ToggleMode	1 Always toggle / 0 Normal toggle	0	R/W	
		D5	EnEndPoint	1 Enable endpoint / 0 Disable endpoint	0	R/W	
		D4	–	–	–	–	0 when being read.
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)	0	R/W	
		D2	EndPointNumber[2]		0		
		D1	EndPointNumber[1]		0		
		D0	EndPointNumber[0]		0		

This register sets up the endpoint EPc.
Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 INxOUT

Set the transfer direction of the endpoint.

D6 ToggleMode

Set the operation mode of the toggle sequence bit. (Only for the IN transaction)
Normal toggle - Perform the toggle only when the transaction ends normally.
Always toggle - Always perform the toggle for every transaction.

D5 EnEndPoint

Setting this bit to "1" enables this endpoint.
When this bit is "0", access to an endpoint is neglected.
Perform the setup according to the SetConfiguration request from the host.

D4 Reserved

D3–D0 EndPointNumber

Set an endpoint number between 0x1 and 0xF.

0x100005B EPcConfig_1 (EPc Configuration 1)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPcConfig_1 (EPc configuration 1)	**100005B (B)**	D7	**ISO**	1	ISO	0	Non-ISO	0	R/W	
		D6	**ISO_CRCmode**	1	CRC mode	0	Normal ISO	0	R/W	
		D5–0	**–**	–				–	–	0 when being read.

This register sets up the endpoint EPc.
Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 ISO

Set the isochronous mode.

D6 ISO_CRCmode

According to USB spec, a packet must be discarded when CRC error occours in isochronous transaction. When this bit is set, a packet with CRC error is not discarded. This bit is valid when ISO bit (D7) is set.

D5–D0 Reserved

0x100005C EPdMaxSize_H (EPd Max Packet Size HIGH)
0x100005D EPdMaxSize_L (EPd Max Packet Size LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPdMaxSize_H (EPd max packet size high)	100005C (B)	D7–2	–	–	–	–	0 when being read.
		D1	EPdMaxSize[9]	Endpoint EPd max packet size	0	R/W	
		D0	EPdMaxSize[8]		0		
EPdMaxSize_L (EPd max packet size low)	100005D (B)	D7	EPdMaxSize[7]	Endpoint EPd max packet size	0	R/W	
		D6	EPdMaxSize[6]		0		
		D5	EPdMaxSize[5]		0		
		D4	EPdMaxSize[4]		0		
		D3	EPdMaxSize[3]		0		
		D2	EPdMaxSize[2]		0		
		D1	EPdMaxSize[1]		0		
		D0	EPdMaxSize[0]		0		

EPdMaxSize[9:0]

This register sets the MaxPacketSize of the endpoint EPd.
When using this endpoint for the bulk transfer, 8, 16, 32, or 64 bytes should be set.
When using this endpoint for the interrupt transfer, up to 64 bytes can be set.
If the area of the endpoint EPd is smaller than specified here, the macro does not operate normally.

0x100005E EPdConfig_0 (EPd Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPdConfig_0 (EPd configuration 0)	100005E (B)	D7	INxOUT	1 In / 0 Out	0	R/W	
		D6	ToggleMode	1 Always toggle / 0 Normal toggle	0	R/W	
		D5	EnEndPoint	1 Enable endpoint / 0 Disable endpoint	0	R/W	
		D4	–	–	–	–	0 when being read.
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)	0	R/W	
		D2	EndPointNumber[2]		0		
		D1	EndPointNumber[1]		0		
		D0	EndPointNumber[0]		0		

This register sets up the endpoint EPd.
Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 INxOUT

Set the transfer direction of the endpoint.

D6 ToggleMode

Set the operation mode of the toggle sequence bit. (Only for the IN transaction)
Normal toggle - Perform the toggle only when the transaction ends normally.
Always toggle - Always perform the toggle for every transaction.

D5 EnEndPoint

Setting this bit to "1" enables this endpoint.
When this bit is "0", access to an endpoint is neglected.
Perform the setup according to the SetConfiguration request from the host.

D4 Reserved

D3–D0 EndPointNumber

Set an endpoint number between 0x1 and 0xF.

0x100005F EPdConfig_1 (EPd Configuration 1)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPdConfig_1 (EPd configuration 1)	100005F (B)	D7	ISO	1	ISO	0	Non-ISO	0	R/W	
		D6	ISO_CRCmode	1	CRC mode	0	Normal ISO	0	R/W	
		D5–0	–	–				–	–	0 when being read.

This register sets up the endpoint EPd.
Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 ISO

Set the isochronous mode.

D6 ISO_CRCmode

According to USB spec, a packet must be discarded when CRC error occours in isochronous transaction. When this bit is set, a packet with CRC error is not discarded. This bit is valid when ISO bit (D7) is set.

D5–D0 Reserved

0x1000070 EPaStartAdrs_H (EPa FIFO Start Address HIGH)
0x1000071 EPaStartAdrs_L (EPa FIFO Start Address LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPaStartAdrs_H (EPa FIFO start address high)	1000070 (B)	D7–4	–	–	–	–	0 when being read.
		D3	EPaStartAdrs[11]	Endpoint EPa start address	0	R/W	
		D2	EPaStartAdrs[10]		0		
		D1	EPaStartAdrs[9]		0		
		D0	EPaStartAdrs[8]		0		
EPaStartAdrs_L (EPa FIFO start address low)	1000071 (B)	D7	EPaStartAdrs[7]	Endpoint EPa start address	0	R/W	
		D6	EPaStartAdrs[6]		0		
		D5	EPaStartAdrs[5]		0		
		D4	EPaStartAdrs[4]		0		
		D3	EPaStartAdrs[3]		0		
		D2	EPaStartAdrs[2]		0		
		D1–0	–	–	–	–	0 when being read.

EPaStartAdrs[11:2]

Sets the start address of the FIFO area allocated to the endpoint EPa.
The area that is allocated to the endpoint EPa is from the address set by the EPaStartAdrs and to the address one byte before the one set by the EPbStartAdrs.
After setting the StartAdrs of all endpoints, be sure to set the AllFIFO_Clr bit of the EPnControl register to "1" to clear all FIFOs.
If the EPaMaxSize of the endpoint EPa is larger than the area specified in here, the macro does not operate normally.
Set the total of the FIFO area secured for all endpoints does not exceed the total capacity of the built-in RAM.
Allocate the FIFO area to the endpoints in the order from the lower order address to higher order address like EP0, EPa, EPb, EPc, EPd.
The FIFO of the endpoint EP0 is allocated from the address 0 to up to the size specified as the MaxPacketSize of the endpoint EP0 set in the EP0MaxSize register. Allocate the succeeding area for other endpoints.
Since the FIFO capacity is 1K bytes, do not let the EPd end address exceed 0x3FF. And do not let the EPaStartAdrs exceed the setting value of the EPbStartAdrs.

0x1000072 EPbStartAdrs_H (EPb FIFO Start Address HIGH)
0x1000073 EPbStartAdrs_L (EPb FIFO Start Address LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPbStartAdrs_H (EPb FIFO start address high)	**1000072 (B)**	D7–4	**–**	–	–	–	0 when being read.
		D3	**EPbStartAdrs[11]**	Endpoint EPb start address	0	R/W	
		D2	**EPbStartAdrs[10]**		0		
		D1	**EPbStartAdrs[9]**		0		
		D0	**EPbStartAdrs[8]**		0		
EPbStartAdrs_L (EPb FIFO start address low)	**1000073 (B)**	D7	**EPbStartAdrs[7]**	Endpoint EPb start address	0	R/W	
		D6	**EPbStartAdrs[6]**		0		
		D5	**EPbStartAdrs[5]**		0		
		D4	**EPbStartAdrs[4]**		0		
		D3	**EPbStartAdrs[3]**		0		
		D2	**EPbStartAdrs[2]**		0		
		D1–0	**–**	–	–	–	0 when being read.

EPbStartAdrs[11:2]

Sets the start address of the FIFO area allocated to the endpoint EPb.
The area that is allocated to the endpoint EPb is from the address set by the EPbStartAdrs and to the address one byte before the one set by the EPcStartAdrs.
After setting the StartAdrs of all endpoints, be sure to set the AllFIFO_Clr bit of the EPnControl register to "1" to clear all FIFOs.
If the EPbMaxSize of the endpoint EPb is larger than the area specified in here, the macro does not operate normally.
Set the total of the FIFO area secured for all endpoints does not exceed the total capacity of the built-in RAM.
Allocate the FIFO area to the endpoints in the order from the lower order address to higher order address like EP0, EPa, EPb, EPc, EPd.
The FIFO of the endpoint EP0 is allocated from the address 0 to up to the size specified as the MaxPacketSize of the endpoint EP0 set in the EP0MaxSize register. Allocate the succeeding area for other endpoints.
Since the FIFO capacity is 1K bytes, do not let the EPd end address exceed 0x3FF. And do not let the EPbStartAdrs exceed the setting value of the EPcStartAdrs.

0x1000074 EPcStartAdrs_H (EPc FIFO Start Address HIGH)
0x1000075 EPcStartAdrs_L (EPc FIFO Start Address LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPcStartAdrs_H (EPc FIFO start address high)	**1000074 (B)**	D7–4	–	–	–	–	0 when being read.
		D3	**EPcStartAdrs[11]**	Endpoint EPc start address	0	R/W	
		D2	**EPcStartAdrs[10]**		0		
		D1	**EPcStartAdrs[9]**		0		
		D0	**EPcStartAdrs[8]**		0		
EPcStartAdrs_L (EPc FIFO start address low)	**1000075 (B)**	D7	**EPcStartAdrs[7]**	Endpoint EPc start address	0	R/W	
		D6	**EPcStartAdrs[6]**		0		
		D5	**EPcStartAdrs[5]**		0		
		D4	**EPcStartAdrs[4]**		0		
		D3	**EPcStartAdrs[3]**		0		
		D2	**EPcStartAdrs[2]**		0		
		D1–0	–	–	–	–	0 when being read.

EPcStartAdrs[11:2]

Sets the start address of the FIFO area allocated to the endpoint EPc.

The area that is allocated to the endpoint EPc is from the address set by the EPcStartAdrs and to the address one byte before the one set by the EPdStartAdrs.

After setting the StartAdrs of all endpoints, be sure to set the AllFIFO_Clr bit of the EPnControl register to "1" to clear all FIFOs.

If the EPcMaxSize of the endpoint EPc is larger than the area specified in here, the macro does not operate normally.

Set the total of the FIFO area secured for all endpoints does not exceed the total capacity of the built-in RAM.

Allocate the FIFO area to the endpoints in the order from the lower order address to higher order address like EP0, EPa, EPb, EPc, EPd.

The FIFO of the endpoint EP0 is allocated from the address 0 to up to the size specified as the MaxPacketSize of the endpoint EP0 set in the EP0MaxSize register. Allocate the succeeding area for other endpoints.

Since the FIFO capacity is 1K bytes, do not let the EPd end address exceed 0x3FF. And do not let the EPcStartAdrs exceed the setting value of the EPdStartAdrs.

0x1000076 EPdStartAdrs_H (EPd FIFO Start Address HIGH)
0x1000077 EPdStartAdrs_L (EPd FIFO Start Address LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPdStartAdrs_H (EPd FIFO start address high)	1000076 (B)	D7–4	–	–	–	–	0 when being read.
		D3	EPdStartAdrs[11]	Endpoint EPd start address	0	R/W	
		D2	EPdStartAdrs[10]		0		
		D1	EPdStartAdrs[9]		0		
		D0	EPdStartAdrs[8]		0		
EPdStartAdrs_L (EPd FIFO start address low)	1000077 (B)	D7	EPdStartAdrs[7]	Endpoint EPd start address	0	R/W	
		D6	EPdStartAdrs[6]		0		
		D5	EPdStartAdrs[5]		0		
		D4	EPdStartAdrs[4]		0		
		D3	EPdStartAdrs[3]		0		
		D2	EPdStartAdrs[2]		0		
		D1–0	–	–	–	–	0 when being read.

EPdStartAdrs[11:2]

Sets the start address of the FIFO area allocated to the endpoint EPd.

The area that is allocated to the endpoint EPd is from the address set by the EPdStartAdrs and to the end address of the FIFO.

After setting the StartAdrs of all endpoints, be sure to set the AllFIFO_Clr bit of the EPnControl register to "1" to clear all FIFOs.

If the EPdMaxSize of the endpoint EPd is larger than the area specified in here, the macro does not operate normally.

Set the total of the FIFO area secured for all endpoints does not exceed the total capacity of the built-in RAM.

Allocate the FIFO area to the endpoints in the order from the lower order address to higher order address like EP0, EPa, EPb, EPc, EPd.

The FIFO of the endpoint EP0 is allocated from the address 0 to up to the size specified as the MaxPacketSize of the endpoint EP0 set in the EP0MaxSize register. Allocate the succeeding area for other endpoints.

Since the FIFO capacity is 1K bytes, do not let the EPd end address exceed 0x3FF.

0x1000080 CPU_JoinRd (CPU Join FIFO Read)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
CPU_JoinRd (CPU join FIFO read)	1000080 (B)	D7–4	–	–				–	–	0 when being read.
		D3	**JoinEPdRd**	1	Join EPd FIFO read	0	Do nothing	0	R/W	
		D2	**JoinEPcRd**	1	Join EPc FIFO read	0	Do nothing	0	R/W	
		D1	**JoinEPbRd**	1	Join EPb FIFO read	0	Do nothing	0	R/W	
		D0	**JoinEPaRd**	1	Join EPa FIFO read	0	Do nothing	0	R/W	

This register can be set up to read the FIFO data of the endpoint through the CPU Interface. When the EPnFIFOforCPU register is read after the setup of this register is completed, the FIFO data of the relevant endpoint can be read. The remained data quantity of the FIFO can be referred by the EPnRdRemain_H, L register.
This register can set only one bit to "1" at the same time. When "1" is written into multiple bits at the same time, writing in higher order bit is regarded as valid. When all bits are set to "0", EP0 will be joined.
The reading data from CPU I/F through the endpoint used by USB I/F or DMA I/F is not allowed.
If CPU I/F needs to read from the IN direction endpoint, use the ForceNAK bit to avoid reading data from USB I/F.
If CPU I/F needs to read from the OUT direction endpoint, check the DMA_Running bit of the DMA_Control register to avoid reading data from DMA I/F at the same time.
This register is valid when EnEPnFIFO_Access.EnEPnFIFO_Rd bit is set.

D7–D4 Reserved

D3 JoinEPdRd

If this bit is set to "1", the FIFO data of the endpoint EPd can be read from the EPnFIFOforCPU register. In addition, reference to the data quantity in the FIFO of the endpoint EPd by the EPnRdRemain_H, L register is enabled.

D2 JoinEPcRd

If this bit is set to "1", the FIFO data of the endpoint EPc can be read from the EPnFIFOforCPU register. In addition, reference to the data quantity in the FIFO of the endpoint EPc by the EPnRdRemain_H, L register is enabled.

D1 JoinEPbRd

If this bit is set to "1", the FIFO data of the endpoint EPb can be read from the EPnFIFOforCPU register. In addition, reference to the data quantity in the FIFO of the endpoint EPb by the EPnRdRemain_H, L register is enabled.

D0 JoinEPaRd

If this bit is set to "1", the FIFO data of the endpoint EPa can be read from the EPnFIFOforCPU register. In addition, reference to the data quantity in the FIFO of the endpoint EPa by the EPnRdRemain_H, L register is enabled.

0x1000081 CPU_JoinWr (CPU Join FIFO Write)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
CPU_JoinWr (CPU join FIFO write)	1000081 (B)	D7–4	–	–				–	–	0 when being read.
		D3	JoinEPdWr	1	Join EPd FIFO write	0	Do nothing	0	R/W	
		D2	JoinEPcWr	1	Join EPc FIFO write	0	Do nothing	0	R/W	
		D1	JoinEPbWr	1	Join EPb FIFO write	0	Do nothing	0	R/W	
		D0	JoinEPaWr	1	Join EPa FIFO write	0	Do nothing	0	R/W	

This register can be set up to write the FIFO data of the endpoint through the CPU Interface. When the EPnFIFOforCPU register is written after the setup of this register is completed, the FIFO data of the relevant endpoint can be read. The space capacity of the FIFO can be referred by the EPnWrRemain_H, L register.
This register can set only one bit to "1" at the same time. When "1" is written into multiple bits at the same time, writing in higher order bit is regarded as valid. When all bits are set to "0", EP0 will be joined.
The reading data from CPU I/F through the endpoint used by USB I/F or DMA I/F is not allowed.
If CPU I/F needs to write to the OUT direction endpoint, use the ForceNAK bit to avoid writing data from USB I/F.
If CPU I/F needs to write to the IN direction endpoint, check the DMA_Running bit of the DMA_Control register to avoid writing data from DMA I/F at the same time.
This register is valid when EnEPnFIFO_Access.EnEPnFIFO_Wr bit is set.

D7–D4 Reserved

D3 JoinEPdWr
If this bit is set to "1", the FIFO data of the endpoint EPd can be written into the EPnFIFOforCPU register. In addition, reference to the space capacity in the FIFO of the endpoint EPd by the EPnWrRemain_H, L register is enabled.

D2 JoinEPcWr
If this bit is set to "1", the FIFO data of the endpoint EPc can be written into the EPnFIFOforCPU register. In addition, reference to the space capacity in the FIFO of the endpoint EPc by the EPnWrRemain_H, L register is enabled.

D1 JoinEPbWr
If this bit is set to "1", the FIFO data of the endpoint EPb can be written into the EPnFIFOforCPU register. In addition, reference to the space capacity in the FIFO of the endpoint EPb by the EPnWrRemain_H, L register is enabled.

D0 JoinEPaWr
If this bit is set to "1", the FIFO data of the endpoint EPa can be written into the EPnFIFOforCPU register. In addition, reference to the space capacity in the FIFO of the endpoint EPa by the EPnWrRemain_H, L register is enabled.

0x1000082 EnEPnFIFO_Access (Enable EPn FIFO Access)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EnEPnFIFO_Access (Enable EPn FIFO access)	1000082 (B)	D7–2	–	–				–	–	0 when being read.
		D1	EnEPnFIFO_Wr	1	Enable join EPn FIFO write	0	Do nothing	0	R/W	
		D0	EnEPnFIFO_Rd	1	Enable join EPn FIFO read	0	Do nothing	0	R/W	

This register enables the CPU_JoinRd and CPU_JoinWr registers so that the CPU can access the EPn FIFO.

D7–D2 Reserved

D1 EnEPnFIFO_Wr

If this bit is set to "1", the CPU_JoinWr register is enabled and the CPU can write data to the EPn FIFO selected by the CPU_JoinWr register.

D0 EnEPnFIFO_Rd

If this bit is set to "1", the CPU_JoinRd register is enabled and the CPU can read data from the EPn FIFO selected by the CPU_JoinRd register.

0x1000083 EPnFIFOforCPU (EPn FIFO for CPU)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnFIFOforCPU (EPn FIFO for CPU)	**1000083 (B)**	D7	**EPnFIFOData[7]**	Endpoint EP0 FIFO access from CPU	X	R/W	
		D6	**EPnFIFOData[6]**		X		
		D5	**EPnFIFOData[5]**		X		
		D4	**EPnFIFOData[4]**		X		
		D3	**EPnFIFOData[3]**		X		
		D2	**EPnFIFOData[2]**		X		
		D1	**EPnFIFOData[1]**		X		
		D0	**EPnFIFOData[0]**		X		

This register is used for accessing the FIFO of the endpoint from the CPU Interface.
When a bit of the CPU_JoinRd register is set to "1", the data can be read from the FIFO by reading values from this register.
When a bit of the CPU_JoinWr register is set to "1", the data can be written into the FIFO by writing values into this register.
If values are read from this register without setting the EnEPnFIFO_Rd bit of the EnEPnFIFO_Access register, a dummy data will be output.
If writing is done into this register without setting the EnEPnFIFO_Wr bit of the EnEPnFIFO_Access register, writing into the FIFO is not done.

If this register is read when the FIFO of the relevant endpoint is empty, a dummy data will be read.
If writing is done into this register when the FIFO of the relevant endpoint has no space, writing into the FIFO is not done.

0x1000084 EPnRdRemain_H (EPn FIFO Read Remain HIGH)
0x1000085 EPnRdRemain_L (EPn FIFO Read Remain LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnRdRemain_H	1000084	D7–4	–	–	–	–	0 when being read.
(EPn FIFO read	(B)	D3	EPnRdRemain[11]	Endpoint n FIFO read remain high	0	R	
remain high)		D2	EPnRdRemain[10]		0		
		D1	EPnRdRemain[9]		0		
		D0	EPnRdRemain[8]		0		
EPnRdRemain_L	1000085	D7	EPnRdRemain[7]	Endpoint n FIFO read remain low	0	R	
(EPn FIFO read	(B)	D6	EPnRdRemain[6]		0		
remain low)		D5	EPnRdRemain[5]		0		
		D4	EPnRdRemain[4]		0		
		D3	EPnRdRemain[3]		0		
		D2	EPnRdRemain[2]		0		
		D1	EPnRdRemain[1]		0		
		D0	EPnRdRemain[0]		0		

This register shows the remained data quantity in the FIFO of the endpoint connected to the CPU Interface by the CPU_JoinRd register. When the remained data quantity in the FIFO is acquired, the EPnRdRemain_H and the EPnRdRemain_L registers must be accessed as a pair. When accessing them, access the EPnRdRemain_H register first.

0x1000086 EPnWrRemain_H (EPn FIFO Write Remain HIGH)
0x1000087 EPnWrRemain_L (EPn FIFO Write Remain LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnWrRemain_H	1000086	D7–4	–	–	–	–	0 when being read.
(EPn FIFO	(B)	D3	EPnWrRemain[11]	Endpoint n FIFO write remain high	0	R	
write remain		D2	EPnWrRemain[10]		0		
high)		D1	EPnWrRemain[9]		0		
		D0	EPnWrRemain[8]		0		
EPnWrRemain_L	1000087	D7	EPnWrRemain[7]	Endpoint n FIFO write remain low	0	R	
(EPn FIFO	(B)	D6	EPnWrRemain[6]		0		
write remain		D5	EPnWrRemain[5]		0		
low)		D4	EPnWrRemain[4]		0		
		D3	EPnWrRemain[3]		0		
		D2	EPnWrRemain[2]		0		
		D1	EPnWrRemain[1]		0		
		D0	EPnWrRemain[0]		0		

This register shows the space capacity in the FIFO of the endpoint connected to the CPU Interface by the CPU_JoinWr register. When the space capacity in the FIFO is acquired, the EPnWrRemain_H and the EPnWrRemain_L registers must be accessed as a pair. When accessing them, access the EPnWrRemain_H register first.

0x1000088 DescAdrs_H (Descriptor Address HIGH)
0x1000089 DescAdrs_L (Descriptor Address LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescAdrs_H	**1000088**	D7–4	**–**	–	–	–	0 when being read.
(Descriptor	**(B)**	D3	**DescAdrs[11]**	Descriptor address	0	R/W	
address high)		D2	**DescAdrs[10]**		0		
		D1	**DescAdrs[9]**		0		
		D0	**DescAdrs[8]**		0		
DescAdr_L	**1000089**	D7	**DescAdrs[7]**	Descriptor address	0	R/W	
(Descriptor	**(B)**	D6	**DescAdrs[6]**		0		
address low)		D5	**DescAdrs[5]**		0		
		D4	**DescAdrs[4]**		0		
		D3	**DescAdrs[3]**		0		
		D2	**DescAdrs[2]**		0		
		D1	**DescAdrs[1]**		0		
		D0	**DescAdrs[0]**		0		

DescAdrs[11:0]

Specify the start address of the FIFO used at the start of Descriptor reply operation, Descriptor write operation and Descriptor read operation in the Descriptor reply function.
The Descriptor Address does not have the function to allocate the FIFO area for the Descriptor reply function. The entire FIFO area ranging from 0x0000 to 0x03FF (1K bytes) can be specified for the Descriptor Address, regardless of the FIFO area setting.
In the Description reply, DescAdrs is updated every time the IN transaction completes at the endpoint EP0, as many times as the number of data transmitted. Refer to the item on the ReplyDescriptor of the EP0 Control register, for the Descriptor reply function.
Every time data is written into or read from the Descriptor, the DescAdrs is incremented by 1. Refer to the item on the DescDoor register, for the Descriptor write and read functions.
The FIFO area for the Descriptor reply function is not allocated explicitly. Therefore, specify the DescAdrs_H, L register and the DescSize_H, L register to avoid overlapping with FIFOs of other endpoints. Appropriate area is the area ranging from the end address of the area reserved by the endpoint EP0 (0x0040) to the start address of the endpoint EPa (EPaStartAdrs_H, L).

When referring to the Descriptor Address, read from the DescAdrs_H to the DescAdrs_L.

0x100008A DescSize_H (Descriptor Size High)
0x100008B DescSize_L (Descriptor Size Low)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescSize_H	100008A	D7–2	–	–	–	–	0 when being read.
(Descriptor	(B)	D1	DescSize[9]	Descriptor size	0	R/W	
size high)		D0	DescSize[8]		0		
DescSize_L	100008B	D7	DescSize[7]	Descriptor size	0	R/W	
(Descriptor	(B)	D6	DescSize[6]		0		
size low)		D5	DescSize[5]		0		
		D4	DescSize[4]		0		
		D3	DescSize[3]		0		
		D2	DescSize[2]		0		
		D1	DescSize[1]		0		
		D0	DescSize[0]		0		

DescSize[9:0]

Specify the total number of the data to reply in Descriptor reply function, for the Descriptor Size. Refer to the item on the ReplyDescriptor bit of the EP0 Control register, for the Descriptor reply function.
The area ranging from 0x0000 to 0x03FF can be specified for the Descriptor Size regardless of the FIFO area setting. In the Description reply, DescAdrs is updated every time the IN transaction completes at the endpoint EP0, as many times as the number of data transmitted.
The FIFO area for the Descriptor reply function is not allocated explicitly. Therefore, specify the DescAdrs_H, L register and the DescSize_H, L register to avoid overlapping with FIFOs of other endpoints. Use the area ranging from the end address of the area reserved by the endpoint EP0 (0x0040) to the start address of the endpoint EPa (EPaStartAdrs_H, L).

When referring to the Descriptor Size, read from the DescSize_H to the DescSize_L.

0x100008F DescDoor (Descriptor Door)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescDoor (Descriptor door)	**100008F (B)**	D7	**DescMode[7]**	Descriptor door	0	R/W	
		D6	**DescMode[6]**		0		
		D5	**DescMode[5]**		0		
		D4	**DescMode[4]**		0		
		D3	**DescMode[3]**		0		
		D2	**DescMode[2]**		0		
		D1	**DescMode[1]**		0		
		D0	**DescMode[0]**		0		

This register is the access register that is used for read and write for the Descriptor.
Before starting the write operation, set the start address of the area where the FIFO Descriptor is written, into the DescAdrs_H, L register. And then performing writing one byte by one byte into this register automatically increments the DescAdrs_H, L register one byte by one byte to write data sequentially.
The data written by the DescDoor register can be used by the ReplyDescriptor function repeatedly. Thus the Descriptor reply function protects these data from deletion and overwriting. However, if the area where the Descriptor data is written into, is overlapped with the area secured by other endpoints, the data will be overwritten.
Reading this register allows the FIFO data being read from the address specified in the DescAdrs_H, L register, sequentially. At this time, the address of the DescAdrs_H, L register is also incremented every time when the data is read. Therefore, note that even if you write and read the DescDoor register, the values written just before reading cannot be read.

0x1000090 DMA_FIFO_Control (DMA FIFO Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
DMA_FIFO_Control (DMA FIFO control)	1000090 (B)	D7	FIFO_Running	1	FIFO is running	0	FIFO is not running	0	R	
		D6	AutoEnShort	1	Auto enable short packet	0	Do nothing	0	R/W	
		D5–0	–	–				–	–	0 when being read.

D7 FIFO_Running

Shows that the FIFO of the endpoint connected to the DMA is operating. If the DMA is started, this bit is set to "1". After completing the DMA operation, this bit is set to "0" (to be cleared) when the FIFO becomes empty.

D6 AutoEnShort

When the DMA operation ends and the data smaller than the MaxPacketSize remains in the FIFO, the EnShortPkt bit of that endpoint is set to "1".
This function is valid when the direction of the endpoint connected to the DMA is the IN direction.

D5–D0 Reserved

0x1000091 DMA_Join (DMA Join FIFO)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
DMA_Join (DMA join FIFO)	1000091 (B)	D7–4	–	–				–	–	0 when being read.
		D3	JoinEPdDMA	1	Join EPd to DMA	0	Do nothing	0	R/W	
		D2	JoinEPcDMA	1	Join EPc to DMA	0	Do nothing	0	R/W	
		D1	JoinEPbDMA	1	Join EPb to DMA	0	Do nothing	0	R/W	
		D0	JoinEPaDMA	1	Join EPa to DMA	0	Do nothing	0	R/W	

The endpoint to perform the DMA transfer can be specified by setting the JoinEPd–aDMA bits. After setting these bits, the remained data quantity for the endpoint of the OUT direction or the space capacity for endpoint of the IN direction can be referred by the DMA_Remain_H, L register.
This register can set only one bit to "1" at the same time. When "1" is written into multiple bits at the same time, writing in higher order bit is regarded as valid.

D7–D4 Reserved

D3–D0 JoinEPdDMA, JoinEPcDMA, JoinEPbDMA, JoinEPaDMA

When this bit is set to "1", the DMA transfer is enabled through the endpoint EPx (x=a,b,c,d). In addition, reference to the space capacity (for the IN direction) or the data quantity (for the OUT direction) in the FIFO of the endpoint EPx (x=a,b,c,d) by the DMA_Remain H, L register, is enabled.

0x1000092 DMA_Control (DMA Control)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
DMA_Control (DMA control)	1000092 (B)	D7	DMA_Running	1	DMA is running	0	DMA is not running	0	R	
		D6	PDREQ	PDREQ (#DMAREQ3) signal logic				0	R	
		D5	PDACK	PDACK (#CE4) signal logic				0	R	
		D4	–	–				–	–	0 when being read.
		D3	CounterClr	1	Clear DMA counter	0	Do nothing	0	W	
		D2	–	–				–	–	
		D1	DMA_Stop	1	Finish DMA	0	Do nothing	0	W	
		D0	DMA_Go	1	Start DMA	0	Do nothing	0	W	

This register controls the DMA transfer and shows the status of the interface.

D7 DMA_Running

This bit is automatically set "1" during the DMA transfer. The DMA_Join register cannot be written when this bit is "1".

D6 PDREQ

Shows the logic level of the PDREQ (#DMAREQ3) signal for monitoring.

D5 PDACK

Shows the logic level of the PDACK (#CE4) signal for monitoring.

D4 Reserved

D3 CounterClr

When this bit is set to "1", the DMA_Count_HH, HL, LH and LL registers are set to 0x00 (to be cleared). When the DMA_Running bit is "1", writing into this bit is neglected.

D2 Reserved

D1 DMA_Stop

Setting this bit to "1" ends the DMA transfer. When the DMA transfer stops, the DMA_Running bit is set to "0" (to be cleared) and the DMA_Cmp bit of the MainIntStat register is set to "1". When restarting the DMA transfer, check the DMA_Running bit or the DMA_Cmp bit, and wait until the DMA operation ends.

If this bit is set and the DMA transfer starts during the asynchronous DMA transfer, the data defect may occur. In such a case, stop the operation on the master side first, and then set this bit.

D0 DMA_Go

Setting this bit to "1" starts the DMA transfer.

0x1000094 DMA_Config_0 (DMA Configuration 0)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
DMA_Config_0 (DMA configuration 0)	1000094 (B)	D7	ActivePort	1	Activate DMA port	0	Disactivate DMA port	0	R/W	
		D6–4	–	–				–	–	0 when being read.
		D3	PDREQ_Level	1	Active-low	0	Active-high	0	R/W	
		D2	PDACK_Level	1	Active-low	0	Active-high	0	R/W	
		D1	PDRDWR_Level	1	Active-low	0	Active-high	0	R/W	
		D0	–	–				–	–	0 when being read.

This register sets fields on the bus of the DMA interface.

D7 ActivePort
Set the DMA interface to "active".
When this bit is set to "0", the DMA interface signals become "Hi-Z/Don't care" state.

D6–D4 Reserved

D3 PDREQ_Level
Set the PDREQ (#DMAREQ3) logic level. Set to "0" (active-high).

D2 PDACK_Level
Set the PDACK (#CE4) logic level. Set to "1" (active-low).

D1 PDRDWR_Level
Set the logic levels of the PDRD (#RD) and PDWR (#WRL) signals. Set to "1" (active-low).

D0 Reserved

0x1000095 DMA_Config_1 (DMA Configuration 1)

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
DMA_Config_1 (DMA configuration 1)	1000095 (B)	D7	RcvLimitMode	1	Recive limit mode	0	Normal	0	R/W	
		D6–4	–	–				–	–	0 when being read.
		D3	SingleWord	1	Single word	0	Multi word	0	R/W	
		D2–1	–	–				–	–	0 when being read.
		D0	CountMode	1	Count-down mode	0	Free-run mode	0	R/W	

This register sets fields on the operation mode of the DMA interface.

D7 RcvLimitMode

Setting this bit to "1" realizes the RcvLimit mode. This function is available only during write operation for the asynchronous multi-word DMA transfer, and not available in the count down mode.
During the asynchronous DMA write operation in the RcvLimit mode, data up to 16 bytes can be received even after this macro negates the PDREQ (#DMAREQ3) signal.
In this mode, the PDREQ (#DMAREQ3) signal is negated when the space of the endpoint becomes less than 32 bytes by the DMA write operation. However, when the PDREQ (#DMAREQ3) signal is negated, 16-byte data that are not written into the endpoint may exist in the internal circuit. Therefore, the data that can be received after the PDREQ (#DMAREQ3) signal is negated, is 16 bytes or less.
In this mode, the PDREQ (#DMAREQ3) signal is negated before the endpoint becomes completely full. When the area of the endpoint set by the EP{a,b,c,d}StartAdress registers is the same as the value set by the EP{a,b,c,d}MaxSize register (Single Buffer), the endpoint never becomes full. Therefore, the data cannot be transmitted by the IN transfer of the USB.
To avoid this limitation, when using the RcvLimit mode, be sure to enter the value of the EP{a,b,c,d}MaxSize register + 32-byte or larger area, into the EP{a,b,c,d}StartAdress register.

D6–D4 Reserved

D3 SingleWord

Sets the handshake mode in the Asynchronous (handshake) mode.
In the Single Word mode, the PDREQ (#DMAREQ3) signal is negated every time when one word is transferred.
In the Multi-Word mode, the PDREQ (#DMAREQ3) signal is not negated if the next data communication is possible when one word is transferred.

D2–D1 Reserved

D0 CountMode

Sets the mode to control the number of the DMA transmissions.
In the free-run mode, the DMA transfer operation is continued until the DMA_Stop is enabled. The Transfer Byte Counter (DMA_Count_HH, HL, LH, LL) shows the number of transmissions for reference.
In the Count-down mode, the DMA transfer is continued up to the number of bytes set in the Transfer Byte Counter (DMA_Count_HH, HL, LH, LL) or until the DMA_Stop is enabled to stop it. The Transfer Byte Counter shows the remained transmission quantity, for reference.

0x1000097 DMA_Latency (DMA Latency)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Latency (DMA latency)	1000097 (B)	D7–4	–	–	–	–	0 when being read.
		D3	DMA_Latency[3]	Latency	0	R/W	
		D2	DMA_Latency[2]		0		
		D1	DMA_Latency[1]		0		
		D0	DMA_Latency[0]		0		

This register sets the Data transfer latency for the transfer in the Asynchronous (handshake) mode.
The unit time of the latency is approximately 130 ns.

D7–D4 Reserved

D3–D0 DMA_Latency[3:0]

If a value between 0x1 and 0xF is written, the PDREQ (#DMAREQ3) signal is negated every time when the 4-word is transmitted either in the Single Word mode or in the Multi-Word mode, and the PDREQ (#DMAREQ3) signal is not be asserted for (130 × N) ns period.

0x1000098 DMA_Remain_H (DMA FIFO Remain HIGH)
0x1000099 DMA_Remain_L (DMA FIFO Remain LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Remain_H	1000098	D7–4	–	–	–	–	0 when being read.
(DMA FIFO	(B)	D3	DMA_Remain[11]	DMA FIFO remain high	0	R	
remain high)		D2	DMA_Remain[10]		0		
		D1	DMA_Remain[9]		0		
		D0	DMA_Remain[8]		0		
DMA_Remain_L	1000099	D7	DMA_Remain[7]	DMA FIFO remain low	0	R	
(DMA FIFO	(B)	D6	DMA_Remain[6]		0		
remain low)		D5	DMA_Remain[5]		0		
		D4	DMA_Remain[4]		0		
		D3	DMA_Remain[3]		0		
		D2	DMA_Remain[2]		0		
		D1	DMA_Remain[1]		0		
		D0	DMA_Remain[0]		0		

When the direction of the endpoint connected to the DMA by the DMA_Join register is the OUT direction, this register shows the remained data quantity in the FIFO of the endpoint.
When the direction of the endpoint connected to the DMA by the DMA_Join register is the IN direction, this register shows the space capacity in the FIFO of the endpoint.
The DMA_Remain_H register and the DMA_Remain_L register must be accessed as a pair. When accessing them, access the DMA_Remain_H register first.

0x100009C DMA_Count_HH (DMA Transfer Byte Counter HIGH/HIGH)
0x100009D DMA_Count_HL (DMA Transfer Byte Counter HIGH/LOW)
0x100009E DMA_Count_LH (DMA Transfer Byte Counter LOW/HIGH)
0x100009F DMA_Count_LL (DMA Transfer Byte Counter LOW/LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Count_HH (DMA transfer byte counter high/high)	100009C (B)	D7	DMA_Count[31]	DMA transfer byte counter	0	R/W	
		D6	DMA_Count[30]		0		
		D5	DMA_Count[29]		0		
		D4	DMA_Count[28]		0		
		D3	DMA_Count[27]		0		
		D2	DMA_Count[26]		0		
		D1	DMA_Count[25]		0		
		D0	DMA_Count[24]		0		
DMA_Count_HL (DMA transfer byte counter high/low)	100009D (B)	D7	DMA_Count[23]	DMA transfer byte counter	0	R/W	
		D6	DMA_Count[22]		0		
		D5	DMA_Count[21]		0		
		D4	DMA_Count[20]		0		
		D3	DMA_Count[19]		0		
		D2	DMA_Count[18]		0		
		D1	DMA_Count[17]		0		
		D0	DMA_Count[16]		0		
DMA_Count_LH (DMA transfer byte counter low/high)	100009E (B)	D7	DMA_Count[15]	DMA transfer byte counter	0	R/W	
		D6	DMA_Count[14]		0		
		D5	DMA_Count[13]		0		
		D4	DMA_Count[12]		0		
		D3	DMA_Count[11]		0		
		D2	DMA_Count[10]		0		
		D1	DMA_Count[9]		0		
		D0	DMA_Count[8]		0		
DMA_Count_LL (DMA transfer byte counter low/low)	100009F (B)	D7	DMA_Count[7]	DMA transfer byte counter	0	R/W	
		D6	DMA_Count[6]		0		
		D5	DMA_Count[5]		0		
		D4	DMA_Count[4]		0		
		D3	DMA_Count[3]		0		
		D2	DMA_Count[2]		0		
		D1	DMA_Count[1]		0		
		D0	DMA_Count[0]		0		

These registers specify the data length in the DMA transfer in units of byte, and displays it. Its setting can be done as large as up to 0xFFFFFFFF bytes.
When the DMA is set to be in the free run mode by the setting of the CountMode bit of the DMA_Config_1 register (CountMode=0), values transmitted by the DMA can be referred at any time. In this mode, when the DMA Transfer Byte Counter exceeds 0xFFFFFFFF, it returns to 0x00000000 and the DMA_CountUp bit of the MainIntStat register is set to "1".
When the DMA is set to be in the countdown mode by the setting of the CountMode bit of the DMA_Config_1 register (CountMode=1), specify the total number of transmissions in the DMA Transfer Byte Counter, set the DMA_Go bit of the DMA_Control register to "1", and then start the DMA transfer.
In this mode, the DMA Transfer Byte Counter is decreased as much as the data quantity transferred by the DMA. When it reaches 0x00000000, the DMA ends. In this mode, the remained quantity of the data to transfer can be referred. Writing into these registers during the DMA transfer is neglected.
For reading these registers, access the DMA_Count_HH, HL, LH and LL registers in this order.

VII LCDC BLOCK

VII-1 INTRODUCTION

The LCDC block consists of an LCD controller and VRAM.

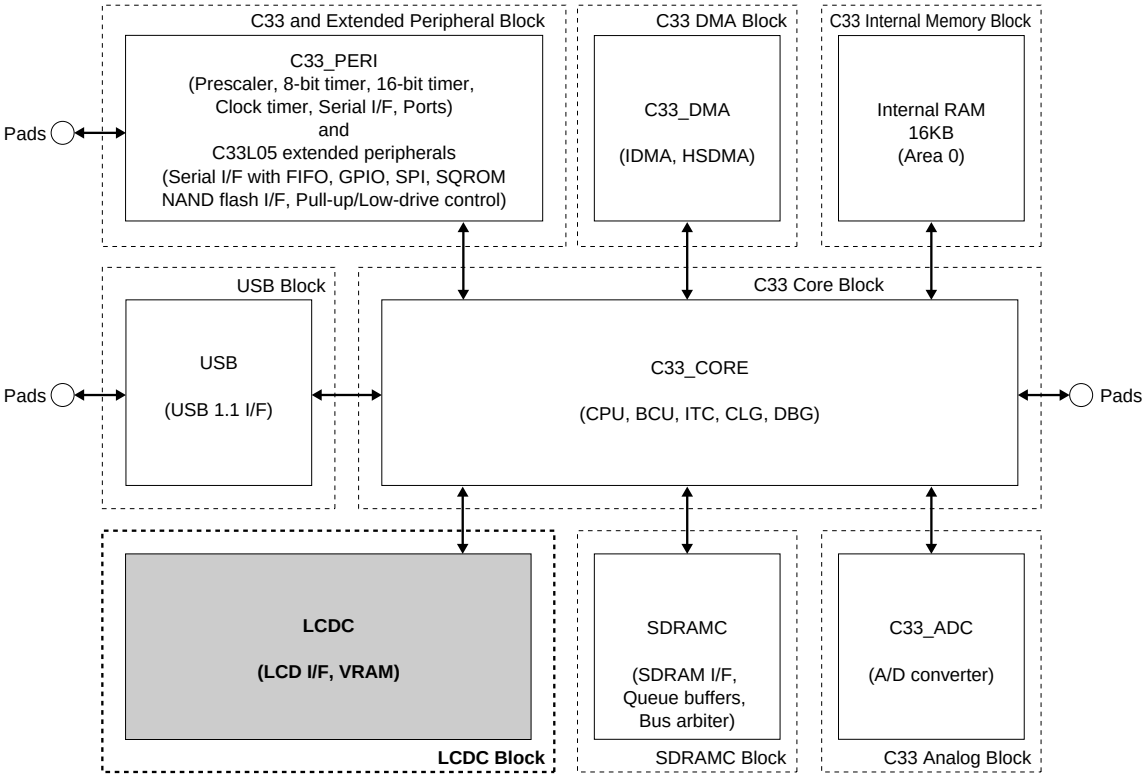


Figure VII.1.1 LCDC Block

THIS PAGE IS BLANK.

VII-2 LCD CONTROLLER

This section describes the functions and control procedures of the LCD controller.

Overview

Features

The features of the LCD controller (LCDC) are described below.

S1C33 core CPU interface and VRAM

- The UMA (Unified Memory Access) method using the Bus Arbiter and IVRAM Arbiter is implemented. This method allows the LCDC to access SDRAM (external VRAM) while the CPU is accessing an internal circuit, or to access IVRAM (internal VRAM) while the CPU is accessing another circuit.
- The LCDC registers are mapped into Area 6 and 16-bit accesses are possible.
- The internal VRAM (40KB) is mapped at addresses 0x3C0000 to 0x3C9FFF.
- The external VRAM map is configurable (max. 1MB in a 32MB SDRAM area).
- The LCDC interrupt signal is assigned to interrupt vector No. 89 (trap table base address + 0x164) in the ITC.

Compatible display types

- 4- or 8-bit monochrome LCD interface
- 4- or 8-bit color LCD interface
- Single-panel, single-drive passive displays
- Typical resolutions
  320 × 240 (16-bpp mode, external VRAM is required) * bpp = bits per pixel
  320 × 240 (4-bpp mode)
  160 × 240 (8-bpp mode)

Display modes

- Due to frame rate modulation, grayscale display is possible in up to 64 shades of gray when a monochrome passive LCD panel is used.
  Two-shade display using a 2 × 6-bit look-up table in 1-bpp mode
  Four-shade display using a 4 × 6-bit look-up table in 2-bpp mode
  16-shade display using a 16 × 6-bit look-up table in 4- or 12-bpp mode
  64-shade display using a 256 × 6-bit look-up table in 8- or 16-bpp mode
- Of 256K colors, a maximum of 64K colors can be simultaneously displayed on a color passive LCD panel.
  Two-color display using three 2 × 6-bit look-up tables in 1-bpp mode
  Four-color display using three 4 × 6-bit look-up tables in 2-bpp mode
  16-color display using three 16 × 6-bit look-up tables in 4-bpp mode
  256-color display using three 256 × 6-bit look-up tables in 8-bpp mode
  4K-color display using a 48 × 6-bit look-up table in 12-bpp mode
  64K-color display using a 128 × 6-bit look-up table in 16-bpp mode

Clock

- The PCLK (pixel clock) and MCLK (memory clock) for the LCD controller are generated in the LCDC clock generator by dividing the OSC3/PLL clock by 1 to 16.
- The UMA clock for the LCD controller can be selected from the BCUCLK or OSC3/PLL clock.
- These clocks are gated, so they can be disabled if not used.

Power save

- Software power-save mode
- DOZE mode is supported for RAM built-in or self-refresh-type LCD panels
- Blank display

Block Diagram

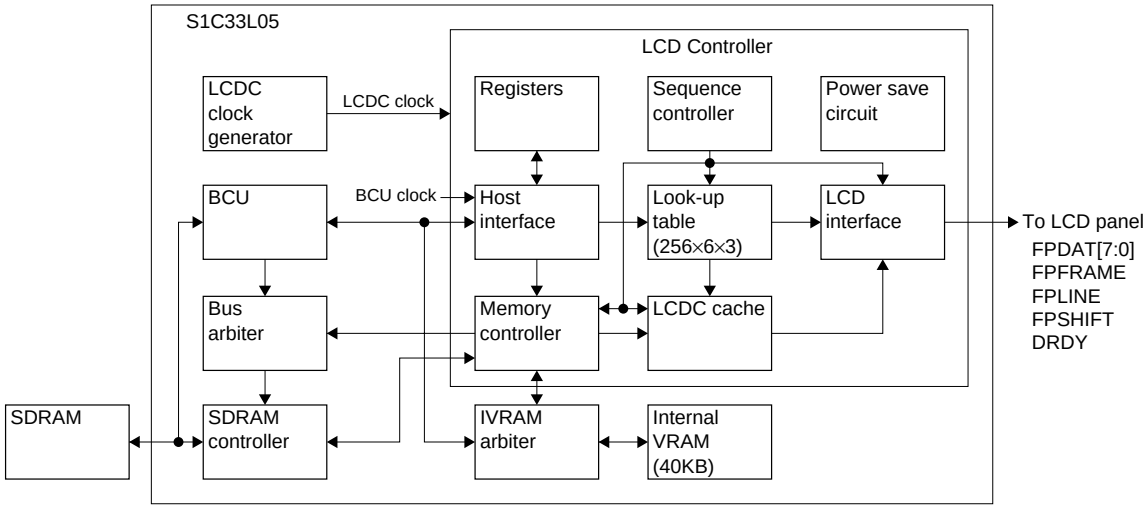


Figure VII.2.1 Block Diagram of the LCD Controller

Host interface

The C33 STD Core accesses the LCDC registers and look-up table though this host interface.

Memory controller

The memory controller arbitrates accesses for display data between the CPU and the LCDC. It also controls the LCDC cache.

LCDC cache

This consists of two 32-byte FIFO used as a display data cache for sending display data to the LCD panel.

Sequence controller

The sequence controller controls data flow from the memory controller to the LCD interface through the look-up table. It also generates display data memory addresses for refreshing display.

Look-up table

This consists of three 256 × 6-bit tables (red, green, and blue) and is used to set up the gray level or color data to be displayed. In monochrome mode, only the green look-up table is used.

LCD interface

The LCD interface performs frame rate modulation for passive LCD panels. It also formats display data and generates the timing control signals for various LCD panels.

Power save circuit

This circuit controls the power save mode in the LCDC.

I/O Pins of the LCD Controller

Table VII.2.1 lists the output pins of the LCD controller. Table VII.2.2 shows the pin configurations classified by type of LCD panel.

Table VII.2.1 I/O Pins of the LCD Controller

Pin name	I/O	Function	Function select bit
PB0(**FPDAT0**)	I/O	I/O port / LCD display data output (D0)	FPB0[1:0](D[1:0])/PB0–PB3 port function extension register(0x300F62)
PB1(**FPDAT1**)	I/O	I/O port / LCD display data output (D1)	FPB1[1:0](D[3:2])/PB0–PB3 port function extension register(0x300F62)
PB2(**FPDAT2**)	I/O	I/O port / LCD display data output (D2)	FPB2[1:0](D[5:4])/PB0–PB3 port function extension register(0x300F62)
PB3(**FPDAT3**)	I/O	I/O port / LCD display data output (D3)	FPB3[1:0](D[7:6])/PB0–PB3 port function extension register(0x300F62)
PB4(**FPDAT4**)	I/O	I/O port / LCD display data output (D4)	FPB4[1:0](D[1:0])/PB4–PB7 port function extension register(0x300F63)
PB5(**FPDAT5**)	I/O	I/O port / LCD display data output (D5)	FPB5[1:0](D[3:2])/PB4–PB7 port function extension register(0x300F63)
PB6(**FPDAT6**)	I/O	I/O port / LCD display data output (D6)	FPB6[1:0](D[5:4])/PB4–PB7 port function extension register(0x300F63)
PB7(**FPDAT7**)	I/O	I/O port / LCD display data output (D7)	FPB7[1:0](D[7:6])/PB4–PB7 port function extension register(0x300F63)
PC0(**FPFRAME**)	I/O	I/O port / LCD vertical scan start pulse output	FPC0[1:0](D[1:0])/PC0–PC3 port function extension register(0x300F64)
PC1(**FPLINE**)	I/O	I/O port / LCD display data latch clock output	FPC1[1:0](D[3:2])/PC0–PC3 port function extension register(0x300F64)
PC2(**FPSHIFT**)	I/O	I/O port / LCD display data shift clock output	FPC2[1:0](D[5:4])/PC0–PC3 port function extension register(0x300F64)
PC3(**DRDY**)	I/O	I/O port / LCD backplane bias signal output	FPC3[1:0](D[7:6])/PC0–PC3 port function extension register(0x300F64)

The LCD I/F signal pins are shared with the PB and PC I/O ports. At cold start, these pins are all set for the I/O port. When using the LCDC, configure these pins for the LCD interface using the PB and PC port function extension registers (0x300F62–0x300F64). At hot start, the registers retain their status from prior to the reset.

Table VII.2.2 Pin Configurations by Type of LCD Panel

Pin name	Monochrome passive panel		Color passive panel		
	4 bits	8 bits	4 bits	8-bit format 1	8-bit format 2
FPFRAME	FPFRAME				
FPLINE	FPLINE				
FPSHIFT	FPSHIFT				
DRDY	MOD	MOD	MOD	FPSHIFT2	MOD
FPDAT7	D3	D7	D3	D7	D7
FPDAT6	D2	D6	D2	D6	D6
FPDAT5	D1	D5	D1	D5	D5
FPDAT4	D0	D4	D0	D4	D4
FPDAT3	PB3	D3	PB3	D3	D3
FPDAT2	PB2	D2	PB2	D2	D2
FPDAT1	PB1	D1	PB1	D1	D1
FPDAT0	PB0	D0	PB0	D0	D0

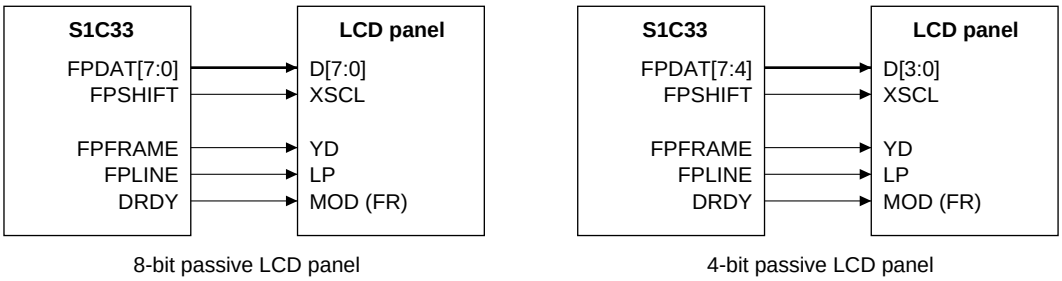


Figure VII.2.2 Typical LCD-Panel Connections

System Settings

Setting the BCU

The LCDC registers and internal VRAM are mapped into Area 6. Therefore, in order for the control registers and VRAM to be accessed, the BCU must be set up in accordance with the procedure described below.

1. A6IO (D9) / Access control register (0x48132) = "1"
   Configure Area 6 for an internal device area.
2. A6WT[2:0] (D[A:8]) / Areas 6-4 setup register (0x4812A) = "000" at x2 speed mode or "001" at x1 speed mode.
   This causes one wait cycle to be inserted when accessing Area 6 at x1 speed mode and no wait cycle to be inserted at x2 speed mode.
3. SWAITE (D0) / Bus control register (0x4812E) = "1"
   Enable the internal #WAIT signal.
4. A6EC (D1) / Access control register (0x48132) = "0"
   Select little endian as the Area 6 data format.
5. Byte, half-word or word access is allowed for IVRAM. But the LCDC registers and look-up table do not support byte access.

Selecting the Display Memory (VRAM)

The S1C33L05 has a built-in 40K-byte video RAM. The internal VRAM is mapped to addresses 0x3C0000 to 0x3C9FFF in Area 6.
The LCDC allows use of an external SDRAM as the display memory (max. 1MB).
Use VRAMSEL (DF) / Screen start address register (0x380210) to select which memory is used as the VRAM.
VRAMSEL = "0": Internal VRAM is used (default)
VRAMSEL = "1": External VRAM is used

When using the internal VRAM, no special setting is required except the BCU settings described above. The internal VRAM is divided into 5 banks and it is designed so that two different banks can be accessed from the CPU and LCDC simultaneously. If the same bank is accessed from the CPU and LCDC at the same time, accessing from the LCDC is enabled first in order to refresh the LCD display normally and the CPU should wait until the LCDC fills the LCDC cache with the display data read from the VRAM. For this control, IVRAM arbiter is provided. Refer to Section VII-3, "Internal VRAM and IVRAM Arbiter", for details.

When using an external SDRAM, it is necessary to set up the SDRAM controller. Refer to Section VIII-2, "SDRAM Controller", for controlling the SDRAM controller.
The LCDC uses only 1M bytes from the beginning with the SDRAM as the VRAM.

Setting the LCDC Clock

The following three clocks must be supplied to the LCD controller before the LCD controller can be used.

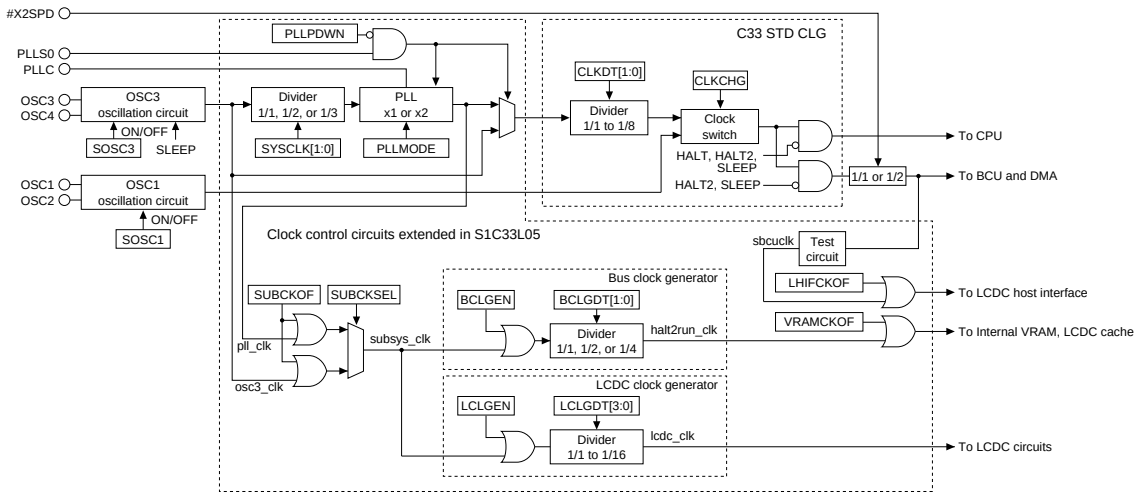


Figure VII.2.3 LCDC Clock

sbcuclk: This clock is used in the LCDC host interface. To supply this clock to the LCDC, set the LHIFCKOF (D0) / Module clock control register 0 (0x300F35) to "0" (default).

lcdc_clk: This clock is generated by the LCDC clock generator and is used as the pixel clock PCLK in the LCDC. To supply this clock to the LCDC, set the LCLGEN (D7) / LCDC clock generator control register (0x300F34) to "1".
The source clock and division ratio for generating the LCDC clock can be selected using the SUBCKSEL (D0) / Sub system clock select register (0x300F32) and the LCLGDT[3:0] (D[3:0]) / LCDC clock generator control register (0x300F34), respectively.

Selecting the source clock (subsys_clk)
SUBCKSEL = "0": OSC3 clock
SUBCKSEL = "1": PLL output clock

Table VII.2.3 Division Ratio for Generating LCDC Clock

LCLGDT3	LCLGDT2	LCLGDT1	LCLGDT0	LCLG source clock
1	1	1	1	subsys_clk / 16
1	1	1	0	subsys_clk / 15
1	1	0	1	subsys_clk / 14
1	1	0	0	subsys_clk / 13
1	0	1	1	subsys_clk / 12
1	0	1	0	subsys_clk / 11
1	0	0	1	subsys_clk / 10
1	0	0	0	subsys_clk / 9
0	1	1	1	subsys_clk / 8
0	1	1	0	subsys_clk / 7
0	1	0	1	subsys_clk / 6
0	1	0	0	subsys_clk / 5
0	0	1	1	subsys_clk / 4
0	0	1	0	subsys_clk / 3
0	0	0	1	subsys_clk / 2
0	0	0	0	subsys_clk

halt2run_clk: This clock is generated by the bus clock generator and is used as the operating clock for the internal VRAM, LCDC cache and SDRAM. To supply this clock to the LCDC, set the BCLGEN (D7) / Bus clock generator control register (0x300F33) to "1" and the VRAMCKOF (D0) / Module clock control register 1 (0x300F36) to "0".
The same source clock as for the lcdc_clk clock (OSC3 or PLL output) is used for generating the halt2run_clk clock. The division ratio can be selected using the BCLGDT[1:0] (D[1:0]) / Bus clock generator control register (0x300F33).

Table VII.2.4 Division Ratio for Generating Bus Clock

BCLGDT1	BCLGDT0	BCLG source clock
1	1	subsys_clk / 4
1	0	subsys_clk / 2
0	1	subsys_clk
0	0	

Refer to Section III-17, "S1C33L05 Clock System and Miscellaneous Registers", for more information about the clock system.

Setting the LCD Panel

Types of Panels

The LCD controller supports the following types of single-LCD panels.

- 4- or 8-bit monochrome passive LCD panel
- 4- or 8-bit color passive LCD panel

Dual panels are not supported.
The type of LCD panel used must be set in the LCD controller in advance, using the control bits described below.

Selecting between color and monochrome

Use COLOR (DE) / LCDC mode register 1 (0x380202) to select the type of LCD panel, either color or monochrome.
COLOR = "1": Color panel selected
COLOR = "0": Monochrome panel selected (default)

Selecting the data width

Use DWT[1:0] (D[B:A]) / LCDC mode register 1 (0x380202) to select the data width and format.

Table VII.2.5 Selection of the LCD Panel

COLOR	DWT1	DWT0	LCD panel
1	1	1	Color Single 8-bit passive LCD format 2
	1	0	Reserved
	0	1	Color Single 8-bit passive LCD format 1
	0	0	Color Single 4-bit passive LCD
0	1	1	Reserved
	1	0	Reserved
	0	1	Mono Single 8-bit passive LCD
	0	0	Mono Single 4-bit passive LCD

Resolution

Set the resolution and non-display period of the LCD panel in accordance with the procedure specified below.

Horizontal resolution

Set the value shown below in the HSIZE[6:0] (D[6:0]) / Horizontal panel size register (0x380042).

$$HSIZE[6:0] = \frac{\text{Horizontal resolution (number of pixels)}}{8} - 1$$

For example, if the LCD panel has a horizontal resolution of 320 dots, set 39 (= 0x27) in HSIZE.

Horizontal non-display period

Set the value shown below in the HNDP[4:0] (D[4:0]) / Horizontal non-display period register (0x380040).

$$HNDP[4:0] = \frac{\text{Horizontal non-display period (number of pixels)}}{8} - 4$$

Vertical resolution

Set the value shown below in VSIZE[9:0] (D[9:0]) / Vertical panel size register (0x38004C).

VSIZE[9:0] = Vertical resolution (number of lines) - 1

For example, if the LCD panel has a vertical resolution of 240 lines, set 239 (= 0xEF) in VSIZE.

Vertical non-display period

Set the value shown below in VNDP[5:0] (D[5:0]) / Vertical non-display period register (0x38004A).

VNDP[5:0] = Vertical non-display period (number of lines)

Display Modes

The number of gray levels in grayscale display and the number of colors in color display are determined by the number of bits representing each pixel (bpp = bits per pixel). Write this bpp value to BPP[2:0] (D[2:0]) / LCDC mode register 0 (0x380200) in order to set the display mode (number of gray levels/colors displayed).

Table VII.2.6 Specification of Display Modes

BPP2	BPP1	BPP0	Display mode	
			Monochrome (COLOR = "0")	Color (COLOR = "1")
1	1	*	Reserved	Reserved
1	0	1	16 bpp, 64 gray levels	16 bpp, 64K colors
1	0	0	12 bpp, 16 gray levels	12 bpp, 4K colors
0	1	1	8 bpp, 64 gray levels	8 bpp, 256 colors
0	1	0	4 bpp, 16 gray levels	4 bpp, 16 colors
0	0	1	2 bpp, 4 gray levels	2 bpp, 4 colors
0	0	0	1 bpp, 2 gray levels	1 bpp, 2 colors

(1) 1-bpp (2-gray-level/2-color) mode

One pixel is represented by 1 bit, displayed in two gray levels or two colors.
For monochrome LCD panels, 2-gray-level display can be obtained by assigning two gray levels from among the 64 gray levels available, including black and white, to two entries in the green look-up table (described later) (one each for bits = "0" and "1").
For color LCD panels, two colors from among the 256K colors available can be set in advance using two entries for pixel data "0" and "1" in each of the red, green, and blue look-up tables.
Data for eight consecutive pixels is stored as one byte in the display memory.

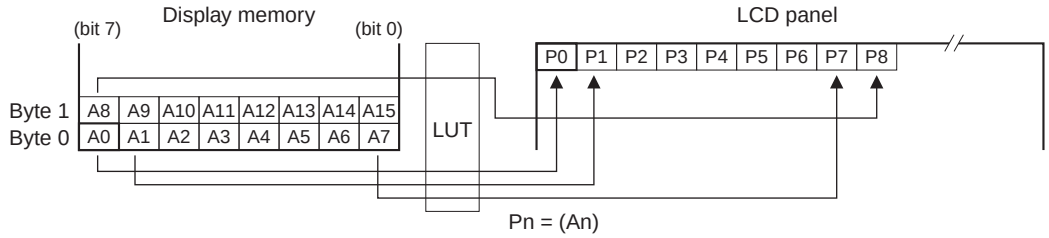


Figure VII.2.4 Data Format in 1-bpp Mode

The look-up table can be bypassed in this mode, i.e., black-white mode, to reduce power consumption.

(2) 2-bpp (4-gray-level/4-color) mode

One pixel is represented by 2 bits, displayed in four gray levels or four colors.
For monochrome LCD panels, 4-gray-level display can be obtained by assigning four gray levels from among the 64 gray levels available, including black and white, to four entries in the green look-up table (one each for bits = "00" to "11").
For color LCD panels, four colors from among the 256K colors available can be set in advance using four entries for pixel data "00" to "11" in each of the red, green, and blue look-up tables.
Data for four consecutive pixels is stored as one byte in the display memory.

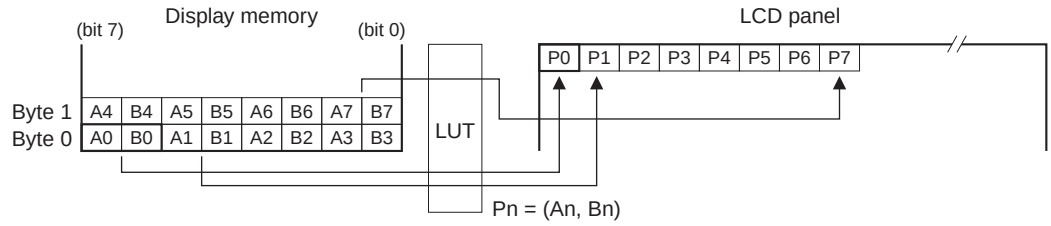


Figure VII.2.5 Data Format in 2-bpp Mode

The look-up table can be bypassed in this mode to reduce power consumption.

(3) 4-bpp (16-gray-level/16-color) mode

One pixel is represented by 4 bits, displayed in 16 gray levels or 16 colors.
For monochrome LCD panels, 16-gray-level display can be obtained by assigning 16 gray levels from among the 64 gray levels available, including black and white, to 16 entries in the green look-up table (one each for bits = "0000" to "1111").
For color LCD panels, 16 colors from among the 256K colors available can be set in advance using 16 entries for pixel data "0000" to "1111" in each of the red, green, and blue look-up tables.
Data for two consecutive pixels is stored as one byte in the display memory.

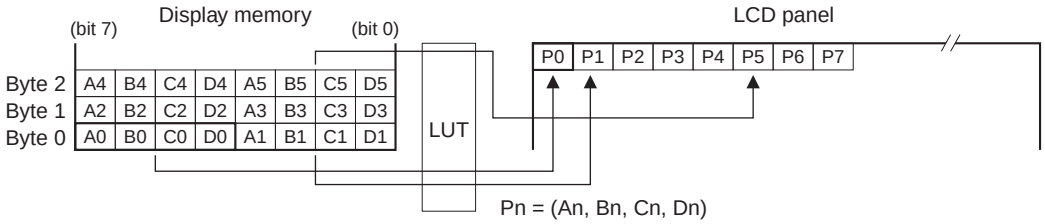


Figure VII.2.6 Data Format in 4-bpp Mode

The look-up table can be bypassed in this mode to reduce power consumption.

(4) 8-bpp (64-gray-level/256-color) mode

One pixel is represented by 8 bits, displayed in 64 gray levels or 256 colors.
For monochrome LCD panels, 64-gray-level display can be obtained by assigning 64 gray levels, including black and white, to 256 entries in the green look-up table.
For color LCD panels, 256 colors from among the 256K colors available can be set in advance using 256 entries for pixel data "0x00" to "0xFF" in each of the red, green, and blue look-up tables.
Data for one pixel is stored as one byte in the display memory.

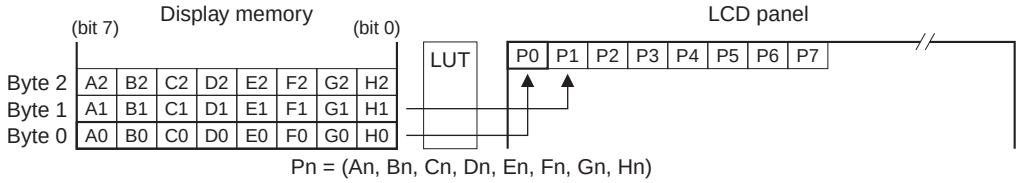


Figure VII.2.7 Data Format in 8-bpp Mode

The look-up tables can be bypassed in this mode. In this case, the display data stored in the display memory directly specifies a gray level or color. The following figure shows the correspondence between the memory data and the pixel data to be sent to the LCD panel.

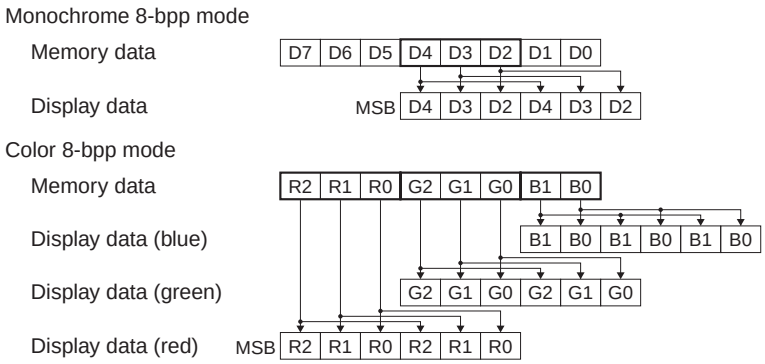


Figure VII.2.8 Pixel Data when LUT is Bypassed

(5) 12-bpp (16-gray-level/4K-color) mode

One pixel is represented by 12 bits, displayed in 16 gray levels or 4K colors.
For monochrome LCD panels, a 16-gray-level display can be obtained by assigning 16 gray levels from among the 64 gray levels available, including black and white, to 16 entries in the green look-up table (one each for bits = "0000" to "1111"). In this mode, only the D7 to D4 bits of the 12-bit pixel data are used to specify a gray level.
Data for two pixels is stored as three bytes in the display memory.

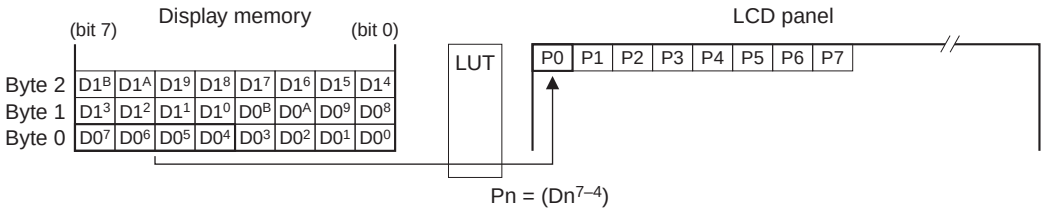


Figure VII.2.9 Monochrome Display Data Format in 12-bpp Mode

For color LCD panels, 4K discrete combinations are configured using 16 entries in each of the red, green and blue look-up tables.
Data for two pixels is stored as three bytes in the display memory.

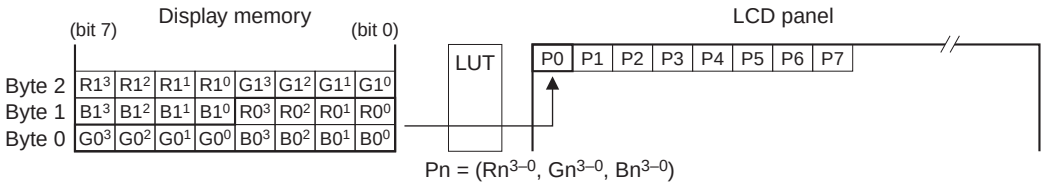


Figure VII.2.10 Color Display Data Format in 12-bpp Mode

The look-up tables can be bypassed in this mode. In this case, the display data stored in the display memory directly specifies a gray level or color. The following figure shows the correspondence between the memory data and the pixel data to be sent to the LCD panel.

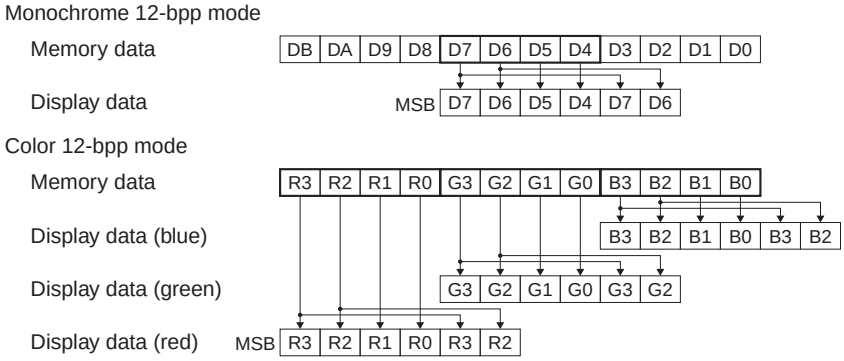


Figure VII.2.11 Pixel Data when LUT is Bypassed

(6) 16-bpp (64-gray-level/64K-color) mode

One pixel is represented by 16 bits, displayed in 64 gray levels or 64K colors.
For monochrome LCD panels, a 64-gray-level display can be obtained by assigning 64 gray levels, including black and white, to 64 entries in the green look-up table (one each for bits = "000000" to "111111"). In this mode, only the D10 to D5 bits of the 16-bit pixel data are used to specify a gray level.
Data for one pixel is stored as two bytes in the display memory.

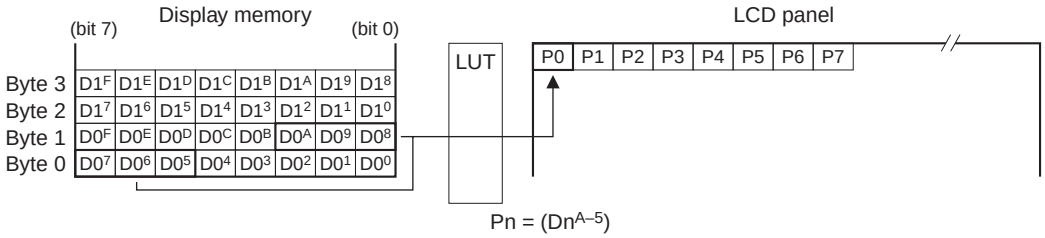


Figure VII.2.12 Monochrome Display Data Format in 16-bpp Mode

For color LCD panels, 64K discrete combinations are configured using 32 entries in each of the red and blue look-up tables, and 64 entries in the green look-up table.
Data for one pixel is stored as two bytes in the display memory.

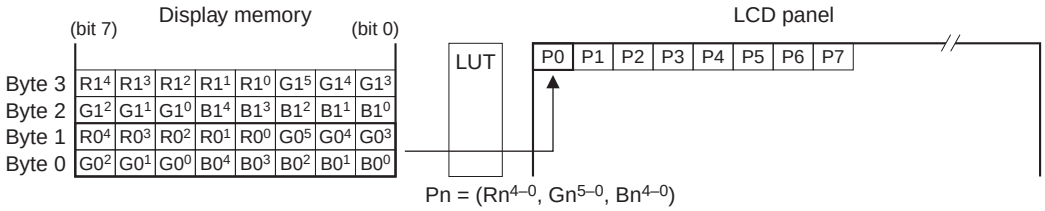


Figure VII.2.13 Color Display Data Format in 16-bpp Mode

The look-up tables can be bypassed in this mode. In this case, the display data stored in the display memory directly specifies a gray level or color. The following figure shows the correspondence between the memory data and the pixel data to be sent to the LCD panel.

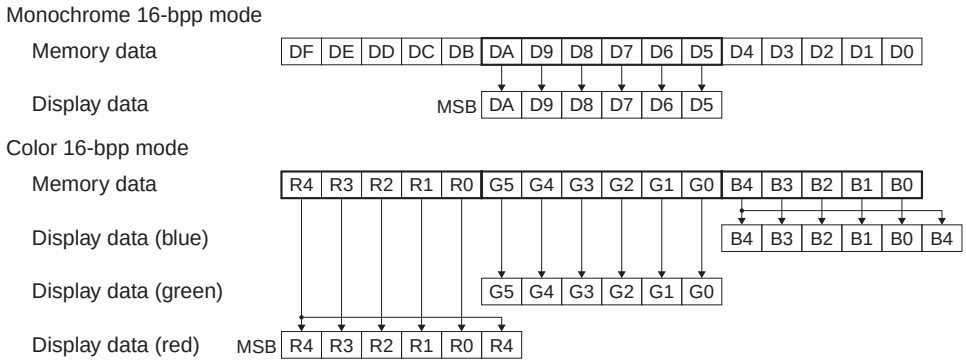


Figure VII.2.14 Pixel Data when LUT is Bypassed

Look-up Tables

The LCD controller contains a look-up table consisting of 256 6-bit entries, one for each of the RGB color elements (red, green, and blue).

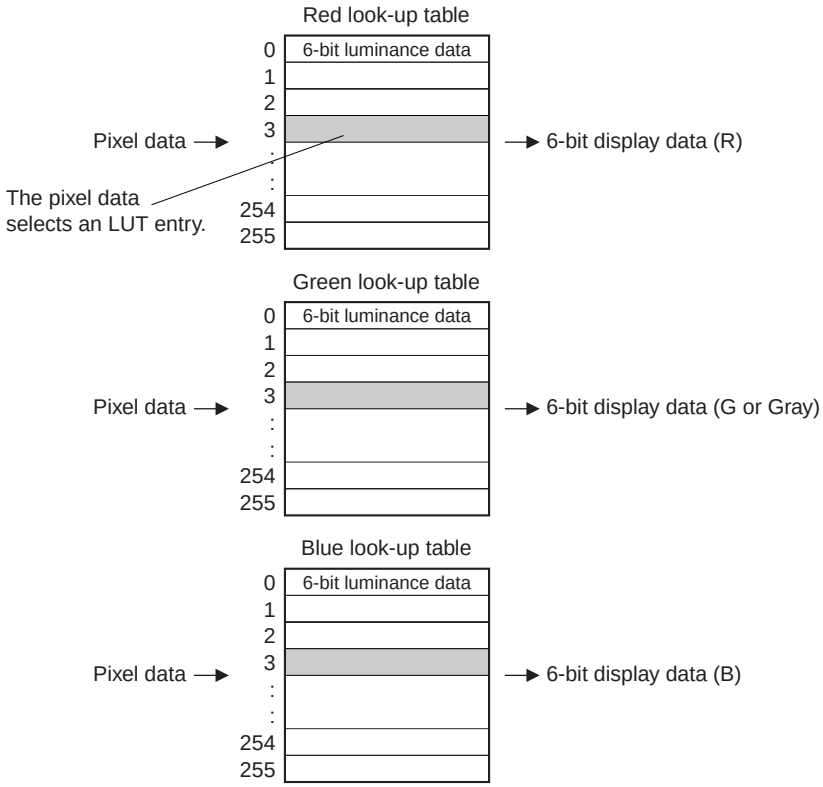


Figure VII.2.15 Configuration of the Look-up Tables

The pixel data in the display memory is used as an index to the look-up tables, so that luminance data is generated based on the values in the entries indicated by the pixel data, before being output to the LCD panel.

The LCD controller can control reversal of the display. This control is exercised on the output of the look-up tables.

The look-up tables can be bypassed in 1-bpp, 2-bpp, 4-bpp, 8-bpp, 12-bpp and 16-bpp modes. In this case, the pixel data stored in the display memory directly specifies a gray level or color. To bypass the look-up table, set LUTPASS (D4) / LCDC mode register 0 (0x380200) to "1".

Grayscale-mode look-up tables

In grayscale mode, the LCD controller uses only the green look-up table. For display in grayscale mode, select the data to be written to the look-up table from the 64 gray levels represented by 6 bits. The data 0x0, 0x20, and 0x3F represent black, 50% gray, and white, respectively. The differences in configuration between display modes are shown below.

(1) 1-bpp (2-gray-level) mode

Use the first two entries of the green look-up table. Select two pieces of data from the 64 gray levels, and write them to the respective entries. The data in entry 0 is output for pixel data "0", and the data in entry 1 is output for pixel data "1". For monochrome display, write 0x0 to entry 0 and 0x3F to entry 1 before using the LCD panel.

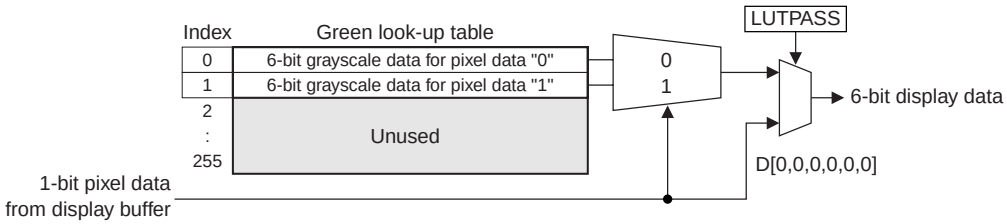


Figure VII.2.16 Look-up Table in 1-bpp (2-Gray-Level) Mode

Table VII.2.7 shows an example of the basic data setting.

Table VII.2.7 Example of Look-up-Table Settings in 1-bpp (2-Gray-Level) Mode

Index	R look-up table	G look-up table	B look-up table
0	0	0	0
1	0	0xFC	0
2–255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(2) 2-bpp (4-gray-level) mode

Use the first four entries of the green look-up table. Select four pieces of data from the 64 gray levels, and write them to the respective entries. The data in entry 0 is output for pixel data "00", and the data in entry 3 is output for pixel data "11".

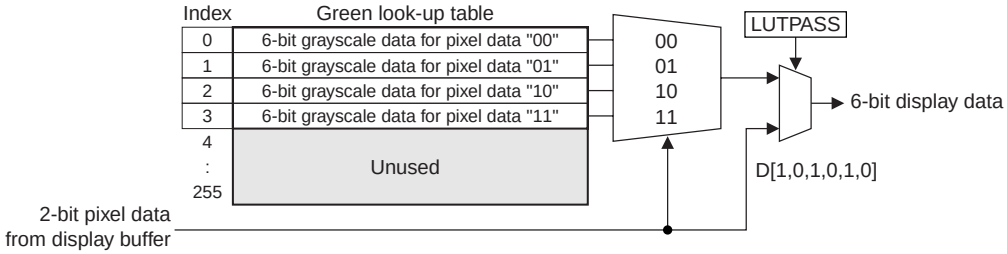


Figure VII.2.17 Look-up Table in 2-bpp (4-Gray-Level) Mode

Table VII.2.8 shows an example of the basic data setting.

Table VII.2.8 Example of Look-up-Table Settings in 2-bpp (4-Gray-Level) Mode

Index	R look-up table	G look-up table	B look-up table
0	0	0	0
1	0	0x54	0
2	0	0xA8	0
3	0	0xFC	0
4–255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(3) 4-bpp (16-gray-level) mode

Use the first 16 entries of the green look-up table. Select 16 pieces of data from the 64 gray levels, and write them to the respective entries. The data in entry 0 is output for pixel data "0000", and the data in entry 15 is output for pixel data "1111".

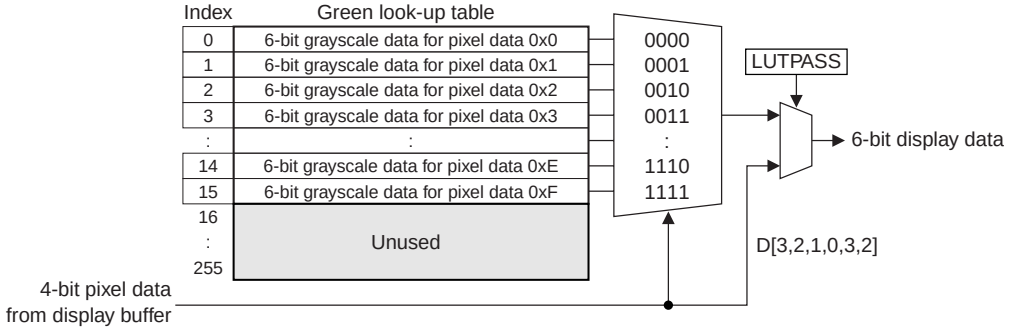


Figure VII.2.18 Look-up Table in 4-bpp (16-Gray-Level) Mode

Table VII.2.9 shows an example of the basic data setting.

Table VII.2.9 Example of Look-up-Table Settings in 4-bpp (16-Gray-Level) Mode

Index	R look-up table	G look-up table	B look-up table
0	0	0	0
1	0	0x10	0
2	0	0x20	0
3	0	0x30	0
4	0	0x44	0
5	0	0x54	0
6	0	0x64	0
7	0	0x74	0
8	0	0x88	0
9	0	0x98	0
10	0	0xA8	0
11	0	0xB8	0
12	0	0xCC	0
13	0	0xDC	0
14	0	0xEC	0
15	0	0xFC	0
16–255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(4) 8-bpp (64-gray-level) mode

Use all entries of the green look-up table. All 64 gray levels can be assigned to the look-up table. The data in entry 0 is output for pixel data "0x0", and the data in entry 255 is output for pixel data "0xFF".

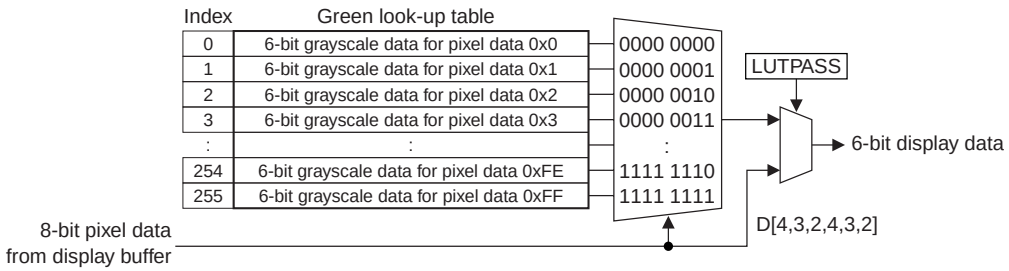


Figure VII.2.19 Look-up Table in 8-bpp (64-Gray-Level) Mode

Table VII.2.10 shows an example of the basic data setting.

Table VII.2.10 Example of Look-up-Table Settings in 8-bpp (64-Gray-Level) Mode

Index	R look-up table	G look-up table	B look-up table
0–3	0	0	0
4–7	0	4	0
8–11	0	8	0
12–15	0	0xC	0
16–19	0	0x10	0
:	0	:	0
240–243	0	0xF0	0
244–247	0	0xF4	0
248–251	0	0xF8	0
252–255	0	0xFC	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(5) 12-bpp (16-gray-level) mode

Use the first 16 entries of the green look-up table. Select 16 pieces of data from the 64 gray levels, and write them to the respective entries. The data in entry 0 is output when D[7:4] of the pixel data is "0000", and the data in entry 15 is output when D[7:4] of the pixel data is "1111".

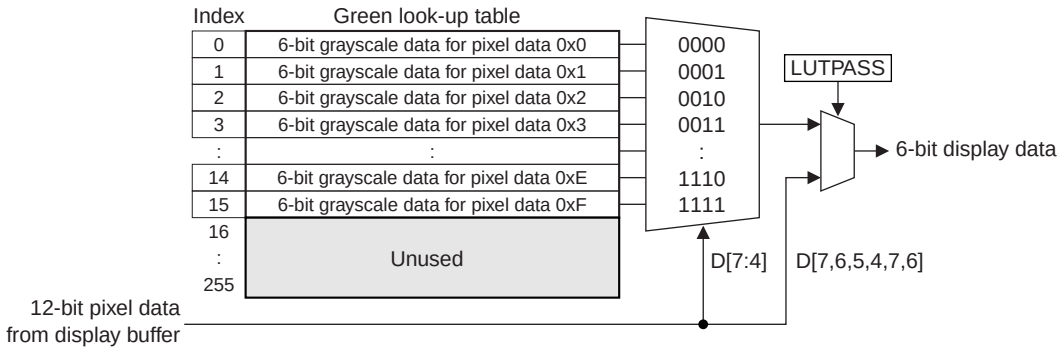


Figure VII.2.20 Look-up Table in 12-bpp (16-Gray-Level) Mode

Table VII.2.11 shows an example of the basic data setting.

Table VII.2.11 Example of Look-up-Table Settings in 12-bpp (16-Gray-Level) Mode

Index	R look-up table	G look-up table	B look-up table
0	0	0	0
1	0	0x10	0
2	0	0x20	0
3	0	0x30	0
4	0	0x44	0
5	0	0x54	0
6	0	0x64	0
7	0	0x74	0
8	0	0x88	0
9	0	0x98	0
10	0	0xA8	0
11	0	0xB8	0
12	0	0xCC	0
13	0	0xDC	0
14	0	0xEC	0
15	0	0xFC	0
16–255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(6) 16-bpp (64-gray-level) mode

Use the first 64 entries of the green look-up table. All 64 gray levels can be assigned to the look-up table. The data in entry 0 is output when D[A:5] of the pixel data is "0x0", and the data in entry 63 is output when D[A:5] of the pixel data is "0x3F".

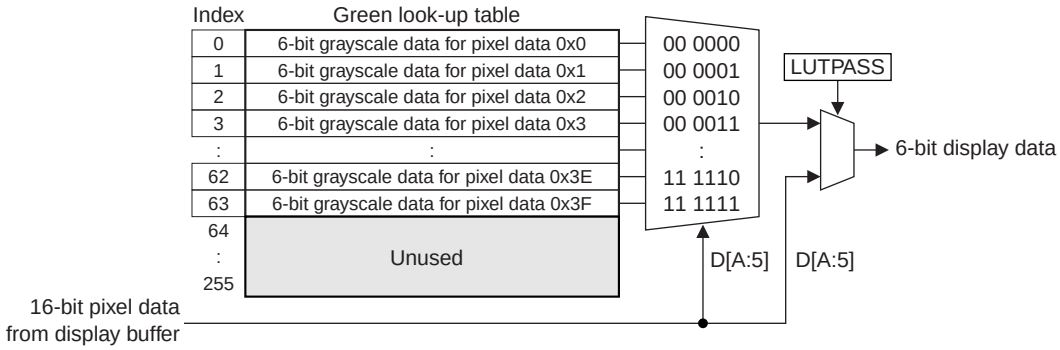


Figure VII.2.21 Look-up Table in 16-bpp (16-Gray-Level) Mode

Table VII.2.12 shows an example of the basic data setting.

Table VII.2.12 Example of Look-up-Table Settings in 16-bpp (64-Gray-Level) Mode

Index	R look-up table	G look-up table	B look-up table
0	0	0	0
1	0	4	0
2	0	8	0
3	0	0xC	0
4	0	0x10	0
:	0	:	0
60	0	0xF0	0
61	0	0xF4	0
62	0	0xF8	0
63	0	0xFC	0
64–255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

Color-mode look-up tables

In color mode, the LCD controller uses the red (R), green (G), and blue (B) look-up tables. Each color element is represented by 6-bit data. RGB = 00•00•00 is black, RGB = 3F•00•00 is red, RGB = 00•20•00 is 50% luminance green, RGB = 3F•00•3F is magenta, RGB = 3F•3F•3F is white, and so on. In this way, colors are determined by the proportions of the three color elements. If the luminance of each color element is represented by 6 bits, then we obtain $64 \times 64 \times 64 = 256K$ colors. Of these, select as many pieces of color data as can be used for the available display mode (2, 4, 16, 256, 4K, or 64K colors), and write them to the valid entries of the look-up tables before using the LCD panel.

The differences in configurations between display modes are shown below.

(1) 1-bpp (2-color) mode

Use the first two entries of each look-up table. Select 2-color data from among the 256K colors, and write it to the respective entries. The RGB data in entry 0 is output for pixel data "0", and the RGB data in entry 1 is output for pixel data "1". For monochrome display, write 0x0 to entry 0 and 0x3F to entry 1 in each of the red, green, and blue look-up tables before using the LCD panel.

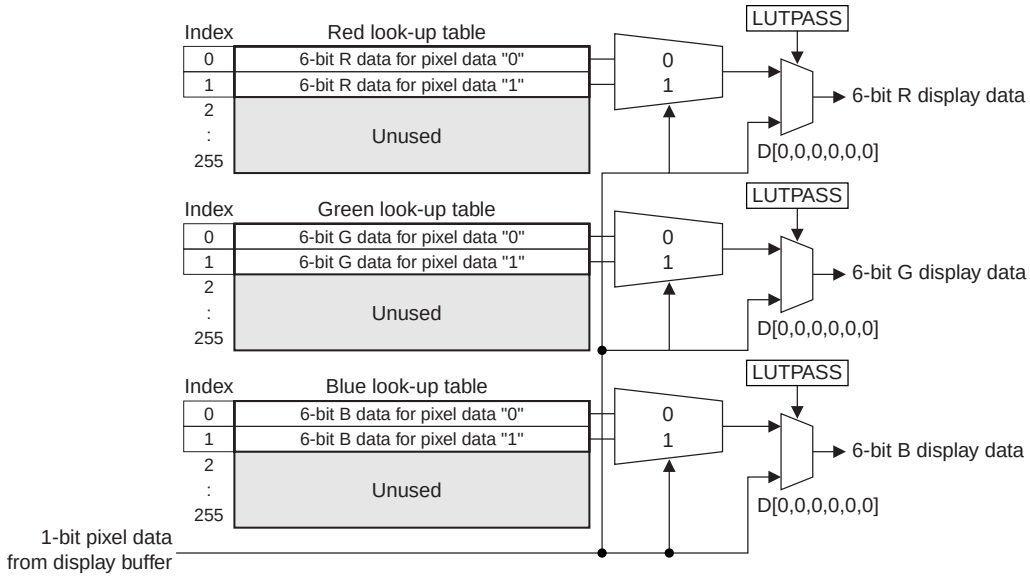


Figure VII.2.22 Look-up Table in 1-bpp (2-Color) Mode

Table VII.2.13 shows an example of the basic data setting.

Table VII.2.13 Example of Look-up-Table Settings in 1-bpp (2-Color) Mode

Index	R look-up table	G look-up table	B look-up table
0	0	0	0
1	0xFC	0xFC	0xFC
2–255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(2) 2-bpp (4-color) mode

Use the first four entries of each look-up table. Select 4-color data from among the 256K colors, and write it to the respective entries. The RGB data in entry 0 is output for pixel data "00", and the RGB data in entry 3 is output for pixel data "11".

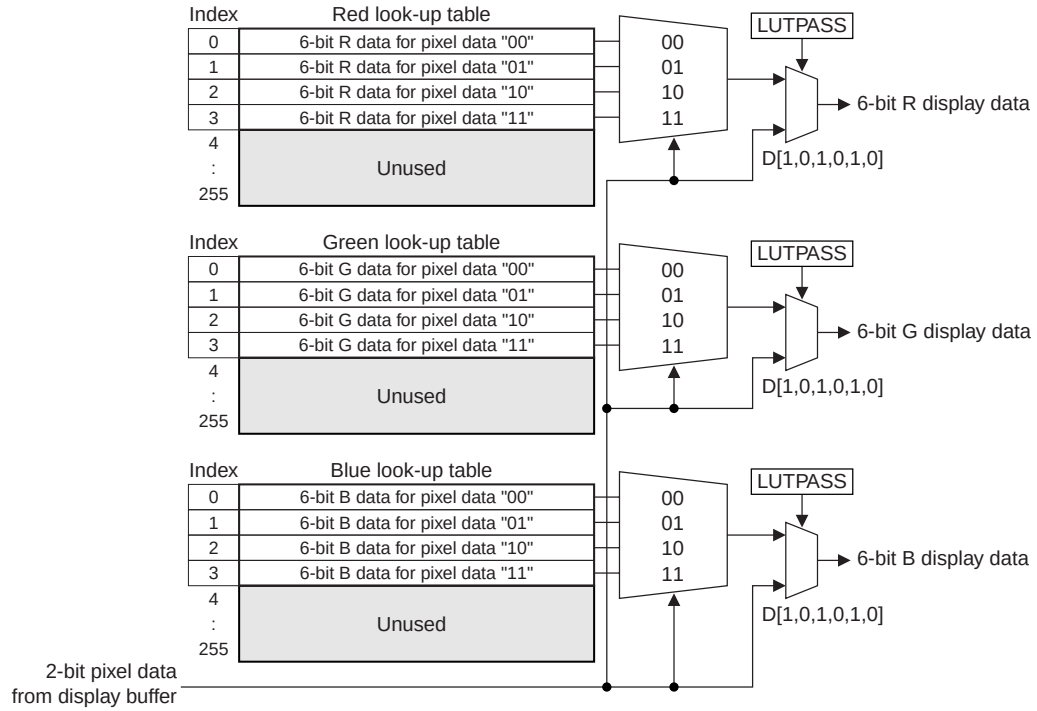


Figure VII.2.23 Look-up Table in 2-bpp (4-Color) Mode

Table VII.2.14 shows an example of the basic data setting.

Table VII.2.14 Example of Look-up-Table Settings in 2-bpp (4-Color) Mode

Index	R look-up table	G look-up table	B look-up table
0	0	0	0
1	0	0	0xFC
2	0xFC	0	0
3	0xFC	0xFC	0xFC
4–255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(3) 4-bpp (16-color) mode

Use the first 16 entries of each look-up table. Select 16-color data from among the 256K colors, and write it to the respective entries. The RGB data in entry 0 is output for pixel data "0000", and the RGB data in entry 15 is output for pixel data "1111".

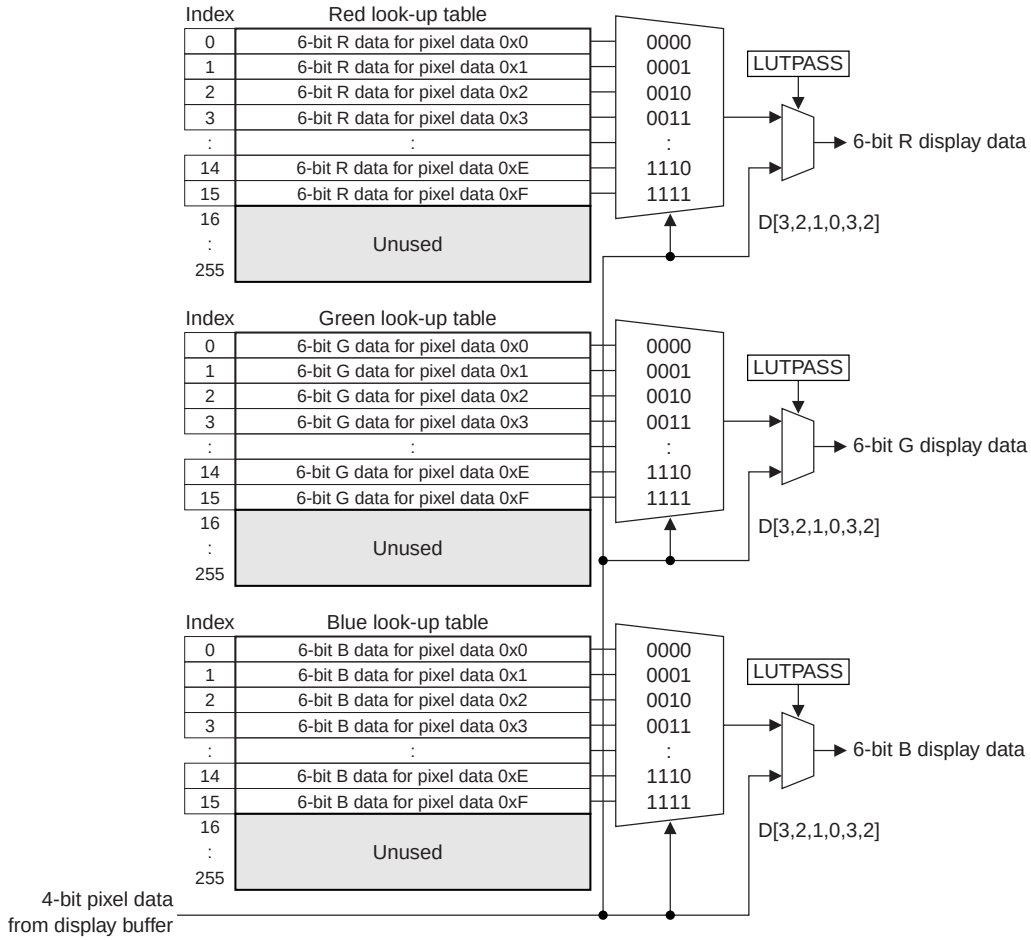


Figure VII.2.24 Look-up Table in 4-bpp (16-Color) Mode

Table VII.2.15 shows an example of the basic data setting.

Table VII.2.15 Example of Look-up-Table Settings in 4-bpp (16-Color) Mode

Index	R look-up table	G look-up table	B look-up table
0	0	0	0
1	0x80	0	0
2	0	0x80	0
3	0x80	0x80	0
4	0	0	0x80
5	0x80	0	0x80
6	0	0x80	0x80
7	0xC0	0xC0	0xC0
8	0x80	0x80	0x80
9	0xFC	0	0
10	0	0xFC	0
11	0xFC	0xFC	0
12	0	0	0xFC
13	0xFC	0	0xFC
14	0	0xFC	0xFC
15	0xFC	0xFC	0xFC
16–255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(4) 8-bpp (256-color) mode

Use all entries of each look-up table. Select 256-color data from among the 256K colors, and write it to the respective entries. The RGB data in entry 0 is output for pixel data "0x0", and the RGB data in entry 255 is output for pixel data "0xFF".

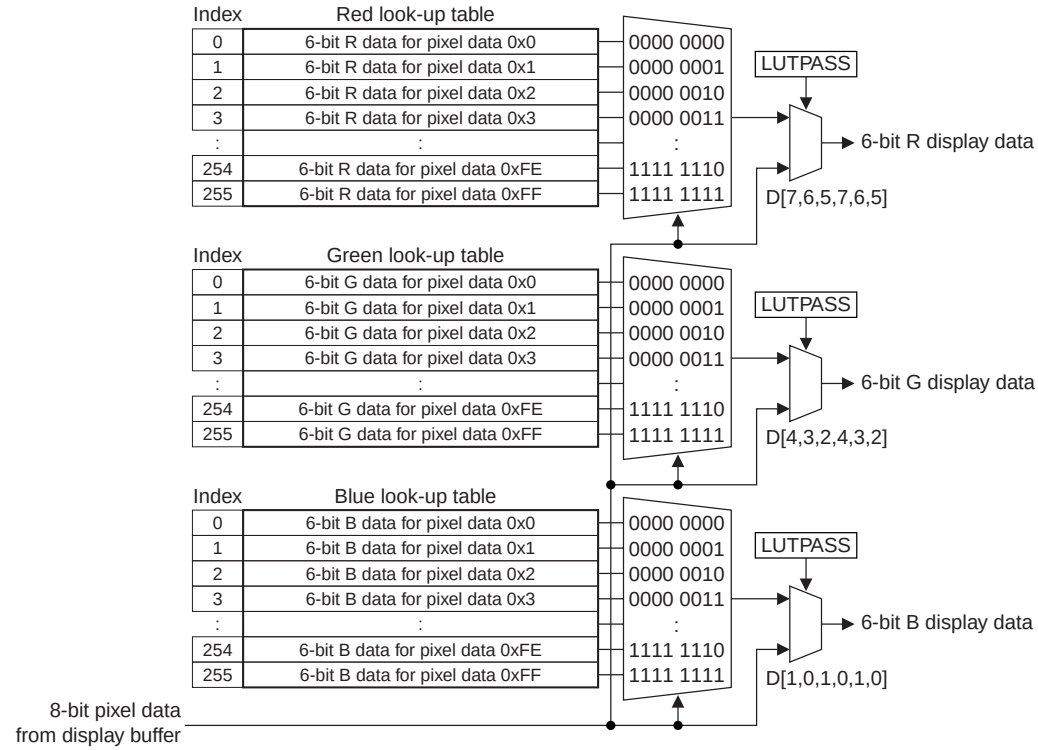


Figure VII.2.25 Look-up Table in 8-bpp (256-Color) Mode

Table VII.2.16 shows an example of the basic data setting.

Table VII.2.16 Example of Look-up-Table Settings in 8-bpp (256-Color) Mode

Index	R	G	B	Index	R	G	B	Index	R	G	B	Index	R	G	B
0	0	0	0	64	0xF0	0x70	0x70	128	0x30	0x30	0x70	192	0	0x40	0x0
1	0	0	0xA0	65	0xF0	0x90	0x70	129	0x40	0x30	0x70	193	0	0x40	0x10
2	0	0xA0	0	66	0xF0	0xB0	0x70	130	0x50	0x30	0x70	194	0	0x40	0x20
3	0	0xA0	0xA0	67	0xF0	0xD0	0x70	131	0x60	0x30	0x70	195	0	0x40	0x30
4	0xA0	0	0	68	0xF0	0xF0	0x70	132	0x70	0x30	0x70	196	0	0x40	0x40
5	0xA0	0	0xA0	69	0xD0	0xF0	0x70	133	0x70	0x30	0x60	197	0	0x30	0x40
6	0xA0	0x50	0	70	0xB0	0xF0	0x70	134	0x70	0x30	0x50	198	0	0x20	0x40
7	0xA0	0xA0	0xA0	71	0x90	0xF0	0x70	135	0x70	0x30	0x40	199	0	0x10	0x40
8	0x50	0x50	0x50	72	0x70	0xF0	0x70	136	0x70	0x30	0x30	200	0x20	0x20	0x40
9	0x50	0x50	0xF0	73	0x70	0xF0	0x90	137	0x70	0x40	0x30	201	0x20	0x20	0x40
10	0x50	0xF0	0x50	74	0x70	0xF0	0xB0	138	0x70	0x50	0x30	202	0x30	0x20	0x40
11	0x50	0xF0	0xF0	75	0x70	0xF0	0xD0	139	0x70	0x60	0x30	203	0x30	0x20	0x40
12	0xF0	0x50	0x50	76	0x70	0xF0	0xF0	140	0x70	0x70	0x30	204	0x40	0x20	0x40
13	0xF0	0x50	0xF0	77	0x70	0xD0	0xF0	141	0x60	0x70	0x30	205	0x40	0x20	0x30
14	0xF0	0xF0	0x50	78	0x70	0xB0	0xF0	142	0x50	0x70	0x30	206	0x40	0x20	0x30
15	0xF0	0xF0	0xF0	79	0x70	0x90	0xF0	143	0x40	0x70	0x30	207	0x40	0x20	0x20
16	0	0	0	80	0xB0	0xB0	0xF0	144	0x30	0x70	0x30	208	0x40	0x20	0x20
17	0x10	0x10	0x10	81	0xC0	0xB0	0xF0	145	0x30	0x70	0x40	209	0x40	0x20	0x20
18	0x20	0x20	0x20	82	0xD0	0xB0	0xF0	146	0x30	0x70	0x50	210	0x40	0x30	0x20
19	0x20	0x20	0x20	83	0xE0	0xB0	0xF0	147	0x30	0x70	0x60	211	0x40	0x30	0x20
20	0x30	0x30	0x30	84	0xF0	0xB0	0xF0	148	0x30	0x70	0x70	212	0x40	0x40	0x20
21	0x40	0x40	0x40	85	0xF0	0xB0	0xE0	149	0x30	0x60	0x70	213	0x30	0x40	0x20
22	0x50	0x50	0x50	86	0xF0	0xB0	0xD0	150	0x30	0x50	0x70	214	0x30	0x40	0x20
23	0x60	0x60	0x60	87	0xF0	0xB0	0xC0	151	0x30	0x40	0x70	215	0x20	0x40	0x20
24	0x70	0x70	0x70	88	0xF0	0xB0	0xB0	152	0x50	0x50	0x70	216	0x20	0x40	0x20
25	0x80	0x80	0x80	89	0xF0	0xC0	0xB0	153	0x50	0x50	0x70	217	0x20	0x40	0x20
26	0x90	0x90	0x90	90	0xF0	0xD0	0xB0	154	0x60	0x50	0x70	218	0x20	0x40	0x30
27	0xA0	0xA0	0xA0	91	0xF0	0xE0	0xB0	155	0x60	0x50	0x70	219	0x20	0x40	0x30
28	0xB0	0xB0	0xB0	92	0xF0	0xF0	0xB0	156	0x70	0x50	0x70	220	0x20	0x40	0x40
29	0xC0	0xC0	0xC0	93	0xE0	0xF0	0xB0	157	0x70	0x50	0x60	221	0x20	0x30	0x40
30	0xE0	0xE0	0xE0	94	0xD0	0xF0	0xB0	158	0x70	0x50	0x60	222	0x20	0x30	0x40
31	0xF0	0xF0	0xF0	95	0xC0	0xF0	0xB0	159	0x70	0x50	0x50	223	0x20	0x20	0x40
32	0	0	0xF0	96	0xB0	0xF0	0xB0	160	0x70	0x50	0x50	224	0x20	0x20	0x40
33	0x40	0	0xF0	97	0xB0	0xF0	0xC0	161	0x70	0x50	0x50	225	0x30	0x20	0x40
34	0x70	0	0xF0	98	0xB0	0xF0	0xD0	162	0x70	0x60	0x50	226	0x30	0x20	0x40
35	0xB0	0	0xF0	99	0xB0	0xF0	0xE0	163	0x70	0x60	0x50	227	0x30	0x20	0x40
36	0xF0	0	0xF0	100	0xB0	0xF0	0xF0	164	0x70	0x70	0x50	228	0x40	0x20	0x40
37	0xF0	0	0xB0	101	0xB0	0xE0	0xF0	165	0x60	0x70	0x50	229	0x40	0x20	0x30
38	0xF0	0	0x70	102	0xB0	0xD0	0xF0	166	0x60	0x70	0x50	230	0x40	0x20	0x30
39	0xF0	0	0x40	103	0xB0	0xC0	0xF0	167	0x50	0x70	0x50	231	0x40	0x20	0x30
40	0xF0	0	0	104	0	0	0x70	168	0x50	0x70	0x50	232	0x40	0x20	0x20
41	0xF0	0x40	0	105	0x10	0	0x70	169	0x50	0x70	0x50	233	0x40	0x30	0x20
42	0xF0	0x70	0	106	0x30	0	0x70	170	0x50	0x70	0x60	234	0x40	0x30	0x20
43	0xF0	0xB0	0	107	0x50	0	0x70	171	0x50	0x70	0x60	235	0x40	0x30	0x20
44	0xF0	0xF0	0	108	0x70	0	0x70	172	0x50	0x70	0x70	236	0x40	0x40	0x20
45	0xB0	0xF0	0	109	0x70	0	0x50	173	0x50	0x60	0x70	237	0x30	0x40	0x20
46	0x70	0xF0	0	110	0x70	0	0x30	174	0x50	0x60	0x70	238	0x30	0x40	0x20
47	0x40	0xF0	0	111	0x70	0	0x10	175	0x50	0x50	0x70	239	0x30	0x40	0x20
48	0	0xF0	0	112	0x70	0	0	176	0	0	0x40	240	0x20	0x40	0x20
49	0	0xF0	0x40	113	0x70	0x10	0	177	0x10	0	0x40	241	0x20	0x40	0x30
50	0	0xF0	0x70	114	0x70	0x30	0	178	0x20	0	0x40	242	0x20	0x40	0x30
51	0	0xF0	0xB0	115	0x70	0x50	0	179	0x30	0	0x40	243	0x20	0x40	0x30
52	0	0xF0	0xF0	116	0x70	0x70	0	180	0x40	0	0x40	244	0x20	0x40	0x40
53	0	0xB0	0xF0	117	0x50	0x70	0	181	0x40	0	0x30	245	0x20	0x30	0x40
54	0	0x70	0xF0	118	0x30	0x70	0	182	0x40	0	0x20	246	0x20	0x30	0x40
55	0	0x40	0xF0	119	0x10	0x70	0	183	0x40	0	0x10	247	0x20	0x30	0x40
56	0x70	0x70	0xF0	120	0	0x70	0	184	0x40	0	0	248	0	0	0
57	0x90	0x70	0xF0	121	0	0x70	0x10	185	0x40	0x10	0	249	0	0	0
58	0xB0	0x70	0xF0	122	0	0x70	0x30	186	0x40	0x20	0	250	0	0	0
59	0xD0	0x70	0xF0	123	0	0x70	0x50	187	0x40	0x30	0	251	0	0	0
60	0xF0	0x70	0xF0	124	0	0x70	0x70	188	0x40	0x40	0	252	0	0	0
61	0xF0	0x70	0xD0	125	0	0x50	0x70	189	0x30	0x40	0	253	0	0	0
62	0xF0	0x70	0xB0	126	0	0x30	0x70	190	0x20	0x40	0	254	0	0	0
63	0xF0	0x70	0x90	127	0	0x10	0x70	191	0x10	0x40	0	255	0	0	0

The above table shows 8-bit values. They can be written to the look-up table data registers and only the high-order 6 bits are set into the look-up table (the low-order 2 bits are ignored).

(5) 12-bpp (4K-color) mode

One pixel is represented by 12 bits, displayed in 4K (4,096) colors. In this mode, 4K discrete combinations are configured using 16 entries in each of the red, green, and blue look-up tables. Bits 11–8 in 12 bits of pixel data are used as an index to the red look-up table, while bits 7–4 and bits 3–0 are used as indices to the green and blue look-up tables, respectively.

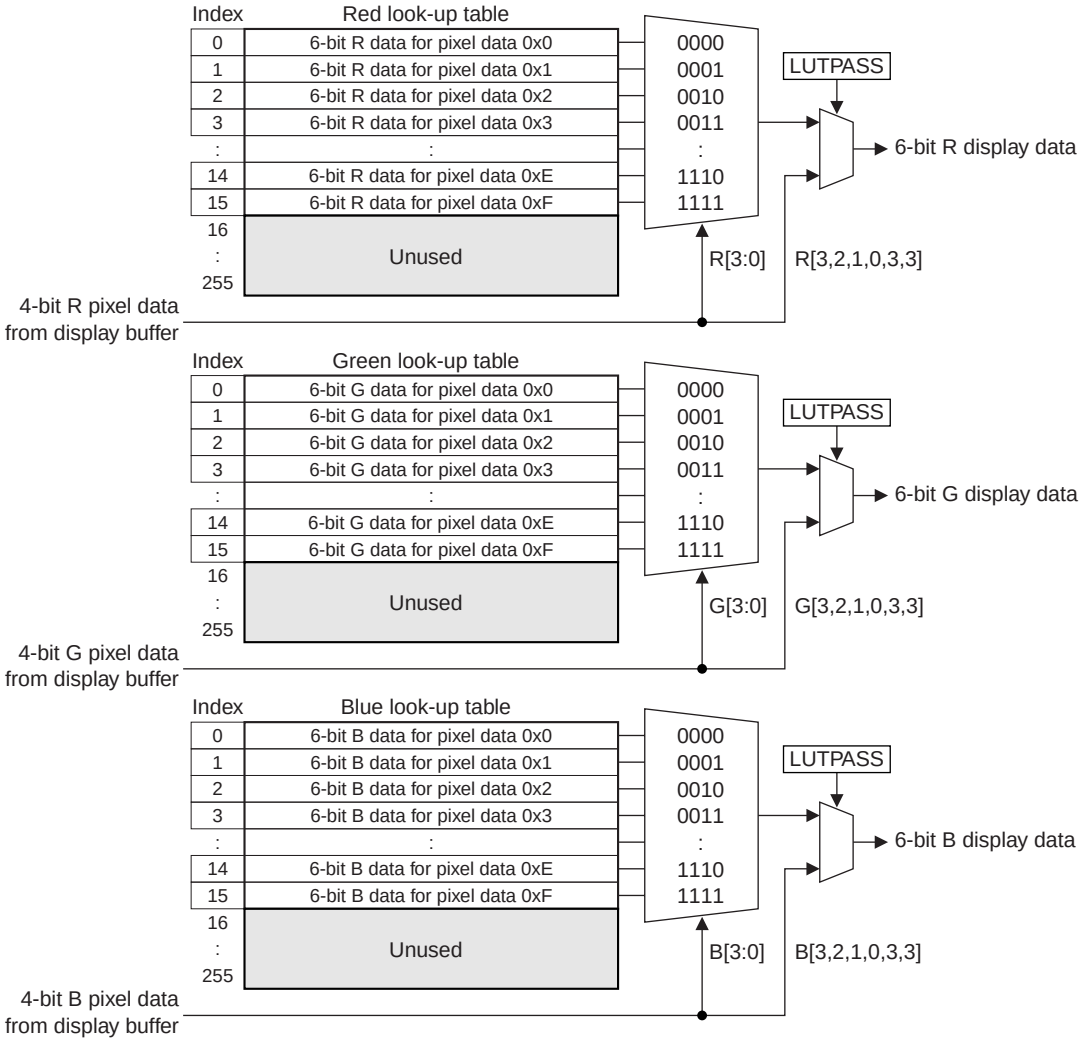


Figure VII.2.26 Look-up Table in 12-bpp (4K-Color) Mode

(6) 16-bpp (64K-color) mode

One pixel is represented by 16 bits, displayed in 64K (65536) colors. In this mode, 64K discrete combinations are configured using 32 entries in each of the red and blue look-up tables, and 64 entries in the green look-up table. Bits 15–11 in 16 bits of pixel data are used as an index to the red look-up table, while bits 10–5 and bits 4–0 are used as indices to the green and blue look-up tables, respectively.

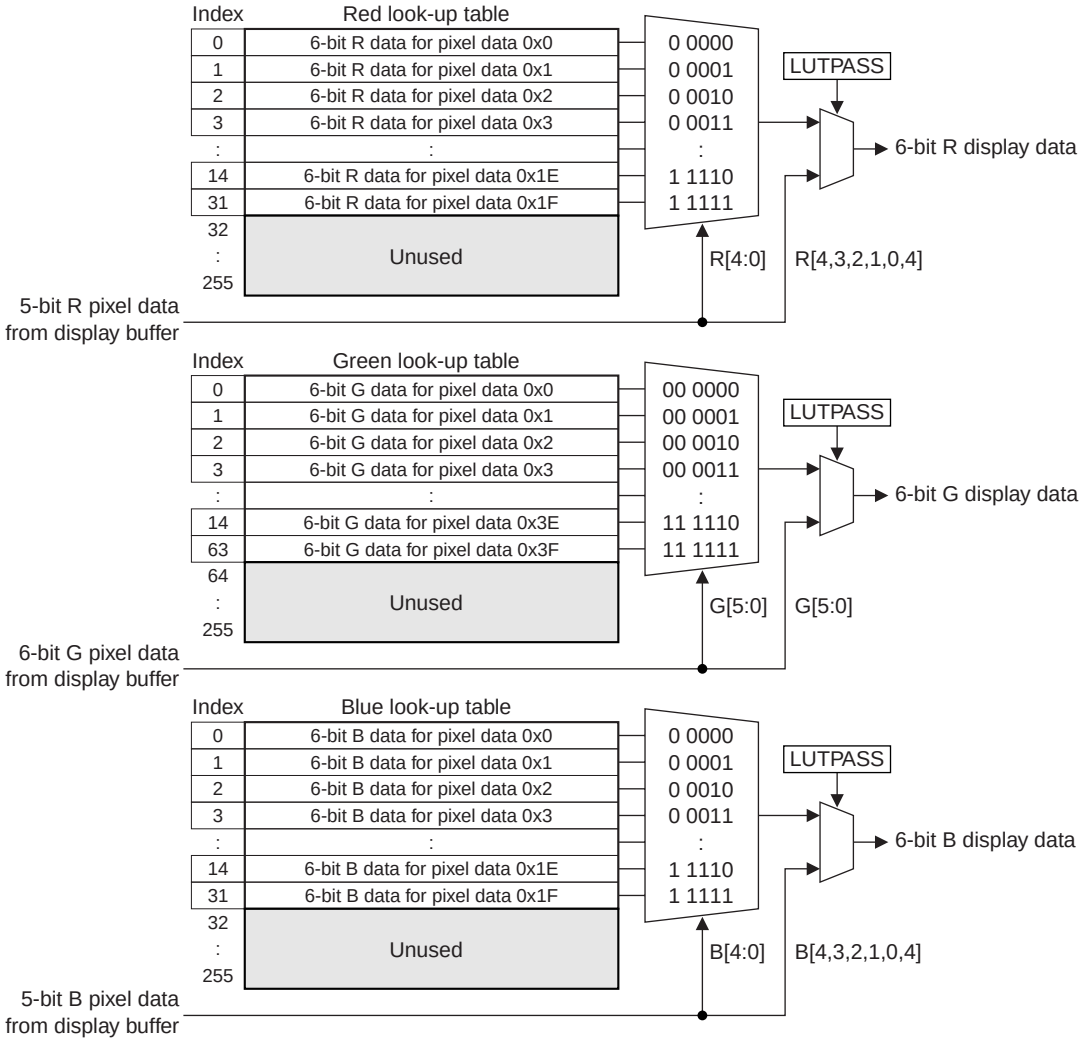


Figure VII.2.27 Look-up Table in 16-bpp (64K-Color) Mode

Setting data in the look-up tables

Use the look-up table data registers for writing and reading 6-bit gray/color data to/from the look-up tables. The look-up table data registers are mapped to addresses 0x380400 to 0x3807FF as shown in the figure below.

Address	LUT index	D[F:A]	D[9:8]	D[7:2]	D[1:0]
0x380400	0x00	G		R	
0x380402	(0)			B	
0x380404	0x01	G		R	
0x380406	(1)			B	
0x380408	0x02	G		R	
0x38040A	(2)			B	
:	:	:	:	:	:
0x3807F8	0xFE	G		R	
0x3807FA	(254)			B	
0x3807FC	0xFF	G		R	
0x3807FE	(255)			B	

Unused (reserved)

Figure VII.2.28 Look-up Table Map

Each word (32 bits) beginning with address 0x380400 corresponds to an index of the look-up tables; 0x380400–0x380403 corresponds to index 0 of the look-up tables and 0x3807FC–0x3807FF corresponds to index 255 (0xFF). The first half word in each word contains the green and red entries and the second half word contains the blue entry. Each of the color data can be written to an entry with an 8-bit data, however, the two low-order bits will be ignored.

When writing data to the look-up table, you can start writing from any LUT index, note, however, that green and red data (first half word) must be written prior to blue data, since green and red data are buffered and an RGB data for an index can be set to the look-up table when the corresponding blue data is written.
Therefore, blue data must be written even in a grayscale mode that uses only green data. In this case, dummy data (ordinary 0) should be written to the red and blue entries.

When area 6 is set to little endian mode, word (32-bit) access is recommended to read/write data from/to the LUT.

Frame Rates

The frame rate is calculated from the LCD panel's resolution, non-display period, and pixel clock frequency, as shown below.

$$\text{Frame rate} = \frac{f_{PCLK}}{(HDP + HNDP) \times (VDP + VNDP)}$$

f_{PCLK}: PCLK frequency (Hz)

This is the input clock frequency for the LCD controller derived by the LCDC clock generator. Refer to Section III-17, "S1C33L05 Clock System and Miscellaneous Registers", for setting the LCDC (PCLK) clock.

HDP: Horizontal display period

This is the LCD panel's horizontal resolution (in pixels). From the set value of HSIZE[6:0] (D[6:0]) / Horizontal panel size register (0x380042), the horizontal display period is calculated as follows:
Horizontal display period = (HSIZE[6:0] + 1) × 8 (Ts) where Ts = PCLK clock cycle

HNDP: Horizontal non-display period

This is a non-display period before the LCD panel starts displaying the next line after it has finished displaying all pixels in one line. Set a value in 8 pixel units in the HNDP[4:0] (D[4:0]) / Horizontal non-display period register (0x380040).
Horizontal non-display period = (HNDP[4:0] + 4) × 8 (Ts)

The value HDP described above plus HNDP comprises the number of PCLK clock cycles per one-line period (FPLINE pulse period).

VDP: Vertical display period

This is the LCD panel's vertical resolution (number of display lines). From the set value of the VSIZE[9:0] (D[9:0]) / Vertical panel size register (0x38004C), the vertical display period is calculated as follows:
Vertical display period = VSIZE[9:0] + 1 (lines)

VNDP: Vertical non-display period

This is a non-display period before the LCD panel starts displaying the next frame after it has finished displaying all display lines in one frame. Set this period based on the number of lines in the VNDP[5:0] (D[5:0]) / Vertical non-display period register (0x38004A).
Vertical non-display period = VNDP[5:0] (lines)

From the above parameters, we obtain the number of PCLK clock cycles required for the display of one frame, as determined by (HDP + HNDP) × (VDP + VNDP). The frame rate is calculated by dividing the PCLK clock frequency by this value.

Note: Most passive (STN) panels are tolerant of nearly any combination of HNDP and VNDP values, however panel specifications generally specify only a few lines of vertical non-display period. The LCDC is capable of generating a vertical non-display period of up to sixty-three lines. This amount of VNDP is far too great a non-display period and will likely degrade display quality. Similarly, setting a large HNDP value may cause a degradation in image quality. If possible the system should be designed such that VNDP values of 7 or less lines and HNDP values of 20 or less characters can be selected.

Other Settings

FPSHIFT mask

When a monochrome passive LCD panel is used, FPSHIFT (shift clock) can be turned on or off during the non-display period using MASK (DD) / LCDC mode register 1 (0x380202).
MASK = "1": Turned off
MASK = "0": Turned on (default)

MASK can only be set when COLOR (DE) / LCDC mode register 1 (0x380202) = "0" (monochrome panel). Otherwise, MASK has no effect.

MOD rate

The period during which the MOD signal is switched can be set using the MOD[5:0] (D[5:0]) / MOD rate register (0x380052).
MOD = "0x0": MOD signal switched at a period of the FPFRAME signal (default)
MOD = other than "0x0": Switched at a period of MOD + 1 FPLINE pulses

Repeating of the FRM pattern

This setup item is provided for EL panels. Whether the frame-rate modulation pattern is to be repeated every 0x40000 frames (counted by the internal frame counter) can be set using FRMRPT (D7) / LCDC mode register 0 (0x380200).
FRMRPT = "1": FRM pattern repeated (for EL panel)
FRMRPT = "0": FRM pattern not repeated (default)

Display Control

Controlling LCD Power Up/Down

The LCD controller is activated by setting LCLGEN (D7) / LCDC clock generator control register (0x300F34) to "1". Setting LCLGEN to "0" causes the LCD controller to stop operating, with the LCD signal output dropped low. For the LCD controller to start display correctly, the clock conditions, LCD-panel parameters and display data must be set after LCLGEN is set to "1".
If the power to the LCD panel is turned on or off while LCD signals are not being correctly output, the panel may be damaged. Therefore, the power to the LCD panel must be turned on only after the LCD controller starts controlling LCD signals. Use an I/O port to control the power to the LCD panel for this purpose. When LCD signals have no effect, disable the LCD power supply by controlling the port output; when LCD signals become effective, enable the LCD power supply using the port.

Following cold reset, the LCD controller is set in such a way that LCLGEN = "0" and power-save mode is on. Setting LCLGEN to "1" does not immediately cause the LCD panel to initiate a power-up sequence and start displaying data. The LCD panel is placed in power-save mode, with all LCD signal output pins fixed low.
To change the LCD controller from power-save mode back into normal mode, set the LPSAVE[1:0] (D[1:0]) / Status and power save register (0x380014) to "0b11". The LCD controller starts a power-up sequence from that point, and outputs LCD signals. Conversely, to change from normal mode to power-save mode, set LPSAVE to "0b00". The LCD controller starts a power-down sequence from that point, and drives the LCD signals low.
The LCD control registers and look-up tables can be accessed even in power-save mode.

The procedure for initializing the LCD at power-on is summarized below.
1. Configure the BCU, clocks, pins, and display memory area (refer to "System Settings").
2. Enable the LCD controller (LCLGEN = "1").
3. Set the LCD-panel parameters, display mode, and look-up tables (refer to "Setting the LCD Panel").
4. Enable the LCDC interrupt.
5. Write display data to the display memory.
6. Set the display start address (refer to "Setting the Display Start Address").
7. Place the LCD controller in normal mode (LPSAVE = "0b11").
8. The LCD controller starts outputting the LCD signals.
9. Wait time should be inserted depending on the LCD panel power source.
10. Control the port to turn the LCD panel power on.

The following is the power-down procedure.
1. Control the port to turn the LCD panel power off.
2. Wait time should be inserted depending on the LCD panel power source.
3. Place the LCD controller in power-save mode (LPSAVE = "0b00").
4. The LCD controller pulls LCD signals down to low.

Setting the Display Start Address

The LCD controller is initially set in such a way that data is displayed beginning with the start address of the display memory. The display memory address from which to start display can be changed as desired using the screen start address register (0x380210). The start address set in the screen start address register corresponds to the upper left edge of the LCD panel.
The value that should actually be set in this register is an offset address from the beginning of the VRAM area. When the internal VRAM is used, for example, the start address of the display memory is 0x0, rather than 0x3C0000. Depending on the VRAM used, the following boundary addresses must be specified.
When the internal VRAM is used: (Screen start address - IVRAM start address) / 2
When the external SDRAM is used: (Screen start address - SDRAM start address) / 32

When VRAMSEL (DF) / Screen start address register (0x380210) is set to "0", the internal VRAM is selected as the display memory. When VRAMSEL is set to "1", the external SDRAM is selected as the display memory.

Writing Display Data

The LCD controller may generate an interrupt at the beginning with the vertical non-display period after finishing each frame refresh sequence. Furthermore, the vertical display status flag VNDPF (D7) / Status and power save register (0x380014) is provided and is set to "1" if the display is in a vertical non-display period.
To eliminate screen flicker, display data, LUT data and the display buffer should be changed in a vertical non-display period by using this interrupt or the VNDPF flag.
For more information on the LCDC interrupt, refer to "LCDC Interrupt and DMA".

Inverting and Blanking the Display

The display can be blanked (the entire screen turned black or white) without rewriting the contents of the display memory. Setting the BLANK (D8) / LCDC mode register 1 (0x380202) to "1" causes the FPDAT signal to go low or high, blanking the display. Setting it to "0" turns the display back on. Whether the screen turns black or white is determined by the SWINV bit (D9/0x380202) described below.

Furthermore, the display can be inverted simply by manipulating a control bit. Setting the SWINV (D9) / LCDC mode register 1 (0x380202) to "1" inverts the display, and setting it to "0" returns the display to normal. This is accomplished by inverting the display data output from the look-up tables, rather than by inverting the pixel data in the display memory.
The screen can be made to blink using these operations. Make sure switching takes place within the vertical non-display period (VNDPF = "1").

LCDC Interrupt and DMA

Frame interrupt

When a frame refresh cycle (vertical display period) has finished, a vertical non-display period begins and the frame interrupt flag FRINTF (DF) / Status and power save register (0x380014) is set to "1". At the same time, the LCD controller outputs an interrupt signal to the ITC when the frame interrupt has been enabled by setting the FRINTEN (DF) / Frame interrupt register (0x380010) to "1". If the interrupt conditions set using the ITC registers are met, an interrupt to the CPU is generated. Occurrence of this interrupt source indicates that the display data can be written to the display memory. This interrupt can also be used to invoke IDMA, enabling data to be written to the display memory by means of a DMA transfer.

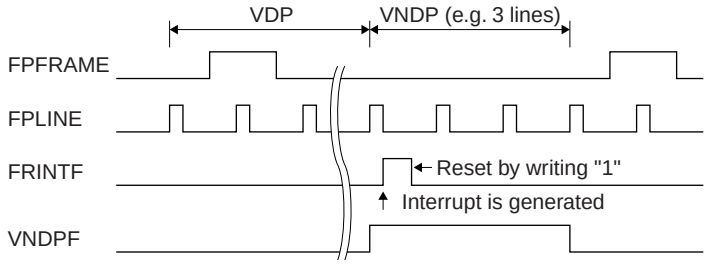


Figure VII.2.29 Frame Interrupt Timing

Once the FRINTF flag is set to "1", it is not reset until the software writes "1" to it. Therefore, when enabling the frame interrupt, write "1" to FRINTF before FRINTEN is set to "1" in order to avoid an unnecessary interrupt.
When not using the LCDC interrupt, set FRINTEN to "0".

Control registers of the interrupt controller

Table VII.2.17 shows the ITC's control registers for the LCDC interrupts.

Table VII.2.17 Control Registers of Interrupt Controller

Interrupt factor flag	Interrupt enable register	Interrupt priority register
FLCDCI (D5/0x402A9)	ELCDCI (D5/0x402A6)	PLCDCI[2:0] (D[6:4]/0x402A2)

When the interrupt factor described above occurs, the 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 LCDC interrupt factors can be used to invoke intelligent DMA (IDMA). This enables display data transfer from the data memory to the display memory to be performed by means of a DAM transfer.
The IDMA channel number set for the LCDC interrupt is 43.

The IDMA request and enable bits shown in Table VII.2.18 must be set to "1" for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

Table VII.2.18 Control Bits for IDMA Transfer

IDMA request bit	IDMA enable bit
RLCDCI (D5/0x402AC)	DELCDCI (D5/0x402AE)

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 a DMA transfer, refer to "IDMA (Intelligent DMA)".

Trap vectors

The trap-vector address for the LCDC interrupt is set to 0x0C00164 by default.
The base address of the trap table can be changed using the TTBR register (0x48134 to 0x48137).

Power Save

The LCD controller has two types of power-save modes. Use LPSAVE[1:0] (D[1:0]) / Status and power save register (0x380014) to set power-save modes.

Table VII.2.19 Settings of Power-Save Modes

LPSAVE1	LPSAVE0	Mode
1	1	Normal operation
1	0	Doze mode
0	1	Reserved
0	0	Power-save mode

Power-save mode

When the LCD controller enters this mode, all LCD signal output pins are dropped low, with the LCD panel placed in power-down mode. All operations of the LCD controller, other than accessing of its control registers and look-up tables are disabled.

The LCD controller is placed in power-save mode by setting LPSAVE to "00".

The LCD controller is taken out of power-save mode by setting LPSAVE to "11".

Doze mode

Doze mode is a power-save mode designed for use with built-in RAM type or self-refresh type LCD panels. These panels do no need to send data constantly in order to refresh the display of the same image. The LCD controller can be set in doze mode during this period. In doze mode, the FPDAT and FPSHIFT signals are fixed low so that no access to the display memory occurs. Although the power-saving effects are not as significant as in power-save mode, this mode helps reduce the current consumption in the LCD panel while keeping the display on.

Comparison of power-save modes

The differences between power-save modes are summarized in Table VII.2.20.

Table VII.2.20 Differences between Power-Save Modes

Item	LCDC disabled	Power-save mode	Doze mode	Normal
Accessing IO registers	Disabled	Enabled	Enabled	Enabled
Accessing look-up table	Disabled	Enabled	Enabled	Enabled
Accessing VRAM	Enabled	Enabled	Enabled	Enabled
Display	Inactive	Inactive	Active	Active
LCDC display-data-fetch operation	Inactive	Inactive	Inactive	Active
FPDAT[7:0], FPSHIFT signals	Low	Low	Low	Active
FPLINE, FPFRAME, DRDY signals	Low	Low	Active	Active

LCDC mode transition

When LPSAVE is changed, the LCDC executes a special sequence for several frames before the LCDC mode changes according to the LPSAVE value. The VRAMF (D6) / Status and power save register (0x380014) is provided to indicate whether the LCDC display-data-fetch operation is active or not. Read VRAMF to check whether the LCDC has entered the specified mode or not.

Host Interface Buffer and LCDC Cache

Host Interface Buffer and Access Timeout Error

The LCDC registers and look-up tables operate with the LCDC clock (generated by the LCDC clock generator). Therefore, the C33 STD core that uses the BCU clock does not access them directly. For correction of this problem, the LCDC host interface provides a handshake scheme using a data buffer.
When the CPU makes an attempt to write data to an LCDC register or look-up table, the data is temporarily written to the buffer using the BCU clock. Then the data is written to the LCDC register or look-up table using the LCDC clock. To avoid the next access from the CPU before the data has been written correctly, the LCDC host interface asserts the #WAIT signal input to the C33 STD core to extend the bus cycle when the CPU writes data until the buffer is empty due to writing compression.
While reading, the #WAIT signal is asserted to extend the CPU read cycle until the LCDC register or look-up table data is written to the buffer and is ready to read from the CPU.

If the LCDC register or look-up table does not respond for a certain period of time after the CPU accesses it, such as when the LCDC clock has not been supplied to the LCDC (LCLGEN = "0"), the embedded timeout counter will terminate the CPU bus cycle. Since the reading/writing by the CPU has failed in this case, the TOERRF flag (D5) / Status and power save register (0x380014) is provided to indicate that a timeout error has occurred. When this flag is "0", the reading/writing from/to the LCDC register or look-up table has completed normally. When the flag is "1", a timeout error has occurred. Once the flag is set to "1", it is not reset until the software writes "1" to the flag.
In addition to the TOERRF flag, the WRBUFF flag (D8) / Status and power save register (0x380014) is also provided to indicate the buffer status. When this flag is set to "1" after a timeout error has occurred, the previous write data remains in the buffer indicating that the writing has not been completed normally. When this flag is "0", the buffer is empty. In other words, the timeout error has occurred during reading.

LCDC Cache

The display memory (internal VRAM or an external SDRAM) is placed in a clock system other than the LCD controller. The display data in the display memory is input to the LCD controller via the IVRAM arbiter (for internal VRAM) or the bus arbiter (for external SDRAM) and temporarily stored in the LCDC cache before sending to the LCD interface.
The LCDC cache consists of two 32-byte asynchronous FIFOs (dual-port SOG-RAM) that provide a ping-pong mechanism. One FIFO is located in the LCDC clock system while the other FIFO is in the BCU clock system. In other words, one FIFO loads display data in synchronization with the display memory, while the other FIFO outputs data to the LCD interface. Two FIFOs alternate their functions every time the 32-byte data process is finished.
When the VRAM side FIFO becomes full, data read from the VRAM is suspended until the other FIFO becomes empty. When the LCD I/F side FIFO becomes empty, it is switched to the VRAM side and requests the following display data to the IVRAM arbiter or bus arbiter. The LCD I/F continues reading data from the FIFO that is filled with the next display data.

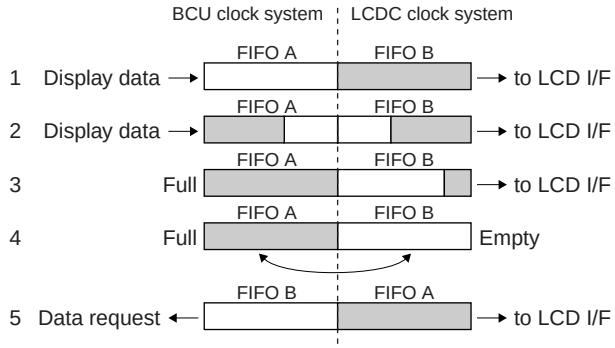


Figure VII.2.30 Operation of the LCDC Cache

If the VRAM side FIFO is not full when the LCD I/F side FIFO becomes empty, the LCDC stops the display sequence until the VRAM side FIFO becomes full. When this condition occurs, the FIFOEF flag (D9) / Status and power save register (0x380014) is set to "1". This flag is not reset until the software writes "1" to it, when it is set to "1".
This error may cause flicker on the display. If this error occurs frequently, it is necessary to increase the system performance. For example, increase halt2run_clk vs. lcdc_clk ratio, increase horizontal non-display period or use the FIFO flexible-burst read function.
The FIFO flexible-burst read function is enabled by setting the FIFOHP (D3) / Status and power save register (0x380014) to "1". In the default setting (FIFOHP = "0"), the display data is loaded into the FIFO by a 16-burst read operation. When the flexible-burst read function is enabled, a 16-, 32-, or 48-burst read is performed on a random basis.

Furthermore, if the external SDRAM cannot be accessed due to some reason, such as when the bus clock generator is disabled, the bus arbiter is disabled, the SDRAM controller is not initialized, etc., a FIFO empty error occurs and FIFOEF (D9) / Status and power save register (0x380014) is set to "1".

I/O Memory of LCD Controller

Table VII.2.21 shows the control bits of the LCD controller.

Table VII.2.21 Control Bits of LCD Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC interrupt priority register	**00402A2 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PLCDCI2**	LCD controller interrupt level	0 to 7	X	R/W	
		D5	**PLCDCI1**			X		
		D4	**PLCDCI0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	–	Interrupt level of extended	0 to 7	X	R/W	
		D1	–	function (reserved)		X		
		D0	–			X		
LCDC, USB, SPI interrupt enable register	**00402A6 (B)**	D7	**ESPII**	SPI interrupts	1 Enabled 0 Disabled	0	R/W	
		D6	**EUSBI**	USB interrupts		0	R/W	
		D5	**ELCDCI**	LCDC interrupts		0	R/W	
		D4	**–**	Extended interrupt (reserved)		0	R/W	Do not write 1.
		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	
LCDC, USB, SPI interrupt factor flag register	**00402A9 (B)**	D7	**FSPII**	SPI interrupts	1 Factor is generated 0 No factor is generated	X	R/W	
		D6	**FUSBI**	USB interrupts		X	R/W	
		D5	**FLCDCI**	LCDC interrupts		X	R/W	
		D4	**–**	Extended interrupt (reserved)		X	R/W	Do not write 1.
		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	
LCDC, USB, SPI IDMA request register	**00402AC (B)**	D7	**RSPII**	SPI interrupts	1 IDMA request 0 Interrupt request	0	R/W	
		D6	**RUSBI**	USB interrupts		0	R/W	
		D5	**RLCDCI**	LCDC interrupts		0	R/W	
		D4	**–**	Extended interrupt (reserved)		0	R/W	Do not write 1.
		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	
LCDC, USB, SPI IDMA enable register	**00402AE (B)**	D7	**DESPII**	SPI interrupts	1 IDMA enabled 0 IDMA disabled	0	R/W	
		D6	**DEUSBI**	USB interrupts		0	R/W	
		D5	**DELCDCI**	LCDC interrupts		0	R/W	
		D4	**–**	Extended interrupt (reserved)		0	R/W	Do not write 1.
		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	
PB0–PB3 port function extension register	**0300F62 (B)**	D7 D6	**FPB31** **FPB30**	PB3 port extended function	FPB3[1:0]: 1 * reserved; 0 1 FPDAT3; 0 0 PB3	0 0	R/W	
		D5 D4	**FPB21** **FPB20**	PB2 port extended function	FPB2[1:0]: 1 * reserved; 0 1 FPDAT2; 0 0 PB2	0 0	R/W	
		D3 D2	**FPB11** **FPB10**	PB1 port extended function	FPB1[1:0]: 1 * reserved; 0 1 FPDAT1; 0 0 PB1	0 0	R/W	
		D1 D0	**FPB01** **FPB00**	PB0 port extended function	FPB0[1:0]: 1 * reserved; 0 1 FPDAT0; 0 0 PB0	0 0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PB4–PB7 port function extension register	**0300F63 (B)**	D7	**FPB71**	PB7 port extended function	FPB7[1:0]: 1 * reserved; 0 1 FPDAT7; 0 0 PB7	0	R/W	
		D6	**FPB70**			0		
		D5	**FPB61**	PB6 port extended function	FPB6[1:0]: 1 * reserved; 0 1 FPDAT6; 0 0 PB6	0	R/W	
		D4	**FPB60**			0		
		D3	**FPB51**	PB5 port extended function	FPB5[1:0]: 1 * reserved; 0 1 FPDAT5; 0 0 PB5	0	R/W	
		D2	**FPB50**			0		
		D1	**FPB41**	PB4 port extended function	FPB4[1:0]: 1 * reserved; 0 1 FPDAT4; 0 0 PB4	0	R/W	
		D0	**FPB40**			0		
PC0–PC3 port function extension register	**0300F64 (B)**	D7	**FPC31**	PC3 port extended function	FPC3[1:0]: 1 * reserved; 0 1 DRDY; 0 0 PC3	0	R/W	
		D6	**FPC30**			0		
		D5	**FPC21**	PC2 port extended function	FPC2[1:0]: 1 * reserved; 0 1 FPSHIFT; 0 0 PC2	0	R/W	
		D4	**FPC20**			0		
		D3	**FPC11**	PC1 port extended function	FPC1[1:0]: 1 * reserved; 0 1 FPLINE; 0 0 PC1	0	R/W	
		D2	**FPC10**			0		
		D1	**FPC01**	PC0 port extended function	FPC0[1:0]: 1 * reserved; 0 1 FPFRAME; 0 0 PC0	0	R/W	
		D0	**FPC00**			0		
Sub system clock select register	**0300F32 (B)**	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**SUBCKSEL**	Sub system source clock select	1 PLL clock; 0 OSC3 clock	0	R/W	
LCDC clock generator control register	**0300F34 (B)**	D7	**LCLGEN**	LCDC clock generator enable	1 Enabled; 0 Disabled	0	R/W	
		D6–4	**–**	reserved	–	–	–	0 when being read.
		D3	**LCLGDT3**	LCDC clock setup (Sub system clock division ratio)	LCDC clock frequency = subsys_clk / (LCLGDT[3:0] + 1) [Hz]	0	R/W	
		D2	**LCLGDT2**			0		
		D1	**LCLGDT1**			0		
		D0	**LCLGDT0**			0		
Frame interrupt register	**0380010 (HW)**	DF	**FRINTEN**	Frame interrupt enable	1 Enabled; 0 Disabled	0	R/W	
		DE–0	**–**	reserved	–	–	–	0 when being read.
Status and power save register	**0380014 (HW)**	DF	**FRINTF**	Frame interrupt flag	1 Generated; 0 Not generated	0	R/W	Reset by writing 1.
		DE–A	**–**	reserved	–	–	–	0 when being read.
		D9	**FIFOEF**	Display FIFO empty flag	1 Empty; 0 Not empty	0	R/W	Reset by writing 1.
		D8	**WRBUFF**	LUT/IO write buffer status	1 Full; 0 Empty	0	R	
		D7	**VNDPF**	Vertical display status	1 VNDP; 0 VDP	1	R	
		D6	**VRAMF**	VRAM status	1 Idle; 0 Busy	1	R	
		D5	**TOERRF**	CPU access timeout error flag	1 Error; 0 Normal	0	R/W	Reset by writing 1.
		D4	**–**	reserved	–	–	–	0 when being read.
		D3	**FIFOHP**	Display FIFO flexible-burst read	1 Flex-burst read; 0 16 burst read	0	R/W	
		D2	**–**	reserved	–	–	–	0 when being read.
		D1	**LPSAVE1**	Power save mode	LPSAVE[1:0]: 1 1 Normal operation; 1 0 Doze mode; 0 1 reserved; 0 0 Power save mode	0	R/W	
		D0	**LPSAVE0**			0		
Horizontal non-display period register	**0380040 (HW)**	DF–5	**–**	reserved	–	–	–	0 when being read.
		D4	**HNDP4**	Horizontal non-display period HNDP4 = MSB HNDP0 = LSB	Non-display period (pixels) / 8 - 4	0	R/W	
		D3	**HNDP3**			0		
		D2	**HNDP2**			0		
		D1	**HNDP1**			0		
		D0	**HNDP0**			0		
Horizontal panel size register	**0380042 (HW)**	DF–7	**–**	reserved	–	–	–	0 when being read.
		D6	**HSIZE6**	Horizontal panel size HSIZE6 = MSB HSIZE0 = LSB	H panel resolution (pixels) / 8 - 1	0	R/W	
		D5	**HSIZE5**			0		
		D4	**HSIZE4**			0		
		D3	**HSIZE3**			0		
		D2	**HSIZE2**			0		
		D1	**HSIZE1**			0		
		D0	**HSIZE0**			0		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vertical non-display period register	**038004A (HW)**	DF–6	**–**	reserved	–	–	–	0 when being read.
		D5	**VNDP5**	Vertical non-display period	Non-display period (lines)	0	R/W	
		D4	**VNDP4**	VNDP5 = MSB		0		
		D3	**VNDP3**	VNDP0 = LSB		0		
		D2	**VNDP2**			0		
		D1	**VNDP1**			0		
		D0	**VNDP0**			0		
Vertical panel size register	**038004C (HW)**	DF–A	**–**	reserved	–	–	–	0 when being read.
		D9	**VSIZE9**	Vertical panel size	Vertical panel resolution (lines) -1	0	R/W	
		D8	**VSIZE8**	VSIZE9 = MSB		0		
		D7	**VSIZE7**	VSIZE0 = LSB		0		
		D6	**VSIZE6**			0		
		D5	**VSIZE5**			0		
		D4	**VSIZE4**			0		
		D3	**VSIZE3**			0		
		D2	**VSIZE2**			0		
		D1	**VSIZE1**			0		
		D0	**VSIZE0**			0		
MOD rate register	**0380052 (HW)**	DF–6	**–**	reserved	–	–	–	0 when being read.
		D5	**MOD5**	LCD MOD rate	0x0 to 0x3F	0	R/W	
		D4	**MOD4**	MOD5 = MSB		0		
		D3	**MOD3**	MOD0 = LSB		0		
		D2	**MOD2**			0		
		D1	**MOD1**			0		
		D0	**MOD0**			0		
LCDC mode register 0	**0380200 (HW)**	DF–8	**–**	reserved	–	–	–	0 when being read.
		D7	**FRMRPT**	Frame repeat for EL panel	1 Repeated / 0 Not repeated	0	R/W	
		D6	**DITHEN**	Dither mode enable	1 Enabled / 0 Disabled	0	R/W	
		D5	**–**	reserved	–	–	–	0 when being read.
		D4	**LUTPASS**	LUT bypass mode	1 Bypassed / 0 Used	0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**BPP2**	Bit-per-pixel select	BPP[2:0] = 1 1 * : reserved	0	R/W	
		D1	**BPP1**		1 0 1 : 16 bpp (64Kc/64gr)	0		
		D0	**BPP0**		1 0 0 : 12 bpp (4Kc/16gr)			
					0 1 1 : 8 bpp (256c/64gr)			
					0 1 0 : 4 bpp (16c/16gr)			
					0 0 1 : 2 bpp (4c/4gr)			
					0 0 0 : 1 bpp (2c/2gr)			
LCDC mode register 1	**0380202 (HW)**	DF	**–**	reserved	–	–	–	0 when being read.
		DE	**COLOR**	Color/mono select	1 Color / 0 Mono	0	R/W	
		DD	**MASK**	FPSHIFT mask enable	1 Enabled / 0 Disabled	0	R/W	
		DC	**–**	reserved	–	–	–	0 when being read.
		DB	**DWT1**	LCD panel data width	DWT[1:0] = 1 1 : 8-bit (format2)	0	R/W	
		DA	**DWT0**		1 0 : reserved	0		
					0 1 : 8-bit (format1)			
					0 0 : 4-bit			
		D9	**SWINV**	Software video invert	1 Inverted / 0 Normal	0	R/W	
		D8	**BLANK**	Display blank enable	1 Blank / 0 Normal	0	R/W	
		D7–0	**–**	reserved	–	–	–	0 when being read.
Screen start address register	**0380210 (HW)**	DF	**VRAMSEL**	VRAM select	1 External / 0 Internal	0	R/W	
		DE	**SADDR14**	Screen start address	0x0 to 0x7FFF	0	R/W	
		DD	**SADDR13**	SADDR14 = MSB		0		
		DC	**SADDR12**	SADDR0 = LSB		0		
		DB	**SADDR11**			0		
		DA	**SADDR10**			0		
		D9	**SADDR9**			0		
		D8	**SADDR8**			0		
		D7	**SADDR7**			0		
		D6	**SADDR6**			0		
		D5	**SADDR5**			0		
		D4	**SADDR4**			0		
		D3	**SADDR3**			0		
		D2	**SADDR2**			0		
		D1	**SADDR1**			0		
		D0	**SADDR0**			0		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Look-up table R/G data register (Address 0)	**0380400 (HW)**	DF	**LUTG05**	Look-up table G data	0x0 to 0x3F	X	R/W	0 when being read.
		DE	**LUTG04**	LUTG05 = MSB		X		
		DD	**LUTG03**	LUTG00 = LSB		X		
		DC	**LUTG02**			X		
		DB	**LUTG01**			X		
		DA	**LUTG00**			X		
		D9–8	**–**	reserved	–	–	–	0 when being read.
		D7	**LUTR05**	Look-up table R data	0x0 to 0x3F	X	R/W	
		D6	**LUTR04**	LUTR05 = MSB		X		
		D5	**LUTR03**	LUTR00 = LSB		X		
		D4	**LUTR02**			X		
		D3	**LUTR01**			X		
		D2	**LUTR00**			X		
		D1–0	**–**	reserved	–	–	–	0 when being read.
Look-up table B data register (Address 0)	**0380402 (HW)**	DF–8	**–**	reserved	–	–	–	0 when being read.
		D7	**LUTB05**	Look-up table B data	0x0 to 0x3F	X	R/W	
		D6	**LUTB04**	LUTB05 = MSB		X		
		D5	**LUTB03**	LUTB00 = LSB		X		
		D4	**LUTB02**			X		
		D3	**LUTB01**			X		
		D2	**LUTB00**			X		
		D1–0	**–**	reserved	–	–	–	0 when being read.

:

– – – 0x380404 ... 0x3807FA – – –

:

Look-up table R/G data register (Address FF)	03807FC (HW)	DF	**LUTGFF5**	Look-up table G data	0x0 to 0x3F	X	R/W	0 when being read.
		DE	**LUTGFF4**	LUTGFF5 = MSB		X		
		DD	**LUTGFF3**	LUTGFF0 = LSB		X		
		DC	**LUTGFF2**			X		
		DB	**LUTGFF1**			X		
		DA	**LUTGFF0**			X		
		D9–8	**–**	reserved	–	–	–	0 when being read.
		D7	**LUTRFF5**	Look-up table R data	0x0 to 0x3F	X	R/W	
		D6	**LUTRFF4**	LUTRFF5 = MSB		X		
		D5	**LUTRFF3**	LUTRFF0 = LSB		X		
		D4	**LUTRFF2**			X		
		D3	**LUTRFF1**			X		
		D2	**LUTRFF0**			X		
		D1–0	**–**	reserved	–	–	–	0 when being read.
Look-up table B data register (Address FF)	**03807FE (HW)**	DF–8	**–**	reserved	–	–	–	0 when being read.
		D7	**LUTBFF5**	Look-up table B data	0x0 to 0x3F	X	R/W	
		D6	**LUTBFF4**	LUTBFF5 = MSB		X		
		D5	**LUTBFF3**	LUTBFF0 = LSB		X		
		D4	**LUTBFF2**			X		
		D3	**LUTBFF1**			X		
		D2	**LUTBFF0**			X		
		D1–0	**–**	reserved	–	–	–	0 when being read.

FPB01–FPB00: PB0 port extended function (D[1:0]) / PB0–PB3 port function extension register (0x300F62)
FPB11–FPB10: PB1 port extended function (D[3:2]) / PB0–PB3 port function extension register (0x300F62)
FPB21–FPB20: PB2 port extended function (D[5:4]) / PB0–PB3 port function extension register (0x300F62)
FPB31–FPB30: PB3 port extended function (D[7:6]) / PB0–PB3 port function extension register (0x300F62)
FPB41–FPB40: PB4 port extended function (D[1:0]) / PB4–PB7 port function extension register (0x300F63)
FPB51–FPB50: PB5 port extended function (D[3:2]) / PB4–PB7 port function extension register (0x300F63)
FPB61–FPB60: PB6 port extended function (D[5:4]) / PB4–PB7 port function extension register (0x300F63)
FPB71–FPB70: PB7 port extended function (D[7:6]) / PB4–PB7 port function extension register (0x300F63)
FPC01–FPC00: PC0 port extended function (D[1:0]) / PC0–PC3 port function extension register (0x300F64)
FPC11–FPC10: PC1 port extended function (D[3:2]) / PC0–PC3 port function extension register (0x300F64)
FPC21–FPC20: PC2 port extended function (D[5:4]) / PC0–PC3 port function extension register (0x300F64)
FPC31–FPC30: PC3 port extended function (D[7:6]) / PC0–PC3 port function extension register (0x300F64)

Switches the PB and PC port functions.

Table VII.2.22 PB and PC Port Extended Functions

Function extension bit	FPxx[1:0] = "00"	FPxx[1:0] = "01"	FPxx[1:0] = "10"	FPxx[1:0] = "11"
FPB0[1:0]	PB0	FPDAT0	–	–
FPB1[1:0]	PB1	FPDAT1	–	–
FPB2[1:0]	PB2	FPDAT2	–	–
FPB3[1:0]	PB3	FPDAT3	–	–
FPB4[1:0]	PB4	FPDAT4	–	–
FPB5[1:0]	PB5	FPDAT5	–	–
FPB6[1:0]	PB6	FPDAT6	–	–
FPB7[1:0]	PB7	FPDAT7	–	–
FPC0[1:0]	PC0	FPFRAME	–	–
FPC1[1:0]	PC1	FPLINE	–	–
FPC2[1:0]	PC2	FPSHIFT	–	–
FPC3[1:0]	PC3	DRDY	–	–

Set FPBx[1:0], and FPCx[1:0] to "01" when using the LCD controller. The PBx and PCx pins are configured as the LCD interface pins.
At initial reset, FPB and FPC are set to "00".

SUBCKSEL: Sub system source clock select (D0) /Sub system clock select register (0x300F32)

Selects the source clock for the bus clock and LCDC clock generators.

Write "1": PLL output clock
Write "0": OSC3 clock
Read: Valid

When "1" is written to SUBCKSEL, the PLL output clock is selected as the source clock (subsys_clk) for the bus clock and LCDC clock generators in the extended block. When "0" is written, the OSC3 clock is selected.
At initial reset, SUBCKSEL is set to "0" (OSC3 clock).

LCLGEN: LCDC clock generator enable (D7) / LCDC clock generator control register (0x300F34)

Enables/disables the LCDC clock generator.

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

When "1" is written to LCLGEN, the source clock supply is enabled and the LCDC clock generator starts outputting the lcdc_clk clock for the LCD controller. When "0" is written, the LCDC clock generator stops outputting the lcdc_clk clock.
At initial reset, LCLGEN is set to "0" (disabled).

LCLGDT3–LCLGDT0: LCDC clock setup (D[3:0]) / LCDC clock generator control register (0x300F34)

Configures the frequency divider in the LCDC clock generator to set the lcdc_clk clock frequency.

Table VII.2.23 Setting the Division Ratio for LCDC Clock

LCLGDT3	LCLGDT2	LCLGDT1	LCLGDT0	LCLG source clock
1	1	1	1	subsys_clk / 16
1	1	1	0	subsys_clk / 15
1	1	0	1	subsys_clk / 14
1	1	0	0	subsys_clk / 13
1	0	1	1	subsys_clk / 12
1	0	1	0	subsys_clk / 11
1	0	0	1	subsys_clk / 10
1	0	0	0	subsys_clk / 9
0	1	1	1	subsys_clk / 8
0	1	1	0	subsys_clk / 7
0	1	0	1	subsys_clk / 6
0	1	0	0	subsys_clk / 5
0	0	1	1	subsys_clk / 4
0	0	1	0	subsys_clk / 3
0	0	0	1	subsys_clk / 2
0	0	0	0	subsys_clk

At initial reset, LCLGDT is set to "0" (subsys_clk).

FRINTEN: Frame interrupt enable (DF) / Frame interrupt register (0x380010)

Enables the LCDC frame interrupt.

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

When using the frame interrupt, set FRINTEN to "1". The output of the interrupt signal to the ITC is enabled. When this bit is set to "0", the LCDC interrupt will not be generated.
At initial reset, FRINTEN is set to "0" (disabled).

FRINTF: Frame interrupt flag (DF) / Status and power save register (0x380014)

Indicates the frame interrupt status.

When read

Read "1": Interrupt is generated
Read "0": Interrupt is not generated

When written

Write "1": Flag is reset
Write "0": Invalid

FRINTF is set to "1" when a vertical non-display period begins. If FRINTEN is set to "1" to enable the frame interrupt, the interrupt signal is asserted and the LCDC interrupt factor flag in the ITC is set to "1". This flag can only be reset by writing "1" to it.
At initial reset, FRINTF is set to "0" (not generated).

FIFOEF: Display FIFO empty flag (D9) / Status and power save register (0x380014)

Indicates whether the display FIFO is empty or not.

When read

Read "1": Empty
Read "0": Not empty

When written

Write "1": Flag is reset
Write "0": Invalid

FIFOEF is set to "1" if the VRAM side FIFO is not full when the LCD I/F side FIFO becomes empty. In this case the LCDC stops display sequence until the VRAM side FIFO becomes full. This flag can only be reset by writing "1" to it.
At initial reset, FIFOEF is set to "0" (not empty).

WRBUFF: LUT/IO write buffer status (D8) / Status and power save register (0x380014)

Indicates the LCDC host interface buffer status.

Read "1": Full
Read "0": Empty
Write: Invalid

When this flag has been set to "1" after a timeout error has occurred, the previous write data remains in the buffer indicating that the writing has not been completed normally. When this flag is "0", the buffer is empty. In other words, the timeout error has occurred during reading.
At initial reset, WRBUFF is set to "0" (empty).

VNDPF: Vertical display status (D7) / Status and power save register (0x380014)

Indicates whether the LCD panel is in a vertical non-display period or not.

Read "1": Vertical non-display period
Read "0": Vertical display period
Write: Invalid

VNDPF is set to "1" during a vertical non-display period, and set to "0" during a vertical display period. When images must be switched without causing the screen to flicker, it is possible to switch within a vertical non-display period by reading this bit.
At initial reset, VNDPF is set to "1" (vertical non-display period).

VRAMF: VRAM status (D6) / Status and power save register (0x380014)

Indicates whether the VRAM is being accessed or not.

Read "1": Idle
Read "0": Busy
Write: Invalid

VRAMF is set to "0" while the VRAM is being accessed and the FIFO is loading display data from the VRAM. It is set to "1" when the FIFO becomes full and the VRAM idles.
At initial reset, VRAMF is set to "1" (idle).

TOERRF: CPU access timeout error flag (D5) / Status and power save register (0x380014)

Indicates whether a timeout error has occurred or not.

When read

Read "1": Error
Read "0": Normal

When written

Write "1": Flag is reset
Write "0": Invalid

TOERRF is set to "1" if a timeout error has occurred.
This error occurs when the LCDC register or look-up table does not respond to the access from the CPU for a certain period of time. (The LCDC clock may not be supplied to the LCDC.)
This flag can only be reset by writing "1" to it.
At initial reset, TOERRF is set to "0" (normal).

FIFOHP: Display FIFO flexible-burst read (D3) / Status and power save register (0x380014)

Enables the FIFO flexible-burst read function.

Write "1": Flexible-burst read
Write "0": 16-burst read
Read: Valid

When FIFOHP is set to "1", the FIFO flexible-burst read function is enabled and the display data read performance will be increased. When this function is enabled, a 16-, 32-, or 48-burst read is performed on a random basis. When FIFOHP is set to "0", the display data is loaded into the FIFO by a 16-burst read operation.
At initial reset, FIFOHP is set to "0" (16-burst read).

LPSAVE[1:0]: Power-save mode (D[1:0]) / Status and power save register (0x380014)

Selects power-save mode.

Table VII.2.24 Settings of Power-Save Modes

LPSAVE1	LPSAVE0	Mode
1	1	Normal operation
1	0	Doze mode
0	1	Reserved
0	0	Power-save mode

The LCD controller is placed in power-save mode by setting LPSAVE to "00". In this mode, all LCD signal output pins are dropped low and all operations of the LCD controller, other than accessing of its control registers and look-up tables are disabled. The LCD controller is taken out of power-save mode by setting LPSAVE to "11".

Doze mode is a power-save mode designed for use with built-in RAM type or self-refresh type LCD panels. In doze mode, the FPDAT and FPSHIFT signals are fixed low so that no access to the display memory occurs. Although the power-saving effects are not as significant as in power-save mode, this mode helps reduce the current consumption in the LCD panel while keeping the display on.
At initial reset, LPSAVE is set to "00" (power-save mode).

HNDP[4:0]: Horizontal non-display period (D[4:0]) / Horizontal non-display period register (0x380040)

Sets the horizontal non-display period in 8-pixel units. Set the value obtained using the equation below.

$$HNDP[4:0] = \frac{\text{Horizontal non-display period (in pixels)}}{8} - 4$$

At initial reset, HNDP is set to "0x0".

HSIZE[6:0]: Horizontal panel size (D[6:0]) / Horizontal panel size register (0x380042)

Sets the horizontal resolution of the LCD panel in 8-pixel units. Set the value obtained using the equation below.

$$HSIZE[6:0] = \frac{\text{Horizontal resolution (in pixels)}}{8} - 1$$

For an LCD panel with a horizontal resolution of 320 dots, for example, set 39 (= 0x27) in HSIZE. Do not set any value less than 1 in this register.
At initial reset, HSIZE is set to "0x0".

VNDP[5:0]: Vertical non-display period (D[5:0]) / Vertical non-display period register (0x38004A)

Sets the vertical non-display period in units of lines. Set the value obtained using the equation below.

VNDP[5:0] = Vertical non-display period (in lines)

At initial reset, VNDP is set to "0x0".

VSIZE[9:0]: Vertical panel size (D[9:0]) / Vertical panel size register (0x38004C)

Sets the vertical resolution of the LCD panel in units of lines. Set the value obtained using the equation below.

VSIZE[9:0] = Vertical resolution (in lines) - 1

For an LCD panel with a vertical resolution of 240 lines, for example, set 239 (= 0xEF) in VSIZE.
At initial reset, VSIZE is set to "0x0".

MOD[5:0]: MOD rate (D[5:0]) / MOD rate register (0x380052)

Sets the cycle time at which to switch the MOD signal. When this register is 0x0, the MOD signal switches at the cycle time of the FPFRAME signal. If another period is desired, set the FPLINE pulse-count value.
At initial reset, MOD is set to "0x0" (FPFRAME period).

FRMRPT: Frame repeat for EL panel (D7) / LCDC mode register 0 (0x380200)

Selects whether to repeat the frame-rate modulation pattern (effective only for EL panels).

Write "1": Repeated
Write "0": Not repeated
Read: Valid

When FRMRPT is set to "1", the internal 19-bit frame counter is enabled and starts counting the number of frames. Each time this counter overflows (0x40000 → 0), the frame-rate modulation pattern is repeated. When FRMRPT is set to "0", the counter is disabled and the frame-rate modulation pattern is not repeated.
At initial reset, FRMRPT is set to "0" (not repeated).

DITHEN: Dither mode enable (D6) / LCDC mode register 0 (0x380200)

Enables or disables dither mode.

Write "1": Dither mode
Write "0": Normal mode
Read: Valid

When DITHEN is set to "1", a maximum of 256K colors (2^{18}) or 64 gray shades in 1/2/4/8/12/16 bpp mode will be generated if LUT is used. Setting DITHEN to "0" allows use of a maximum of 4096 colors (2^{12}) or 16 gray shades.
At initial reset, DITHEN is set to "0" (normal mode).

LUTPASS: LUT bypass mode (D4) / LCDC mode register 0 (0x380200)

Selects whether the look-up table is bypassed in 1-, 2-, 4-, 8-, 12- or 16-bpp mode.

Write "1": Bypassed
Write "0": Used
Read: Valid

When LUTPASS is set to "1" in 1-, 2-, 4-, 8-, 12- or 16-bpp mode, the look-up table is bypassed and the pixel data in the display memory represents the display data to be sent to the LCD panel. When LUTPASS is set to "0", the look-up table is used to convert pixel data in the display memory into LCD interface data.
At initial reset, LUTPASS is set to "0" (used).

BPP[2:0]: Bit-per-pixel select (D[2:0]) / LCDC mode register 0 (0x380200)

Selects display mode (bpp mode). The contents of selection, including that of COLOR, are listed in Table VII.2.25.

Table VII.2.25 Specification of Display Modes

BPP2	BPP1	BPP0	Display mode	
			Monochrome (COLOR = "0")	Color (COLOR = "1")
1	1	*	Reserved	Reserved
1	0	1	16 bpp, 64 gray levels	16 bpp, 64K colors
1	0	0	12 bpp, 16 gray levels	12 bpp, 4K colors
0	1	1	8 bpp, 64 gray levels	8 bpp, 256 colors
0	1	0	4 bpp, 16 gray levels	4 bpp, 16 colors
0	0	1	2 bpp, 4 gray levels	2 bpp, 4 colors
0	0	0	1 bpp, 2 gray levels	1 bpp, 2 colors

At initial reset, BPP is set to "000" (1-bpp mode).

COLOR: Color/monochrome select (DE) / LCDC mode register 1 (0x380202)

Selects the type of connected LCD panel (color or monochrome).

Write "1": Color panel
Write "0": Monochrome panel
Read: Valid

Setting COLOR to "1" selects a color panel drive method, and setting it to "0" selects a monochrome panel drive method.
At initial reset, COLOR is set to "0" (monochrome panel).

MASK: FPSHIFT mask enable (DD) / LCDC mode register 1 (0x380202)

Enables the FPSHIFT mask (effective only for monochrome LCD panels).

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

When MASK is set to "1", the FPSHIFT signal is masked and is not output during the non-display period. When MASK is set to "0", the FPSHIFT signal is output even during the non-display period. This setting is effective only for monochrome LCD panels (COLOR = "0"). When a color LCD panel is used, the FPSHIFT signal is always masked regardless of the setting of this bit.
At initial reset, MASK is set to "0" (disabled).

DWT[1:0]: LCD panel data width (D[B:A]) / LCDC mode register 1 (0x380202)

Selects the LCD panel's data width and format. The contents of selection, including that of COLOR, are listed in Table VII.2.26.

Table VII.2.26 Selection of LCD Panels

COLOR	DWT1	DWT0	LCD panel
1	1	1	Color Single 8-bit passive LCD format 2
	1	0	Reserved
	0	1	Color Single 8-bit passive LCD format 1
	0	0	Color Single 4-bit passive LCD
0	1	1	Reserved
	1	0	Reserved
	0	1	Mono Single 8-bit passive LCD
	0	0	Mono Single 4-bit passive LCD

At initial reset, DWT is set to "00" (4-bit panel).

SWINV: Software video invert (D9) / LCDC mode register 1 (0x380202)

Inverts the display.

Write "1": Inverted
Write "0": Normal display
Read: Valid

When SWINV is set to "1", the display on the LCD panel is inverted (displayed in inverse video). When SWINV is set to "0", normal display is maintained. Invers operation is applied to output of the look-up tables, and does not affect the display memory.
At initial reset, SWINV is set to "0" (normal display).

BLANK: Display blank enable (D8) / LCDC mode register 1 (0x380202)

Clears the display (entire screen turned blank).

Write "1": Blank
Write "0": Normal display
Read: Valid

When BLANK is set to "0", data in the display memory is displayed on the LCD panel. When BLANK is set to "1", all FPDAT signals are dropped low to clear the display. This setting does not affect the display memory.
At initial reset, BLANK is set to "0" (normal display).

VRAMSEL: VRAM select (DF) / Screen start address register (0x380210)

Selects the VRAM to be used.

Write "1": External SDRAM
Write "0": Internal VRAM
Read: Valid

When VRAMSEL is set to "0", the internal VRAM is selected as the display memory. When VRAMSEL is set to "1", the external SDRAM is selected as the display memory.
At initial reset, VRAMSEL is set to "0" (internal VRAM).

SADDR[14:0]: Screen start address (D[E:0]) / Screen start address register (0x380210)

Sets the screen start address. Depending on the VRAM used, the following addresses must be specified.
When the internal VRAM is used: SADDR = (Screen start address - IVRAM start address) / 2
When the external SDRAM is used: SADDR = (Screen start address - SDRAM start address) / 32
At initial reset, SADDR is set to "0x0" (beginning of the display memory).

LUTR0[5:0]–LUTRFF[5:0]: Look-up table R data (D[7:2]) / Look-up table R/G data register (0x380400–0x3807FC)
LUTG0[5:0]–LUTGFF[5:0]: Look-up table G data (D[F:A]) / Look-up table R/G data register (0x380400–0x3807FC)
LUTB0[5:0]–LUTBFF[5:0]: Look-up table B data (D[7:2]) / Look-up table B data register (0x380402–0x3807FE)

Use these registers to read or write to the look-up tables.
When writing data to the look-up table, you can start writing from any LUT index, note, however, that green and red data (Look-up table R/G data register) must be written prior to blue data (Look-up table B data register), since green and red data are buffered and RGB data for an index can be set to the look-up table when the corresponding blue data is written.
Therefore, blue data must be written even in grayscale mode that uses only green data. In this case, dummy data (ordinary 0) should be written to the red and blue entries.
At initial reset, these registers are undefined.

PLCDCI2–PLCDCI0: LCDC interrupt level (D[6:4]) / LCDC interrupt priority register (0x402A2)

Sets the priority level of the LCDC interrupt.
The interrupt priority level can be set for each channel in the range of 0 to 7.
At initial reset, PLCDCI becomes indeterminate.

ELCDCI: LCDC interrupt enable (D5) / LCDC, USB, SPI interrupt enable register (0x402A6)

Enable or disable interrupt generation to the CPU.

Write "1": Interrupt enabled
Write "0": Interrupt disabled
Read: Valid

The ELCDCI bit is the interrupt enable bit corresponding to the LCDC interrupt factor. When this bit is set to "1", the LCDC interrupt is enabled, and when set to "0", the interrupt is disabled.
At initial reset, ELCDCI is set to "0" (interrupts disabled).

FLCDCI: LCDC interrupt factor flag (D5) / LCDC, USB, SPI interrupt factor flag register (0x402A9)

Indicate the status of LCDC 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

FLCDCI is the interrupt factor flag corresponding to the LCDC interrupt. The flag is set to "1" when an LCDC 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 set value of the corresponding interrupt priority register is higher than the CPU interrupt level (IL).
When using the LCDC 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, FLCDCI becomes indeterminate, so be sure to reset it in the software.

RLCDCI: LCDC IDMA request (D5) / LCDC, USB, SPI IDMA request register (0x402AC)

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 RLCDCI bit is the IDMA request bit corresponding to the LCDC interrupt factor. 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, RLCDCI is set to "0" (interrupt request).

DELCDCI: LCDC IDMA enable (D5) / LCDC, USB, SPI IDMA enable register (0x402AE)

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 DELCDCI bit is the IDMA enable bit corresponding to the LCDC interrupt factors. 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, DELCDCI is set to "0" (IDMA disabled).

Programming Notes

(1) Be sure to configure the LCDC clock generator and supply the LCDC clock to the LCD controller before the LCDC registers and look-up tables can be accessed. Also the bus clock generator must be enabled before starting display as the halt2run_clk clock (generated by bus clock generator) is required for operating the LCDC cache and VRAM.

(2) After the LCDC registers and look-up tables are configured, make sure that the access timeout error flag is not set to "1". If it is set to "1", the LCDC register and look-up table data has not been written normally.

(3) When writing data to an index of the look-up tables, the Look-up table R/G data register must be set up prior to the Look-up table B data register. If the Look-up table B data register is set up first, the index of the look-up tables cannot be updated. Therefore, 32-bit access (little endian) is recommended for writing data to the look-up tables.

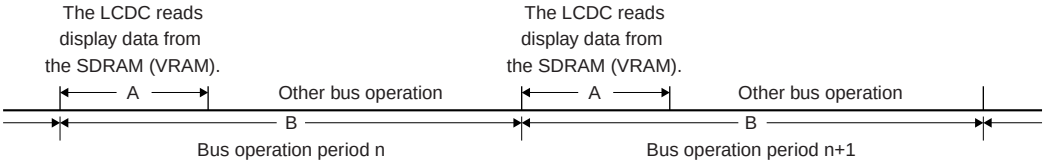
(4) Only 16-bit and 32-bit accesses are allowed for reading/writing from/to the registers other than the look-up table registers. Byte access is forbidden.

(5) The CPU cannot access IVRAM and SDRAM if the CPU clock is faster than the halt2run_clk clock. The LCDC can access these memories and refresh the LCD display correctly, so it is useful to reduce power consumption in HALT/HALT2 mode.

(6) There are some restrictions on setting up the LCDC operating clock frequency (PCLK). Refer to the next section.

System Performance Calculation

External VRAM (SDRAM)



LCDC burst read type	Display data size	Total SDRAM access cycles	LCDC read period A	Bus operation period B	LCDC bus rate
4-burst read	8 bytes	9	187.2 ns	651.2 ns	28.75%
8-burst read	16 bytes	13	270.4 ns	1302.4 ns	20.76%
16-burst read	32 bytes	21	436.8 ns	2604.8 ns	16.77%
32-burst read	64 bytes	37	769.6 ns	5209.6 ns	14.77%

LCD panel: 320 × 240 × RGB
Panel frame rate: 80 Hz
SDRAM parameters: Pre-charge = 1 cycle, Active = 1 cycle, CAS latency = 2 cycles
Bus clock of SDRAM: 48 MHz

Internal VRAM

LCDC burst read type	Display data size	Total SDRAM access cycles	LCDC read period A	Bus operation period B	LCDC bus rate
16-burst read	32 bytes	17	353.6 ns	2604.8 ns	13.57%
32-burst read	64 bytes	33	686.4 ns	5209.6 ns	13.18%

LCD panel: 320 × 240 × RGB
Panel frame rate: 80 Hz
SDRAM parameters: Pre-charge = 1 cycle, Active = 1 cycle, CAS latency = 2 cycles
Bus clock of SDRAM: 48 MHz

Panel frame rate

$$\text{Frame rate} = \frac{f_{PCLK}}{(HDP + HNDP) \times (VDP + VNDP)}$$

f_{PCLK}: LCDC (pixel) clock frequency (Hz)
HDP: Horizontal display period = (HSIZE[6:0] + 1) × 8 (pixels)
HNDP: Horizontal non-display period = (HNDP[4:0] + 4) × 8 (pixels)
VDP: Vertical display period = VSIZE[9:0] + 1 (lines)
VNDP: Vertical non-display period = VNDP[5:0] (lines)

Example:
LCD panel: 320 × 240
Panel frame rate: 80 Hz
LCDC clock frequency (f_{PCLK}): 6.8 MHz

Panel size	Frame rate (Hz)	LCDC clock (MHz)
320 × 240	80	Min. 6.8
	60	Min. 5.9
240 × 160	80	Min. 3.5
	60	Min. 2.6
160 × 160	80	Min. 2.5
	60	Min. 1.8

Sub system clock vs LCDC clock ratio

CPU clock: OSC3 (divided by 1, 2, or 3) → PLL (multiplied by 1 or 2)
→ C33 CLG (divided by 1, 2, 4, or 8)
or OSC1
Sub system clock: OSC3 or PLL output
LCDC clock (PCLK): Sub system clock divided by 1, 2, 3, 4, 5, …15, or 16
VRAM clock (halt2run_clk): Sub system clock divided by 1, 2, or 4

Display mode	VRAM clock : LCDC clock	
	SDRAM	Internal VRAM
1 bpp	3:1 > rate > 1:8	3:1 > rate > 1:16
2 bpp	7:1 > rate > 1:4	6:1 > rate > 1:8
4 bpp	14:1 > rate > 1:2	11:1 > rate > 1:4
8 bpp	rate > 1:1	rate > 1:2
12 bpp	rate > 3:2	rate > 3:4
16 bpp	rate > 2:1	rate > 1:1

Note that the sub system clock frequency may be higher or lower than the CPU clock.

Display mode	Clock rate (VRAM : LCDC)	FIFO empty (VRAM cycles)	One access of BCU (VRAM cycles)	One 16-burst access of LCDC (VRAM cycles)	LCDC bus utilization
16 bpp	2:1	32	< 7	20 or 25 (refresh)	63% or 78%
	3:1	48	< 23	20 or 25 (refresh)	42% or 52%
	4:1	64	< 39	20 or 25 (refresh)	31% or 39%
	7:1	112	< 87	20 or 25 (refresh)	18% or 22%
12 bpp	2:1	42	< 17	20 or 25 (refresh)	48% or 60%
	3:1	64	< 39	20 or 25 (refresh)	31% or 39%
	6:1	128	< 103	20 or 25 (refresh)	16% or 20%
	7:1	147	< 122	20 or 25 (refresh)	14% or 17%
8 bpp	1:1	32	< 7	20 or 25 (refresh)	63% or 78%
	2:1	64	< 39	20 or 25 (refresh)	31% or 39%
	4:1	128	< 103	20 or 25 (refresh)	16% or 20%
	7:1	224	< 199	20 or 25 (refresh)	9% or 11%
4 bpp	1:1	64	< 39	20 or 25 (refresh)	31% or 39%
	2:1	128	< 103	20 or 25 (refresh)	16% or 20%
	4:1	256	< 231	20 or 25 (refresh)	8% or 10%
	7:1	448	< 423	20 or 25 (refresh)	4% or 6%
2 bpp	1:1	128	< 103	20 or 25 (refresh)	16% or 20%
	3:1	384	< 359	20 or 25 (refresh)	5% or 7%
	6:1	768	< 743	20 or 25 (refresh)	3%
1 bpp	1:2	128	< 103	20 or 25 (refresh)	16% or 20%
	1:1	256	< 231	20 or 25 (refresh)	8% or 10%
	2:1	512	< 487	20 or 25 (refresh)	4% or 5%

The LCDC 16-burst access cycles are dependent on the SDRAM refresh status. When a 16-burst access occurs while the SDRAM is in auto-refresh or self-refresh mode, 5 more cycles are required.

THIS PAGE IS BLANK.

VII-3 INTERNAL VRAM AND IVRAM ARBITER

Overview

The S1C33L05 has a built-in 40K-byte synchronous SRAM that can be used as VRAM for storing LCD display data. Addresses 0x3C0000 to 0x3C9FFF in Area 6 are allocated for this SRAM.
This SRAM can be used as not only VRAM but also a general-purpose data RAM.
The SRAM consists of five 8K-byte banks as shown in Figure VII.3.1.

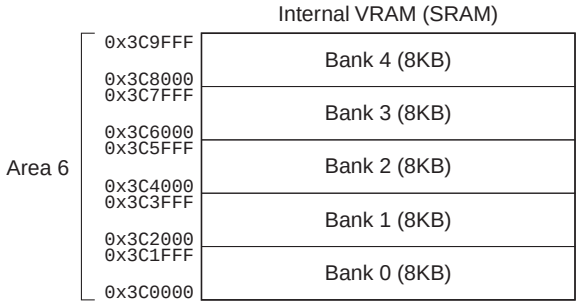


Figure VII.3.1 Configuration of the Internal VRAM

Each bank can be accessed independently. This feature and adoption of the UMA (Unified Memory Access) architecture allows the CPU and LCDC to access different banks simultaneously. When the CPU and LCDC attempt to access the same bank at the same time, the LCDC gets the access authority in order to continue the display refresh sequence normally and the CPU is placed in a wait state until the LCDC cache becomes full. The IVRAM arbiter has charge of arbitrating such bus conflicts.

This SRAM supports 8-bit, 16-bit, and 32-bit accesses.

Internal VRAM Usage Samples

The whole or part of this SRAM area can be used as VRAM. As shown in Figure VII.3.2, the VRAM area start address is specified using the Screen start address register (0x380210) and the area size depends on the panel size and color depth.

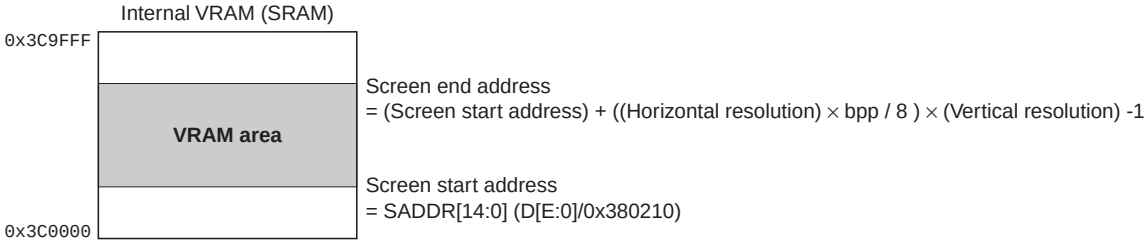


Figure VII.3.2 VRAM Area

Table VII.3.1 shows VRAM usage samples.

Table VII.3.1 VRAM Usage Samples

Panel size	Color depth	Used VRAM size	Screen start address (offset address)	Used VRAM range
320 × 240	4 bpp	37.5K bytes	0x0000	0x3C0000–0x3C95FF
160 × 120	12 bpp	28.125K bytes	0x0000	0x3C0000–0x3C707F
			0x2000	0x3C2000–0x3C907F
160 × 160	1 bpp	3.125K bytes	0x0000	0x3C0000–0x3C0C7F
			0x4444	0x3C4444–0x3C50C3
			0x9000	0x3C9000–0x3C9C7F

Settings the BCU for Using the Internal VRAM

The internal VRAM is mapped in to Area 6. Therefore, in order for the VRAM to be accessed, the BCU register for Area 6 should be set up in accordance with the procedure described below.

1) Configure Area 6 as an internal device area by setting the A6IO (D9) / Access control register (0x48132) to "1".
2) Select little endian as the Area 6 data format by setting the A6EC (D1) / Access control register (0x48132) to "0".
3) Set the number of wait cycles to be inserted when Area 6 is accessed to 1 wait by setting the A6WT[2:0] (D[A:8]) / Areas 6–4 setup register (0x4812A) to "001".
   When the CPU operates in x2 speed mode with a 48 MHz clock, no wait access is allowed, note, however, the other I/O registers mapped in Area 6 need 1 wait insertion.

Tables VII.3.2 and VII.3.3 list the timing parameters and access speeds in the BCU and LCDC.

Table VII.3.2 BCU Timing Parameters and Access Speed

BCU conditions	x1 speed mode (40 MHz)	x2 speed mode (50 MHz)
Software wait cycle	1 min.	0
Output disable delay cycle	0.5	0.5
Read hold cycle	0	0
Read cycle (CPU clock)	2	2
Read speed	50 ns min.	40 ns min.
Write cycle (CPU clock)	2	4
Write speed	50 ns min.	80 ns min.

Longer software-wait cycles, read-hold cycles, or output-disable cycles increase the reliability for accessing, but power consumption is also increased.

Table VII.3.3 LCDC Access Speed

LCDC conditions	x1 speed mode (40 MHz)	x2 speed mode (50 MHz)
16-burst read cycle (halt2run_clk clock)	17	17
Read speed	27 ns	21 ns

IVRAM Arbiter

The IVRAM arbiter circuit arbitrates between the VRAM data read request from the LCDC and the access request from the CPU.

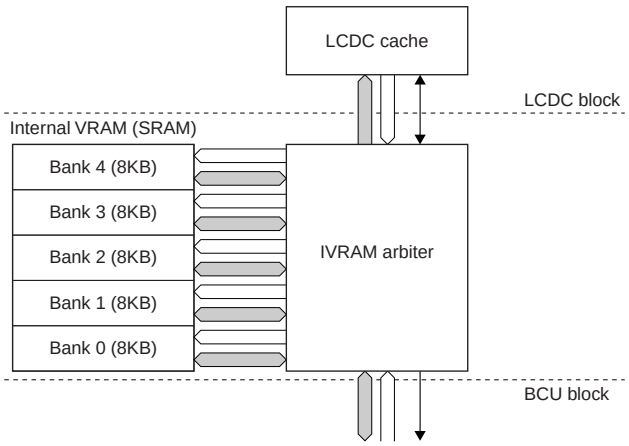
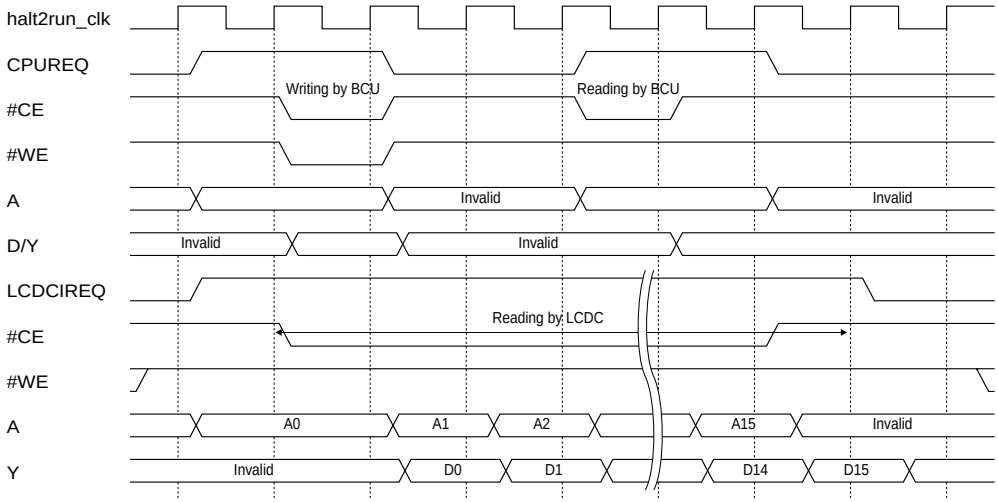
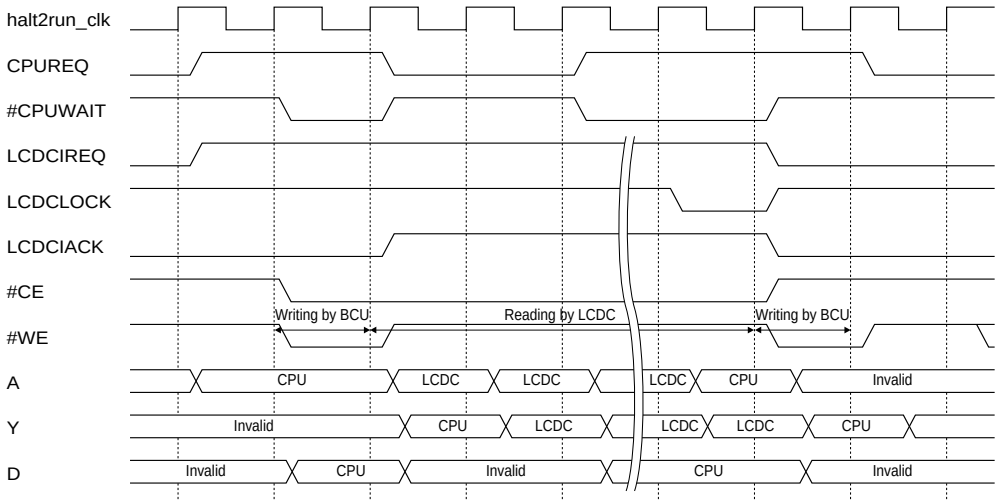


Figure VII.3.3 IVRAM Arbiter

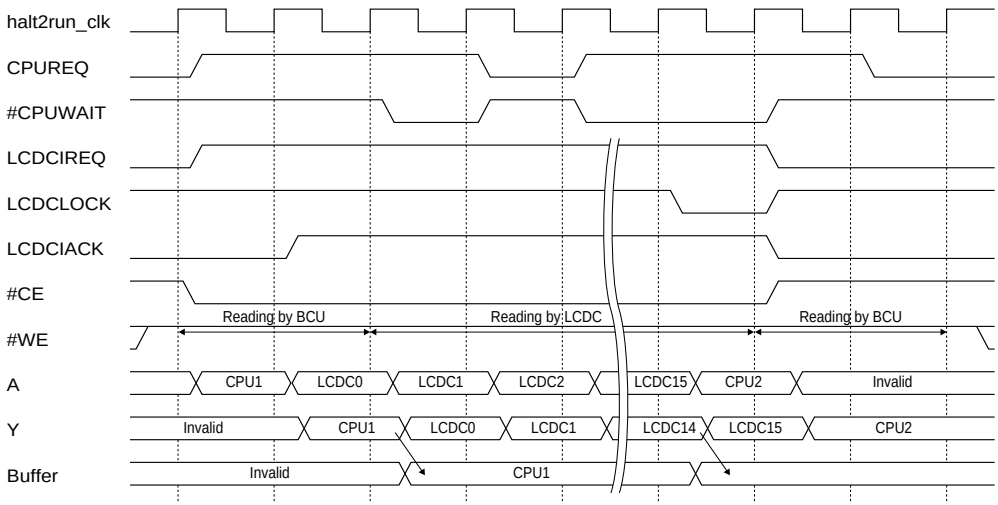
When the CPU and LCDC request to access different banks at the same time, the IVRAM arbiter accepts both requests and the SRAM is accessed by the CPU and LCDC simultaneously.
When the CPU and LCDC request to access the same bank or the CPU requests to access the bank that is being accessed by the LCDC, the IVRAM arbiter gives high priority to the LCDC. The IVRAM arbiter obtains display data requested from the LCDC in burst reading. Therefore, the CPU is placed in a wait state until the LCDC cache becomes full upon completion of burst reading.



1) Accessing different SRAM banks



2) Accessing the same SRAM bank without conflict



3) Accessing the same SRAM bank simultaneously

Figure VII.3.4 Internal VRAM Access Timing Chart

THIS PAGE IS BLANK.

VIII SDRAMC BLOCK

VIII-1 INTRODUCTION

The SDRAMC block consists of an SDRAM interface, instruction/data queue buffers and bus arbiter.

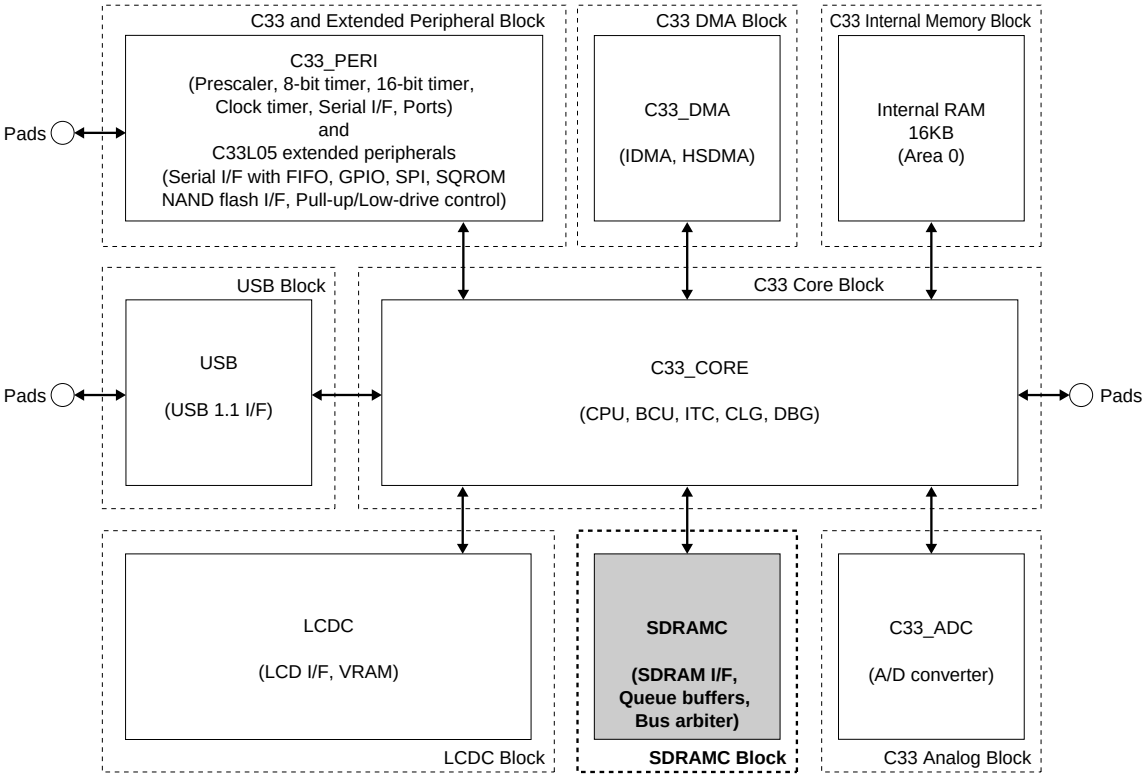


Figure VIII.1.1 SDRAMC Block

THIS PAGE IS BLANK.

VIII-2 SDRAM INTERFACE

The SDRAM controller allows up to 32MB of SDRAM to be connected directly to Areas 13 and 14 or Areas 7 and 8. This section describes how to control the SDRAM interface, and how it operates. For instruction and data queue buffers to improve the SDRAM access performance and the bus arbiter to control SDRAM accesses from the BCU and LCDC, refer to Section VIII-3, "Instruction/Data Queue Buffers", and Section VIII-4, "Bus Arbiter", respectively.

Outline of SDRAM Interface

The following shows the main features and specifications of the SDRAM interface.

- Up to 32MB SDRAM can be connected.
- Two SDRAM areas (Areas 14 and 13 or Areas 8 and 7) are reserved.
  SDRAM configuration examples
  - 16M × 16-bit × 1 chip or 16M × 8-bit × 2 chips (32MB)
  - 8M × 16-bit × 1 chip or 8M × 8-bit × 2 chips (16MB)
  - 4M × 16-bit × 1 chip (8MB)
  - 2M × 8-bit × 2 chips (4MB)
  - 1M × 16-bit × 1 chip (2MB)
- Supports 2 or 4-bank SDRAM (BA1 and BA0 outputs).
  Row address range: 2K (SDA10–SDA0), 4K (SDA11–SDA0), or 8K (SDA12–SDA0)
  Column address range: 256 (SDA7–SDA0), 512 (SDA8–SDA0), or 1K (SDA9–SDA0)
- Supports bank interleaved access.
- Incorporates a programmable 12-bit auto refresh counter.
  The SDRAM can be refreshed as necessary, irrespective of the clock frequency used.
- Intelligent self-refresh mode for low-power operation
- Data bus width: 16 bits
- CAS latency: 2
- Burst length: 2

I/O Pins and Connection

I/O Pins

Table VIII.2.1 lists the pins used for the SDRAM interface.

Table VIII.2.1 I/O Pin List

Pin name	I/O	Function	Function select bit
A[13:12]	O	Address bus (SDA[12:11])	–
A[10:1]	O	Address bus (SDA[9:0])	–
A[15:14]	O	SDRAM bank select signals (SDBA[1:0])	–
D[15:0]	I/O	Data bus (D0–D15)	–
#CE7(#RAS0/#CE13/#RAS2/ P53/#**SDCE**)	I/O	Area 7/13 chip enable / DRAM row strobe / I/O port / SDRAM chip enable	EFP53[1:0](D[7:6])/P50–P53 port function extension register(0x30004A)
#LCAS(PA0/**#SDCAS**)	I/O	DRAM column address strobe (low-byte) / I/O port / SDRAM column address strobe	FPA0[1:0](D[1:0])/PA0–PA2 port function extension register(0x300F60)
#HCAS(PA1/**#SDRAS**)	I/O	DRAM column address strobe (high-byte) / I/O port / SDRAM row address strobe	FPA1[1:0](D[3:2])/PA0–PA2 port function extension register(0x300F60)
BCLK(P60/FOSC1/**SDCLK**)	I/O	Bus clock output / I/O port / OSC1 clock output / SDRAM operating clock output	EFP60[1:0](D[1:0])/P60–P63 port function extension register(0x30004C)
P20(#DRD/**SDCKE**)	I/O	I/O port / DRAM read / SDRAM clock enable	EFP20[1:0](D[1:0])/P20–P21 port function extension register(0x300044)
P21(#DWE/#GAAS/**#SDWE**)	I/O	I/O port / DRAM write / Area address strobe output for GA / SDRAM write	EFP21[1:0](D[3:2])/P20–P21 port function extension register(0x300044)
P61(**SDA10**)	I/O	I/O port / SDRAM address bus 10	EFP61[1:0](D[3:2])/P60–P63 port function extension register(0x30004C)
P62(**LDQM**)	I/O	I/O port / SDRAM data (low-byte) input/output mask signal output	EFP62[1:0](D[5:4])/P60–P63 port function extension register(0x30004C)
P63(**UDQM**)	I/O	I/O port / SDRAM data (high-byte) input/output mask signal output	EFP63[1:0](D[7:6])/P60–P63 port function extension register(0x30004C)

Connection Examples

Figure VIII.2.1 shows an example of how to connect a 16-bit SDRAM to the S1C33. Figure VIII.2.2 shows an example of how to connect two 8-bit SDRAMs to the S1C33.

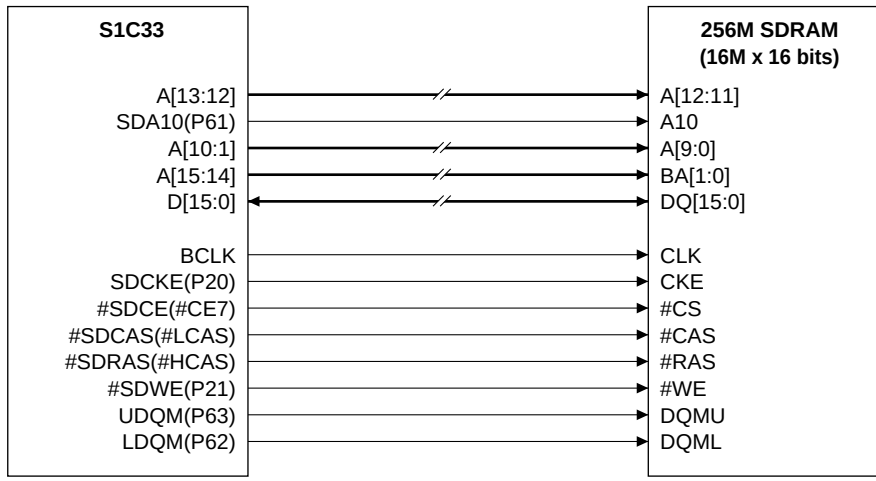


Figure VIII.2.1 Connecting a 16-bit SDRAM (32MB)

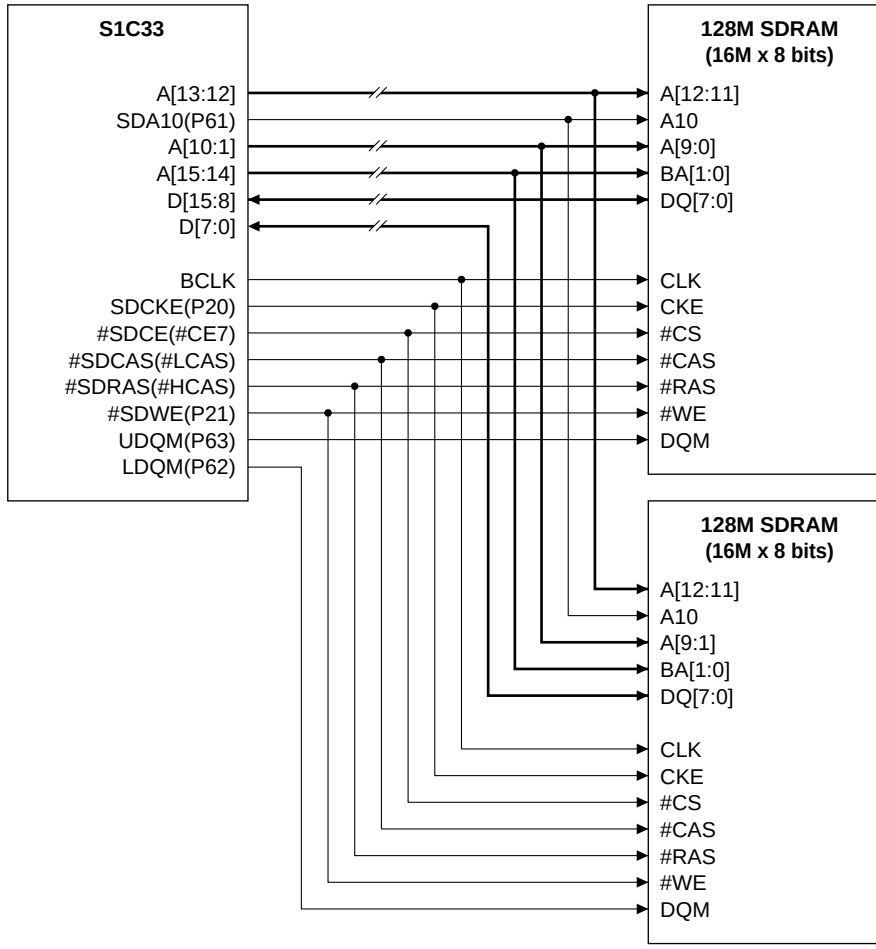


Figure VIII.2.2 Connecting two 8-bit SDRAM (16MB × 2)

Notes:
- Because the SDRAM address bus pins differ in bit numbers from ordinary external address pin names, care must be taken when connecting an SDRAM to the S1C33. (SDRAM address SDA0 is output from the A1 pin, and SDA12 is output from the A13 pin.) Furthermore, the SDA10 signal with a special function is assigned to the P61 pin, and not to the address bus A11.
- To prevent a malfunction, take measures against noise when designing the board patterns for the SDRAM.
- The SDRAM controller supports either Area 7, Area 8, Area 13 or Area 14 as the SDRAM areas. When CEFUNC[1:0] in the BCU is set to "00", only Area 7 and Area 8 are available for SDRAM. When CEFUNC[1:0] in the BCU is set to "01" or "1x", only Area 13 and Area 14 are available for SDRAM.
- The SDRAM controller supports only a 16-bit data bus and a chip enable output for SDRAM. Therefore, only one 16-bit SDRAM device or a pair of 8-bit SDRAM devices can be used. Two 16-bit SDRAM or one 8-bit SDRAM configuration are not available.

Table VIII.2.2 lists several examples of SDRAM chip configurations.

Table VIII.2.2 Chip Configuration Example

SDRAM	Number of devices	Memory size
16M × 16-bit	1	32M bytes
16M × 8-bit	2	32M bytes
8M × 16-bit	1	16M bytes
8M × 8-bit	2	16M bytes
4M × 16-bit	1	8M bytes
2M × 8-bit	2	4M bytes
1M × 16-bit	1	2M bytes

SDRAM Configuration

Pin Configuration

The SDRAM I/F signal pins are shared with other I/O and control signal output ports. At cold start, these pins are all set for the I/O and other ports. When using the SDRAM controller, configure these pins for the SDRAM interface using the port function extension registers listed in Table VIII.2.1. At hot start, the port function extension registers retain their status from prior the reset.

Setting the Bus Clock for SDRAMC

The halt2run_clk clock must be supplied to the SDRAM controller before the SDRAM controller can be used.

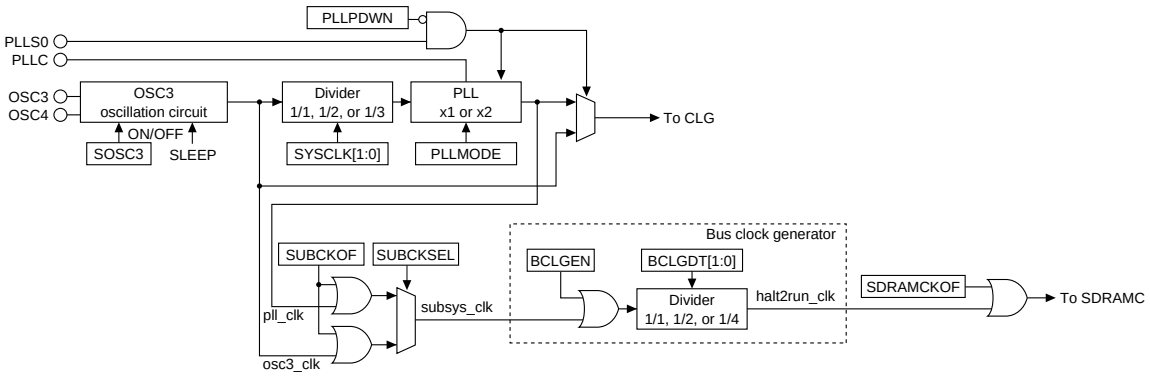


Figure VIII.2.3 SDRAMC Clock

halt2run_clk: This clock is generated by the bus clock generator and is used as the SDRAMC operating clock. To supply this clock to the SDRAMC, set BCLGEN (D7) / Bus clock generator control register (0x300F33) to "1".
The source clock and division ratio for generating the halt2run_clk clock can be selected using SUBCKSEL (D0) / Sub system clock select register (0x300F32) and BCLGDT[1:0] (D[1:0]) / Bus clock generator control register (0x300F33), respectively.

Selecting the source clock (subsys_clk)
SUBCKSEL = "0": OSC3 clock
SUBCKSEL = "1": PLL output clock

Table VIII.2.3 Division Ratio for Generating halt2run_clk Clock

BCLGDT1	BCLGDT0	BCLG source clock
1	1	subsys_clk / 4
1	0	subsys_clk / 2
0	1	subsys_clk
0	0	

Refer to Section III-17, "S1C33L05 Clock System and Miscellaneous Registers", for more information for the clock system.

Note: To operate the SDRAMC, the CPU clock must be slower or equal to the halt2run_clk clock.

BCU Configuration

The SDRAM interface control registers are allocated to addresses 0x3A0200–0x3A0210 in Area 6. Therefore, before the control registers can be accessed, the BCU must be set up following the procedure described below.

1. A6IO (D9) / Access control register (0x48132) = "1"
   This ensures that the internal devices are accessed in Area 6.

2. A6WT[2:0] (D[A:8]) / Areas 6–4 set-up register (0x4812A)
   The number of wait cycles for Area 6 should be set-up properly using the Areas 6–4 set-up register (0x4812A). Otherwise data may not be written to the control register normally. The table below shows examples of wait-cycle settings according to the OSC3 clock frequency and bus speed mode.

Table VIII.2.4 Number of Wait Cycles for Area 6

OSC3 clock	Bus speed mode	Wait cycle count (min.)
48 MHz	x2 speed	0 wait cycles
32 MHz	x1 speed	2 wait cycles

3. SWAITE (D0) / Bus control register (0x4812E) = "1"
   This enables the #WAIT signal.

When the above settings are finished, the SDRAM control registers in Area 6 can be accessed.
Next, configure Areas 7–8 or Areas 13–14 in which SDRAMs are connected.

1. CEFUNC[1:0] (D[A:9]) / DRAM timing set-up register (0x48130)
   CEFUNC[1:0] should be set to "01" or "1x" to use SDRAM in Areas 13–14, or set to "00" to use SDRAM in Areas 7–8.

2. A8IO (DA) or A14IO (DD) / Access control register (0x48132) = "0"
   This sets Areas 7–8 or Areas 13–14 for external access.

3. A8WT[2:0] (D[2:0]) / Areas 8–7 set-up register (0x48128) = "000", "001", or "010"
   A14WT[2:0] (D[2:0]) / Areas 14–13 set-up register (0x48122) = "000", "001", or "010"
   Set the number of wait cycles that will be inserted during SDRAM accessing. The required number of wait cycles varies depending on the operating speed.

Table VIII.2.5 Number of Wait Cycles for Areas 7–8 or 13–14

Max. frequency	Bus speed mode	Wait cycle count (min.)
48 MHz	x2 speed	0 wait cycles
48 MHz	x1 speed	2 wait cycles
40 MHz	x1 speed	1 wait cycle

4. A8SZ (D6) / Areas 8–7 set-up register (0x48128) = "0" or
   A14SZ (D6) / Areas 14–13 set-up register (0x48122) = "0"
   Set the device size of Areas 7–8 or Areas 13–14 to 16 bits.

5. A8DF[1:0] (D[5:4]) / Areas 8–7 set-up register (0x48128) = "00" or
   A14DF[1:0] (D[5:4]) / Areas 14–13 set-up register (0x48122) = "00"
   Set the output disable delay time of Areas 7–8 or Areas 13–14 to 0.5 cycles in order to reduce the SDRAM access time.

6. A8RH(D2) or A14RH(D5) / Read cycle hold time control register (0x4813C) = "0"
   Set Areas 7–8 or Areas 13–14 so that no read-hold cycle will be inserted to reduce the SDRAM access time.

This completes the BCU settings necessary to access the SDRAM.
Make sure the BCU parameters other than those discussed above are set appropriately for the system.

SDRAM Setting Conditions

The SDRAM interface allows the following conditions to be selected.

Table VIII.2.6 SDRAM Interface Parameters

Parameter	Selectable condition	Initial setting	Control bits
Area 7/13 configuration	SDRAM or Another	Another device	AR13C (D1) / SDRAM application configuration register (0x3A0210)
Area 8/14 configuration	SDRAM or Another	Another device	AR14C (D2) / SDRAM application configuration register (0x3A0210)
SDRAM size	1M × 16-bit × 1 chip 4M × 16-bit × 1 chip 8M × 16-bit × 1 chip 16M × 16-bit × 1 chip 2M × 8-bit × 2 chips 8M × 8-bit × 2 chips 16M × 8-bit × 2 chips	1M × 16-bit × 1	ADDRC[2:0] (D[2:0]) / SDRAM control register (0x3A0200)
t_{RC}, t_{RFC}, t_{XSR}	2, 3, 4, or 5 cycles	5 cycles	T80NS[1:0] (D[2:1]) / SDRAM timing setup register (0x3A0208)
t_{RP}, t_{RCD}	1 or 2 cycles	2 cycles	T24NS (D0) / SDRAM timing setup register (0x3A0208)
High performance mode	Low or High performance	High performance	SDHP (D3) / SDRAM application configuration register (0x3A0210)
CAS latency	2 (fixed)	2 (fixed)	–
Burst length	2 (fixed)	2 (fixed)	–

Area configuration

Use the registers described below to select the area(s) to be used for SDRAM.

Area 7/13: AR13C (D1) / SDRAM application configuration register (0x3A0210)
Area 8/14: AR14C (D2) / SDRAM application configuration register (0x3A0210)

Writing "1" to AR13C/AR14C sets the corresponding area for SDRAM use. When AR13C/AR14C = "0" (default), the area is not used for SDRAM. Table VIII.2.7 lists the area configuration for SDRAM.

Table VIII.2.7 Area Configuration for SDRAM

Settings		Area used for SDRAM	
AR13C	AR14C	CEFUNC = "00"	CEFUNC = "01" or "1x"
1	1	Areas 7 and 8	Areas 13 and 14
1	0	Area 7	Area 13
0	1	Area 8	Area 14
0	0	Not used	Not used

Although #SDCE is assigned to the #CE7/#CE13 pin, it is not fixed to Area 7 or Area 13. Even when using Area 8 or Area 14 for SDRAM, the chip enable used for the SDRAM can be #SDCE (#CE7/13).

SDRAM size

Use the ADDRC[2:0] (D[2:0]) / SDRAM control register (0x3A0200) to select SDRAM size and chip configuration. This selection also sets up the bank size, column address size (page size), and row address size.

Table VIII.2.8 Selecting SDRAM Size

ADDRC2	ADDRC1	ADDRC0	Bank	Row	Column	SDRAM configuration	Memory size
1	1	1	–	–	–	reserved	–
1	1	0	4	4K	1K	16M × 8-bit × 2	32M bytes
1	0	1	4	4K	512	8M × 8-bit × 2	16M bytes
1	0	0	2	2K	512	2M × 8-bit × 2	4M bytes
0	1	1	4	8K	512	16M × 16-bit × 1	32M bytes
0	1	0	4	4K	512	8M × 16-bit × 1	16M bytes
0	0	1	4	4K	256	4M × 16-bit × 1	8M bytes
0	0	0	2	2K	256	1M × 16-bit × 1	2M bytes

The SDRAM controller uses only the lower 25 bits of the 28-bit address bus. The relationship between the CPU addresses and the Bank, Column, and Row addresses is shown below.

A(m+n+p)	A(m+n+1)	A(m+n)	· · ·	A(m+1)	A(m)	· · ·	A1	A0
Bank address		Row address			Column address			DQM

Figure VIII.2.4 SDRAM Address

When reading/writing byte data, the SDRAM controller decodes A0/BSL and WRH/BSH into LDQM and UDQM.

m: Column address size (number of bits)
n: Row address size (number of bits)
p: Bank address size (number of bits)

Upper address bits that are not used (depending on memory size) are all set to 0s.

Figures VIII.2.5 to VIII.2.8 list the area configuration and address ranges according to the SDRAM to be used.

Area 13: Used (AR13C = "1", CEFUNC = "01" or "1x") Area 14: Not used (AR14C = "0", CEFUNC = "01" or "1x")		Area 13: Not used (AR13C = "0", CEFUNC = "01" or "1x") Area 14: Used (AR14C = "1", CEFUNC = "01" or "1x")	
1M × 16-bit × 1 (ADDRC[2:0] = "000")	0x2FFFFFF 0x2200000 Mirror 0x21FFFFF 0x2100000 Bank 1 0x20FFFFF 0x2000000 Bank 0 Area 13	1M × 16-bit × 1 (ADDRC[2:0] = "000")	0x3FFFFFF 0x3200000 Mirror 0x31FFFFF 0x3100000 Bank 1 0x30FFFFF 0x3000000 Bank 0 Area 14
4M × 16-bit × 1 (ADDRC[2:0] = "001")	0x2FFFFFF 0x2800000 Mirror 0x27FFFFF 0x2600000 Bank 3 0x25FFFFF 0x2400000 Bank 2 0x23FFFFF 0x2200000 Bank 1 0x21FFFFF 0x2000000 Bank 0 Area 13	4M × 16-bit × 1 (ADDRC[2:0] = "001")	0x3FFFFFF 0x3800000 Mirror 0x37FFFFF 0x3600000 Bank 3 0x35FFFFF 0x3400000 Bank 2 0x33FFFFF 0x3200000 Bank 1 0x31FFFFF 0x3000000 Bank 0 Area 14
8M × 16-bit × 1 (ADDRC[2:0] = "010") or 8M × 8-bit × 2 (ADDRC[2:0] = "101")	0x2FFFFFF 0x2C00000 Bank 3 0x2BFFFFF 0x2800000 Bank 2 0x27FFFFF 0x2400000 Bank 1 0x23FFFFF 0x2000000 Bank 0 Area 13	8M × 16-bit × 1 (ADDRC[2:0] = "010") or 8M × 8-bit × 2 (ADDRC[2:0] = "101")	0x3FFFFFF 0x3C00000 Bank 3 0x3BFFFFF 0x3800000 Bank 2 0x37FFFFF 0x3400000 Bank 1 0x33FFFFF 0x3000000 Bank 0 Area 14
16M × 16-bit × 1 (ADDRC[2:0] = "011") or 16M × 8-bit × 2 (ADDRC[2:0] = "110")	0x2FFFFFF 0x2800000 Bank 1 0x27FFFFF 0x2000000 Bank 0 Area 13	16M × 16-bit × 1 (ADDRC[2:0] = "011") or 16M × 8-bit × 2 (ADDRC[2:0] = "110")	0x3FFFFFF 0x3800000 Bank 1 0x37FFFFF 0x3000000 Bank 0 Area 14
2M × 8-bit × 2 (ADDRC[2:0] = "100")	0x2FFFFFF 0x2400000 Mirror 0x23FFFFF 0x2200000 Bank 1 0x21FFFFF 0x2000000 Bank 0 Area 13	2M × 8-bit × 2 (ADDRC[2:0] = "100")	0x3FFFFFF 0x3400000 Mirror 0x33FFFFF 0x3200000 Bank 1 0x31FFFFF 0x3000000 Bank 0 Area 14

Figure VIII.2.5 SDRAM Map (when Area 13 or 14 is used)

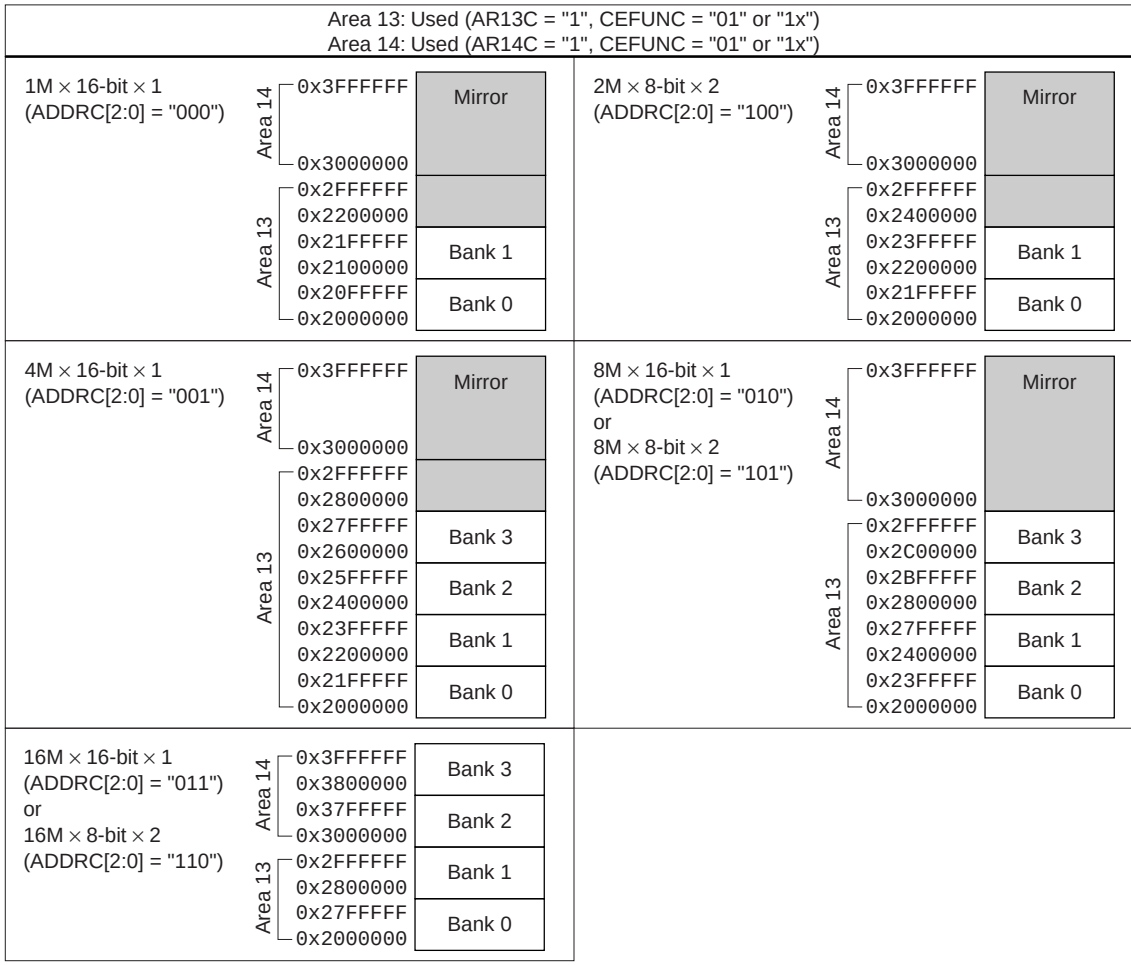


Figure VIII.2.6 SDRAM Map (when Areas 13 and 14 are used)

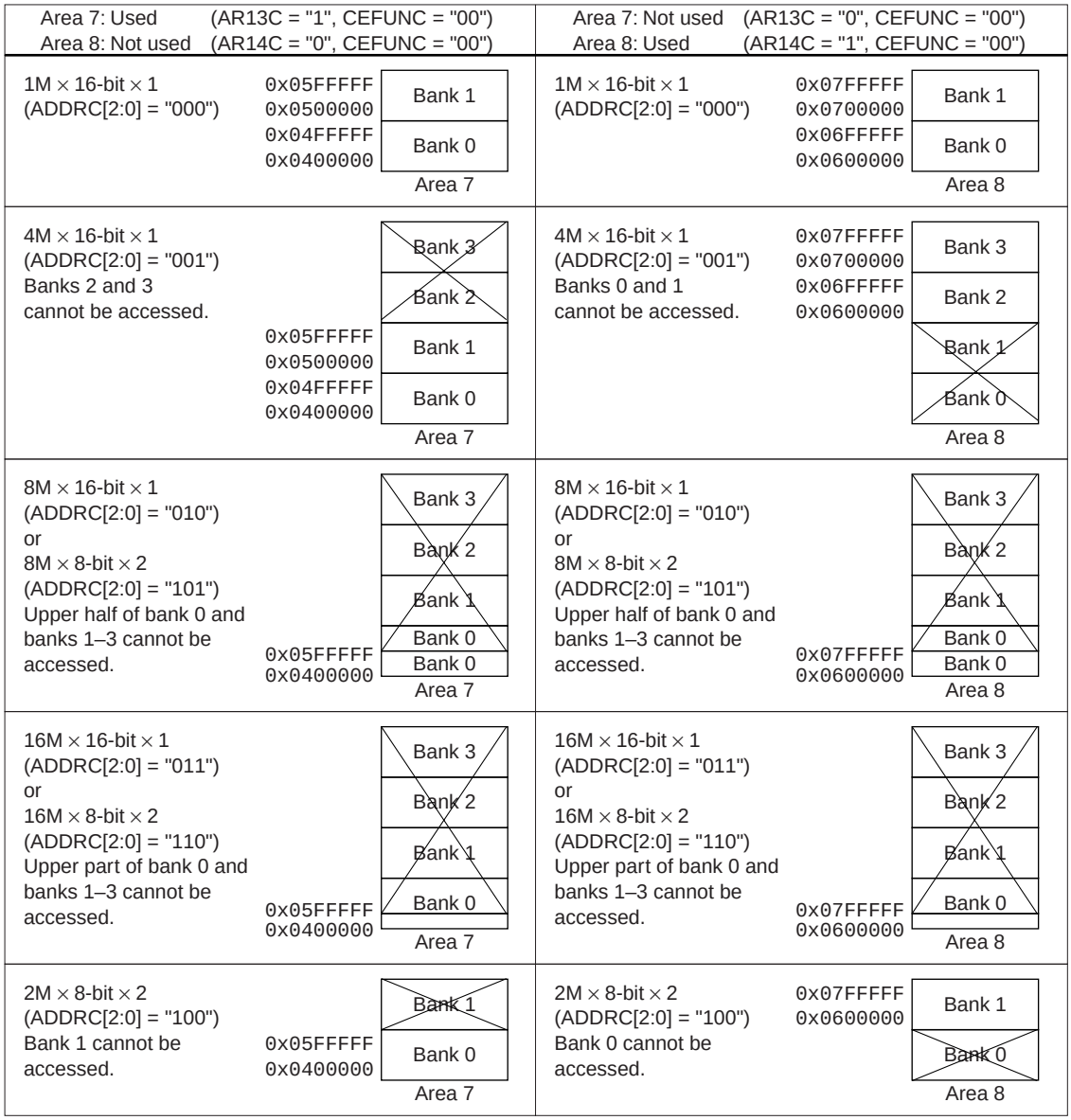


Figure VIII.2.7 SDRAM Map (when Area 7 or 8 is used)

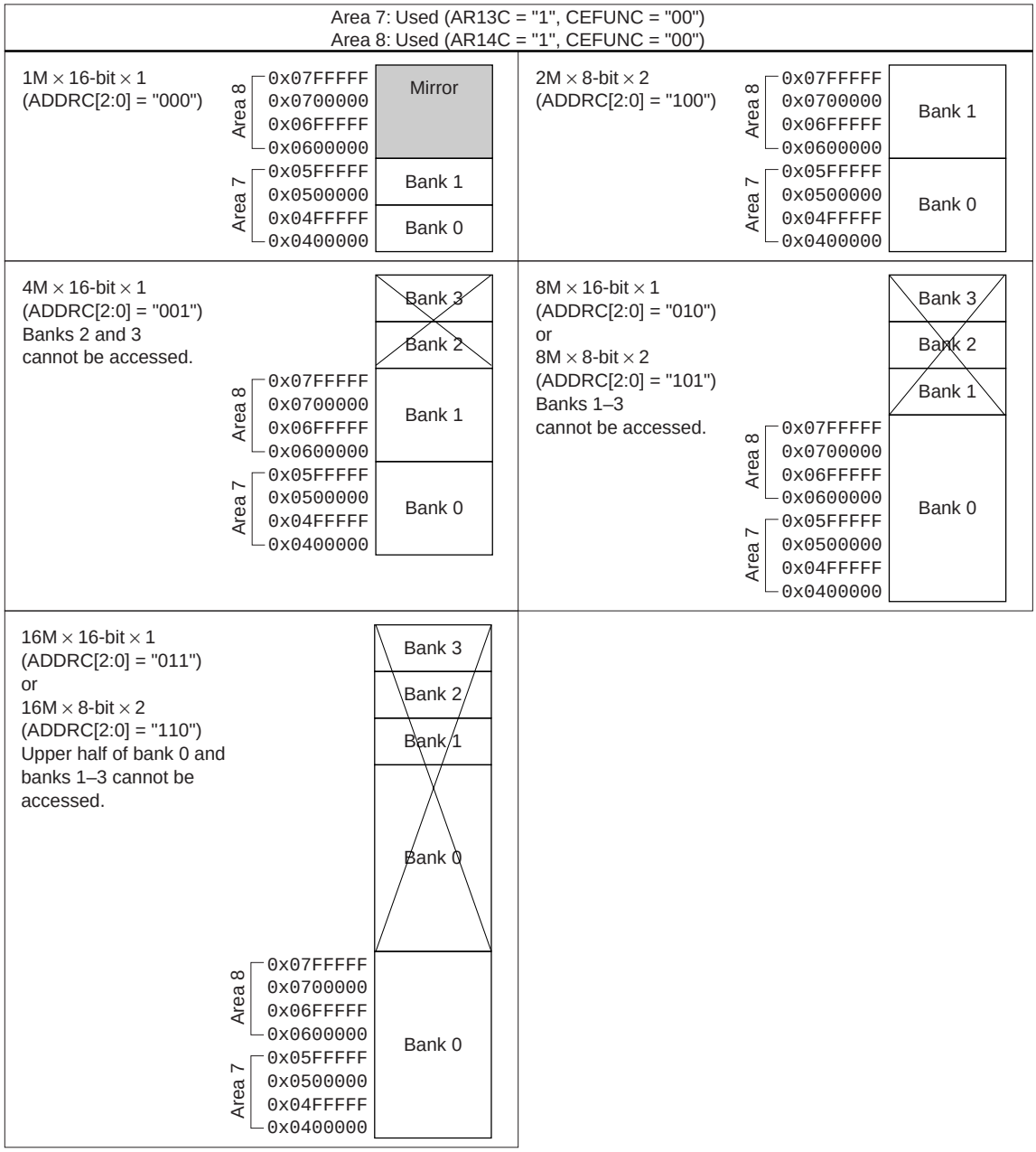


Figure VIII.2.8 SDRAM Map (when Areas 7 and 8 are used)

Timing setup

The following parameters can be set in conformity with SDRAM specifications before use.

Table VIII.2.9 SDRAM Parameters

Symbol	SDRAM parameter	Set values (# of cycles)	Control bits
t_{RC} t_{RFC} t_{XSR}	ACTIVE to ACTIVE command period REFRESH command period Exit SELF REFRESH to ACTIVE command period	2, 3, 4, or 5	T80NS[1:0] (D[2:1]) / SDRAM timing setup register (0x3A0208)
t_{RP} t_{RCD}	PRECHARGE command period ACTIVE to READ or WRITE delay time	1 or 2	T24NS (D0) / SDRAM timing setup register (0x3A0208)

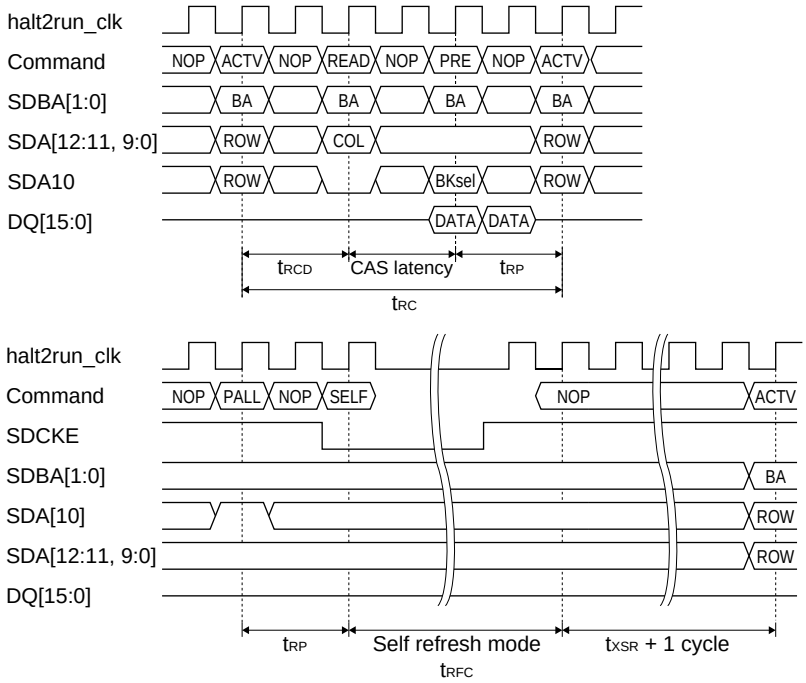


Figure VIII.2.9 SDRAM Timing Parameters

SDRAM Operation

Power-up and Initialization

The following describes the processing sequence for powering up the SDRAM.

1. Setting the BCU and SDRAM access conditions
   Set the BCU and the SDRAM controller as explained in "SDRAM Configuration".
2. SDON (D3) / SDRAM control register (0x3A0200) = "1"
   The SDRAM controller starts operating and the SDRAM interface signals are generated.
3. Set the proper value to the ADDRC[2:0] (D[2:0]) / SDRAM control register (0x3A0200).
4. Set the proper value to the AURCO[11:0] (D[B:0]) / SDRAM auto-refresh counter register (0x3A0204).
5. Set the proper values to the SCKON (D8), SELEN (D7) and SELCO[6:0] (D[6:0]) / SDRAM self-refresh counter register (0x3A0206).
6. Set the proper values to the T80NS[1:0] (D[2:1]) and T24NS (D0) / SDRAM timing register (0x3A0208).
7. Set the proper value to the SDHP (D3), AR14C (D2), AR13C (D1) and IQBON (D0) / SDRAM application configuration register (0x3A0210).
8. Configure the SDRAM I/F pins using the Port function extension register (0x300044, 0x30004A, 0x30004C, 0x300F60).
9. Wait for 100 µs or more after turning on the power to the SDRAM
   After the power to the SDRAM is turned on, the SDRAM must be held in an NOP state (#SDCE = high) for at least 100 µs. Because the duration of this period varies with each SDRAM, consult the specifications for your SDRAM.
10. Controlling INIPRE (D1), INIREF (D0), INIMRS (D2) / SDRAM initial register (0x3A0202)
    In order to initialize the SDRAM, the PALL (Precharge All), REF (Auto Refresh), and MRS (Mode Register Set) commands must be executed sequentially. Note that the initialization sequence depends on the SDRAM.
    Type A: 1. PALL → 2. REF → 3. MRS
    Type B: 1. PALL → 2. MRS → 3. REF

    Refer to the specifications of the SDRAM to be used for the initialization sequence.
    Each command can be executed using the control bit shown below.

    To execute the PALL (Precharge All) command:
    Write "1" to INIPRE (D1/0x3A0202).

    To execute the REF (Auto Refresh) command:
    Write "1" to INIREF (D0/0x3A0202). The REF command should be executed twice and insert the nop instruction between the executions. More REF command executions may be required depending on the SDRAM. Refer to the specifications of the SDRAM for details.
    1) Write "1" to INIREF
    2) Execute the nop instruction
    3) Write "1" to INIREF again

    To execute the MRS (Mode Register Set) command:
    Write "1" to INIMRS (D2/0x3A0202). In this SDRAM interface, the value that is set to the SDRAM mode register is fixed as "00100_010_0_001" by the hardware.

Note: The self-refresh function must be disabled until the SDRAM has finished initialization.

11. Checking SDEN (D3) / SDRAM initial register (0x3A0202)
    SDEN is reset to "0" after power-on, and is set to "1" upon completion of the initialization sequence shown above. Make sure that SDEN is set to "1" before the SDRAM is accessed.
    In addition to being reset at power-on, SDEN is reset to "0" by writing "0" to SDON (D3/0x3A0200).

This completes the SDRAM initialization sequence, allowing access to the SDRAM.

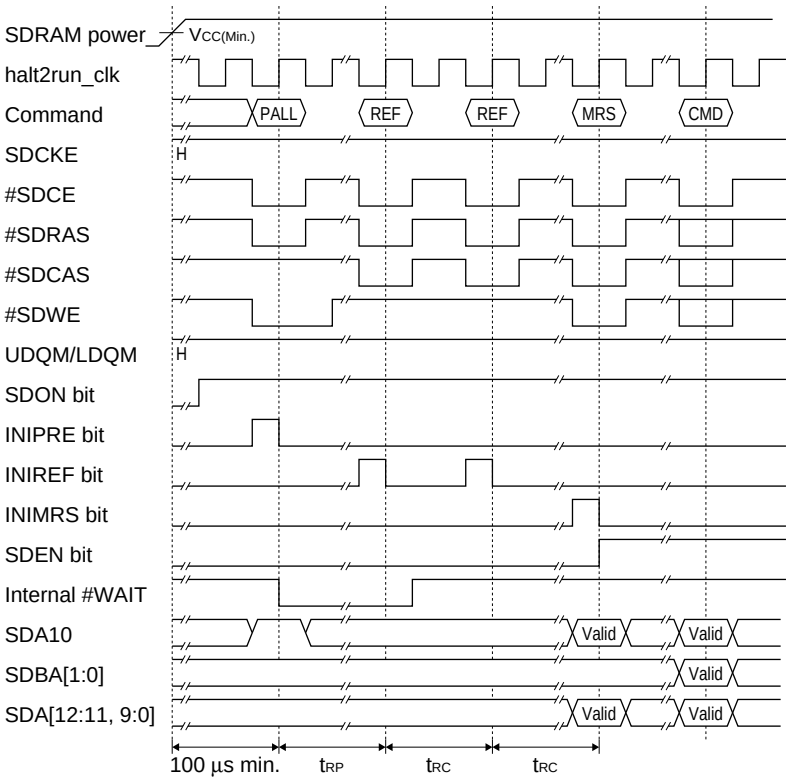


Figure VIII.2.10 SDRAM Power-up and Initialization

SDRAM Commands

The SDRAM is controlled by commands that are comprised of a combination of high or low logic level signals. Table VIII.2.10 lists the commands output by the SDRAM controller.

Table VIII.2.10 List of the Supported SDRAM Commands

Command		Pins								
Function	Symbol	SDCKE	DQM U/LDQM	Bank A[15:14]	SDA10	SDA A[13:1]	#SDCE	#SDRAS	#SDCAS	#SDWE
Deselect	–	H	X	X	X	X	H	X	X	X
Bank Active	ACTV	H	X	V	V	V	L	L	H	H
Bank Precharge	PRE	H	X	V	L	X	L	L	H	L
Precharge All	PALL	H	X	X	H	X	L	L	H	L
Write	WRIT	H	X	V	L	V	L	H	L	L
Read	READ	H	X	V	L	V	L	H	L	H
Mode Register Set	MRS	H	X	X	V	V	L	L	L	L
NOP	NOP	H	X	X	X	X	L	H	H	H
Auto Refresh	REF	H	X	X	X	X	L	L	L	H
Self Refresh Entry	SELF	H → L	X	X	X	X	L	L	L	H
Self Refresh Exit	–	L → H	X	X	X	X	H	X	X	X
Data Write/Output Enable	–	H	L	X	X	X	X	X	X	X
Data Write/Output Disable	–	H	H	X	X	X	X	X	X	X

V = valid, X = don't care, L = low level, H = high level

Because all of these commands are output by the SDRAM controller as necessary, they do not need to be controlled by the user program, except for initializing the SDRAM.

Read Cycle

The SDRAM controller always reads data from the SDRAM in bursts. The burst length is fixed at 2.
Figure VIII.2.11 shows an example of timing chart when reading out 4-word data from the same row address.
Note that the following examples assume that instruction/data queue buffers are disabled. Refer to Section VIII-3, "Instruction/Data Queue Buffers", for operations when the queue buffers are enabled.

Example of parameter settings: t_{RCD} = 2 cycles, t_{RP} = 2 cycles

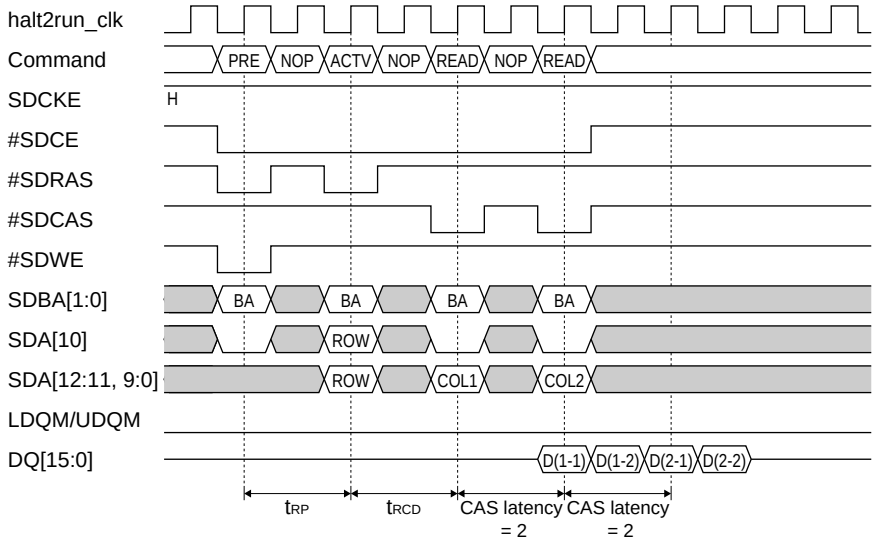


Figure VIII.2.11 Burst Read in the Same Page

Figure VIII.2.12 shows an example of a timing chart in cases where the row address is changed during burst read.

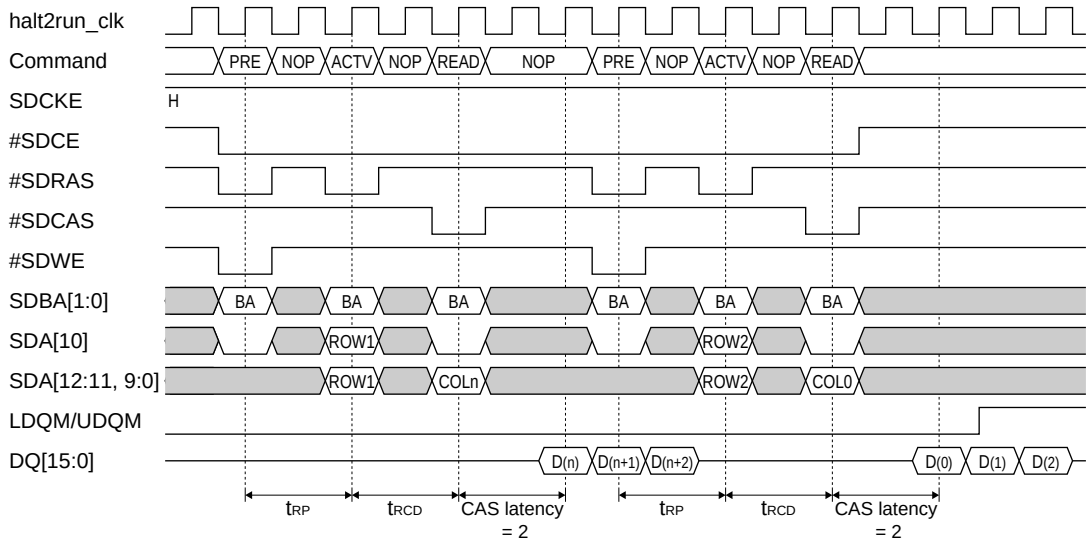


Figure VIII.2.12 Changing Row Address During Burst Read

Write Cycle

When writing to the SDRAM, data are always written in a single operation.

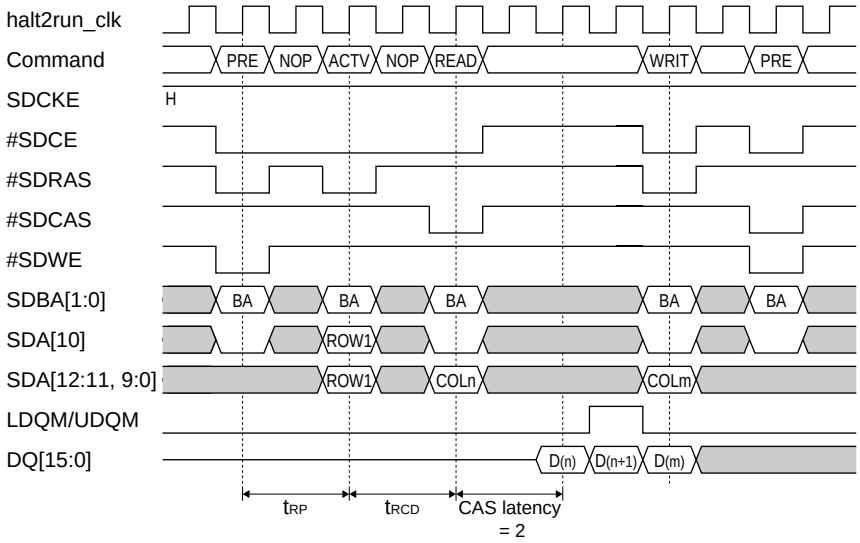


Figure VIII.2.13 Burst Read to Single Write (same page)

Bank Interleaved Access

Multiple banks (up to four banks) can be activated at the same time. This makes it possible to access the SDRAM successively, one bank after another without issuing the ACTV (Active) command.

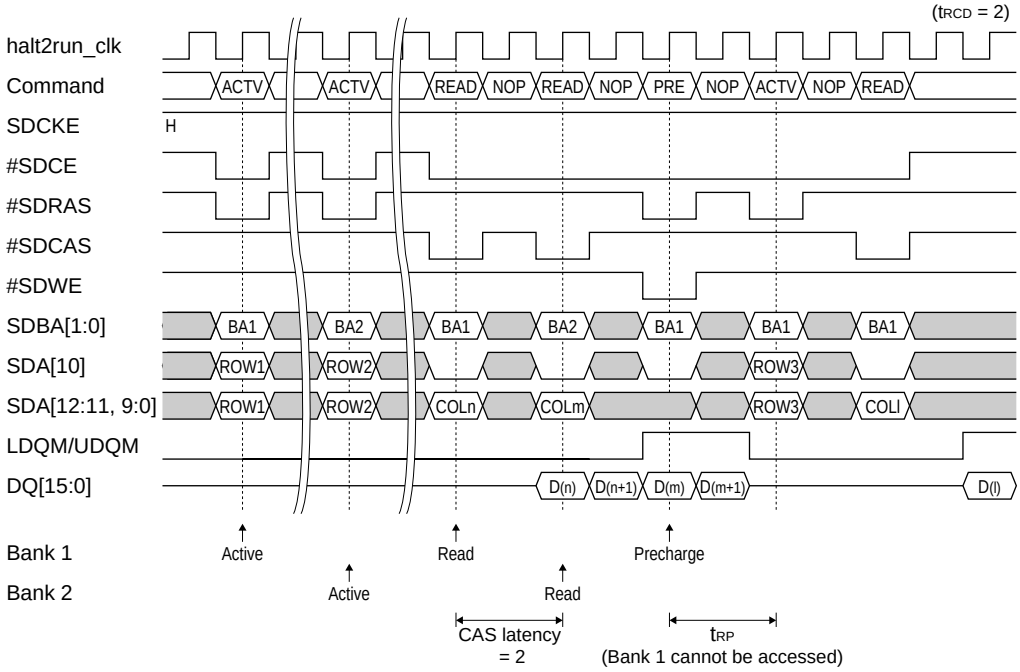


Figure VIII.2.14 Bank Interleaved Access

Refresh Mode

The SDRAM controller supports two SDRAM refresh modes: auto refresh and self-refresh.

Auto refresh

The SDRAM controller incorporates a 12-bit auto refresh counter. This counter continues counting on the halt2run_clk clock edges, and when a specified count is reached, commands are sent to the SDRAM that precharges and auto-refreshes all banks. The counter is reset at that time, and starts counting for the next refresh period. The counter is also reset by self-refresh.

The auto-refresh period is determined by the halt2run_clk clock frequency and the count value set in the Auto refresh counter register AURCO[11:0] (D[B:0]/0x3A0204). For AURCO, set the appropriate value meeting the specifications of your SDRAM. The count value is obtained by the equation below.

$$AURCO \leq \frac{RFP}{ROWS} \times f_{CLK} - BL - CL - 2 \times t_{RP} - t_{RCD} - 3$$

RFP: Maximum refresh period [s]
ROWS: Row address size
f_{CLK}: halt2run_clk clock frequency [Hz]
BL: Burst length (= 2)
CL: CAS latency (= 2)
t_{RP}: PRECHARGE command period [Number of cycles]
t_{RCD}: ACTIVE to READ or WRITE delay time [Number of cycles]

If RFP = 64 ms, ROWS = 4,096, f_{CLK} = 20 MHz, t_{RP} = 2, and t_{RCD} = 2, for example, the value to set is calculated as follows:

$$AURCO \leq \frac{0.064}{4{,}096} \times 20{,}000{,}000 - 2 - 2 - 2 \times 2 - 2 - 3 = 299$$

Therefore, set any value equal to or less than 299 (0x12b) for AURCO.

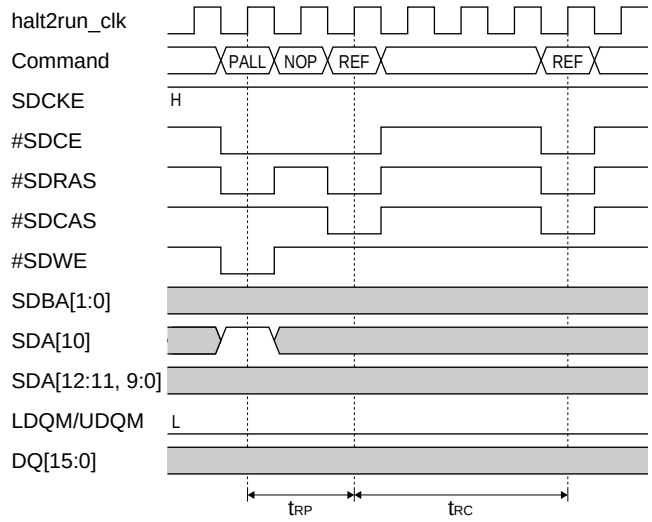


Figure VIII.2.15 Auto Refresh

Self refresh

Self-refresh uses the SDRAM's self-refresh function and does not require clock pulses during the refresh period, thus helping to reduce the chip's power consumption. This self-refresh function is also used for data retention during power-down mode.

To cause the SDRAM to be self-refreshed, set the SELEN (D7) / SDRAM self-refresh counter register (0x3A0206) to "1". This enables the SDRAM controller to send the self-refresh command (which sets the SDCKE output to low) to the SDRAM. The command is actually sent a certain time after accessing or auto-refreshing the SDRAM, so the SDRAM controller contains a self-refresh counter to count this time. The counter counts on halt2run_clk clock edges, and when the designated count is reached, the SDRAM controller sends the refresh command to the SDRAM. When an SDRAM access or auto-refresh command is issued, the counter is reset and starts counting again. The designated value for the counter can be specified in a range of 0 to 127 by using the SELCO[6:0] (D[6:0] / SDRAM self-refresh counter register (0x3A0206).

When an SDRAM access occurs during self-refresh mode, SDCKE is returned high and the SDRAM is taken out of self-refresh mode. Also the SDRAM is taken out of self-refresh mode when the auto-refresh command is issued. Therefore, the self-refresh counter value to be set must be smaller than the auto-refresh counter value.

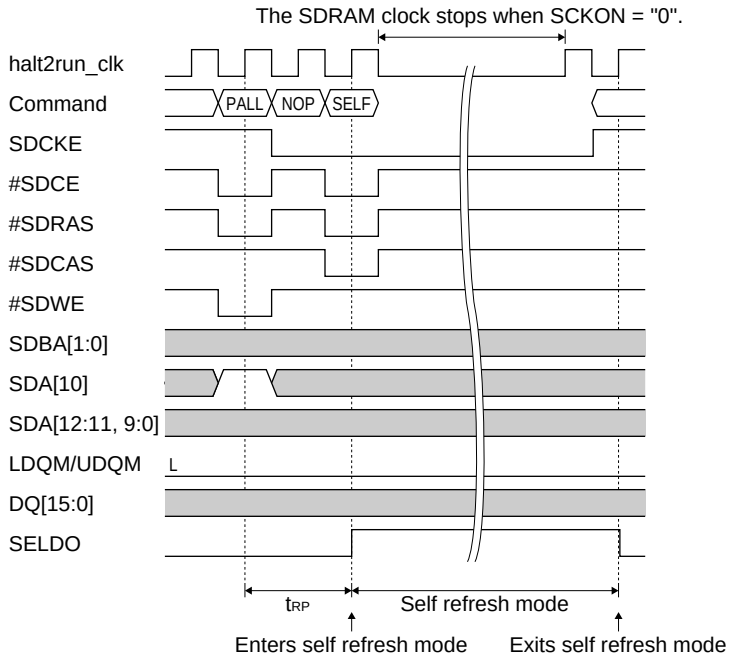


Figure VIII.2.16 Self Refresh

During self-refresh (while SDCKE = low), the SELDO (D9) / SDRAM self-refresh counter register (0x3A0206) remains "1". Therefore, it is possible to determine whether or not self-refresh is in operation by reading this status register.

Furthermore, SDRAM clock output during self-refresh can be turned off in order to reduce the chip's power consumption by setting the SCKON (D8) / SDRAM self-refresh counter register (0x3A0206) to "0".

Power-down Mode

The SDRAM controller supports three power-down modes for the S1C33 Core (HALT, HALT2, and SLEEP).
In HALT mode, the bus clock is not turned off. Therefore, this mode can be set at any time.
In HALT2 mode, the SDRAM cannot be accessed from the BCU, however, the LCDC can read the SDRAM for refreshing the LCD display when the SDRAM is used as VRAM.
In SLEEP mode, the SDRAM clock stops. Therefore, the SDRAM must be placed in self-refresh mode before entering SLEEP mode, by following the procedure described below.

1. Set SELEN (D7/0x3A0206) to "1" in order to enable the SDRAM's self-refresh function.
2. Check to see that SELDO (D9/0x3A0206) = "1" (i.e., SDRAM is being self-refreshed).
3. Execute the slp instruction.

Notes: • Because the SDRAM is taken out of self-refresh mode when accessed, steps 2 and 3 of the above procedure must be executed on other memory than SDRAMs.

• A few kinds of SDRAM have strict timing requirements for the RAS and CAS signals during self-refresh mode when the SELEN (D7) / SDRAM self-refresh counter register (0x3A0206) is set to "1", especially, depending on the SDCLK, for the signal hold time. To avoid the hold time violation caused by the PCB layout or other factors, we recommend the following circuit adjustment for users' reference:

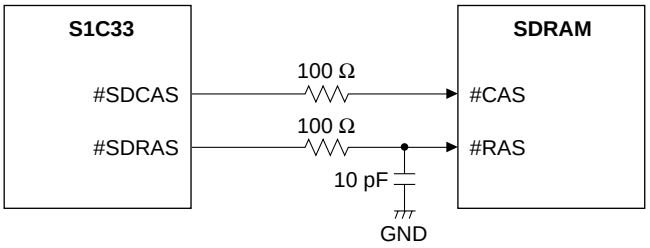


Figure VIII.2.17 Connecting the #SDRAS and #SDCAS signals

I/O Memory of SDRAM Interface

Table VIII.2.11 shows the control bits of the SDRAM interface. These registers are mapped into Area 6.

Table VIII.2.11 Control Bits for SDRAM Interface

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
P20–P21 port function extension register	**0300044 (B)**	D7–4	**–**	reserved	–			–	–	0 when being read.
		D3	**EFP211**	P21 port extended function	EFP21[1:0]		Function	0	R/W	
		D2	**EFP210**		1	*	reserved	0		
					0	1	#SDWE			
					0	0	P21/#DWE			
		D1	**EFP201**	P20 port extended function	EFP20[1:0]		Function	0	R/W	
		D0	**EFP200**		1	*	reserved	0		
					0	1	SDCKE			
					0	0	P20/#DRD			
P50–P53 port function extension register	**030004A (B)**	D7	**EFP531**	P53 port extended function	EFP53[1:0]		Function	0	R/W	
		D6	**EFP530**		1	1	reserved	0		
					1	0	#SDCE			
					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.			
P60–P63 port function extension register	**030004C (B)**	D7	**EFP631**	P63 port extended function	EFP63[1:0]		Function	0	R/W	
		D6	**EFP630**		1	*	reserved	0		
					0	1	UDQM			
					0	0	P63			
		D5	**EFP621**	P62 port extended function	EFP62[1:0]		Function	0	R/W	
		D4	**EFP620**		1	*	reserved	0		
					0	1	LDQM			
					0	0	P62			
		D3	**EFP611**	P61 port extended function	EFP61[1:0]		Function	0	R/W	
		D2	**EFP610**		1	*	reserved	0		
					0	1	SDA10			
					0	0	P61			
		D1	**EFP601**	P60 port extended function	EFP60[1:0]		Function	0	R/W	
		D0	**EFP600**		1	1	SDCLK	0		
					1	0	FOSC1			
					0	1	P60			
					0	0	BCLK			
Sub system clock select register	**0300F32 (B)**	D7–1	**–**	reserved	–			–	–	0 when being read.
		D0	**SUBCKSEL**	Sub system source clock select	1	PLL clock	0 OSC3 clock	0	R/W	
Bus clock generator control register	**0300F33 (B)**	D7	**BCLGEN**	Bus clock generator enable	1	Enabled	0 Disabled	0	R/W	
		D6–2	**–**	reserved	–			–	–	0 when being read.
		D1	**BCLGDT1**	Bus clock setup	BCLGDT[1:0]		Division ratio	0	R/W	
		D0	**BCLGDT0**	(Sub system clock division ratio)	1	1	subsys_clk / 4	0		
					1	0	subsys_clk / 2			
					0	*	subsys_clk / 1			

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PA0–PA2 port function extension register	**0300F60 (B)**	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**FPA21**	PA2 port extended function	FPA2[1:0]: 1 * reserved; 0 1 PA2; 0 0 P30, etc.	0	R/W	
		D4	**FPA20**			0		
		D3	**FPA11**	PA1 port extended function	FPA1[1:0]: 1 1 reserved; 1 0 #SDRAS; 0 1 PA1; 0 0 #HCAS	0	R/W	
		D2	**FPA10**			0		
		D1	**FPA01**	PA0 port extended function	FPA0[1:0]: 1 1 reserved; 1 0 #SDCAS; 0 1 PA0; 0 0 #LCAS	0	R/W	
		D0	**FPA00**			0		
SDRAM control register	**03A0200 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**SDON**	SDRAM controller enable	1 Enabled; 0 Disabled	0	R/W	
		D2	**ADDRC2**	SDRAM address configuration	ADDRC[2:0]: 1 1 1 reserved; 1 1 0 16M x 8 bits x 2; 1 0 1 8M x 8 bits x 2; 1 0 0 2M x 8 bits x 2; 0 1 1 16M x 16 bits x 1; 0 1 0 8M x 16 bits x 1; 0 0 1 4M x 16 bits x 1; 0 0 0 1M x 16 bits x 1	0	R/W	
		D1	**ADDRC1**			0		
		D0	**ADDRC0**			0		
SDRAM initial register	**03A0202 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**SDEN**	SDRAM initialize flag	1 Initialized; 0 Not initialized	0	R	
		D2	**INIMRS**	MRS trigger	1 Trigger; 0 Invalid	0	W	0 when being read.
		D1	**INIPRE**	PRE trigger	1 Trigger; 0 Invalid	0	W	
		D0	**INIREF**	REF trigger	1 Trigger; 0 Invalid	0	W	
SDRAM auto-refresh counter register	**03A0204 (HW)**	DF–C	**–**	reserved	–	–	–	0 when being read.
		DB	**AURCO11**	SDRAM auto-refresh counter	0x0 to 0xFFF	0	R/W	
		DA	**AURCO10**			0		
		D9	**AURCO9**			0		
		D8	**AURCO8**			0		
		D7	**AURCO7**			0		
		D6	**AURCO6**			0		
		D5	**AURCO5**			0		
		D4	**AURCO4**			0		
		D3	**AURCO3**			0		
		D2	**AURCO2**			0		
		D1	**AURCO1**			0		
		D0	**AURCO0**			0		
SDRAM self-refresh counter register	**03A0206 (HW)**	DF–A	**–**	reserved	–	–	–	0 when being read.
		D9	**SELDO**	SDRAM self-refresh status	1 Refresh mode; 0 Done	0	R	
		D8	**SCKON**	SDRAM clock during self-refresh	1 Enabled; 0 Disabled	0	R/W	
		D7	**SELEN**	SDRAM self-refresh enable	1 Enabled; 0 Disabled	0	R/W	
		D6	**SELCO6**	SDRAM self-refresh counter	0x0 to 0x7F	1	R/W	
		D5	**SELCO5**			1		
		D4	**SELCO4**			1		
		D3	**SELCO3**			1		
		D2	**SELCO2**			1		
		D1	**SELCO1**			1		
		D0	**SELCO0**			1		
SDRAM timing setup register	**03A0208 (HW)**	DF–3	**–**	reserved	–	–	–	0 when being read.
		D2	**T80NS1**	SDRAM t_{RC}, t_{RFC} and t_{XSR} cycles	T80NS[1:0]: 1 1 5 cycles; 1 0 4 cycles; 0 1 3 cycles; 0 0 2 cycles	1	R/W	
		D1	**T80NS0**			1		
		D0	**T24NS**	SDRAM t_{RP} and t_{RCD} cycles	1 2 cycles; 0 1 cycle	1	R/W	
SDRAM application configuration register	**03A0210 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**SDHP**	SDRAM high-performance access	1 High perform.; 0 Low perform.	0	R/W	
		D2	**AR14C**	Area 8/14 configuration	1 SDRAM; 0 Other	0	R/W	
		D1	**AR13C**	Area 7/13 configuration	1 SDRAM; 0 Other	0	R/W	
		D0	**IQBON**	Instruction queue buffer enable	1 Enabled; 0 Disabled	0	R/W	

EFP201–EFP200: P20 port extended function (D[1:0]) / P20–P21 port function extension register (0x300044)
EFP211–EFP210: P21 port extended function (D[3:2]) / P20–P21 port function extension register (0x300044)
EFP531–EFP530: P53 port extended function (D[7:6]) / P50–P53 port function extension register (0x30004A)
EFP601–EFP600: P60 port extended function (D[1:0]) / P60–P63 port function extension register (0x30004C)
EFP611–EFP610: P61 port extended function (D[3:2]) / P60–P63 port function extension register (0x30004C)
EFP621–EFP620: P62 port extended function (D[5:4]) / P60–P63 port function extension register (0x30004C)
EFP631–EFP630: P63 port extended function (D[7:6]) / P60–P63 port function extension register (0x30004C)
FPA01–FPA00: PA0 port extended function (D[1:0]) / PA0–PA2 port function extension register (0x300F60)
FPA11–FPA10: PA1 port extended function (D[3:2]) / PA0–PA2 port function extension register (0x300F60)

Switches the port functions.

Table VIII.2.12 Port Extended Functions

Function extension bit	EFPxx[1:0] = "00" FPxx[1:0] = "00"	EFPxx[1:0] = "01" FPxx[1:0] = "01"	EFPxx[1:0] = "10" FPxx[1:0] = "10"	EFPxx[1:0] = "11" FPxx[1:0] = "11"
EFP20[1:0]	P20/#DRD	**SDCKE**	–	–
EFP21[1:0]	P21/#DWE	**#SDWE**	–	–
EFP53[1:0]	#CE7/#CE13, etc.	P53	**#SDCE**	–
EFP60[1:0]	BCLK	P60	FOSC1	**SDCLK**
EFP61[1:0]	P61	**SDA10**	–	–
EFP62[1:0]	P62	**LDQM**	–	–
EFP63[1:0]	P63	**UDQM**	–	–
FPA0[1:0]	#LCAS	PA0	**#SDCAS**	–
FPA1[1:0]	#HCAS	PA1	**#SDRAS**	–

Select the SDRAM I/F functions that are listed in a boldface type.
At initial reset, EFP and FP are set to "00".

SUBCKSEL: Sub system source clock select (D0) /Sub system clock select register (0x300F32)

Selects the source clock for the bus clock and LCDC clock generators.

Write "1": PLL output clock
Write "0": OSC3 clock
Read: Valid

When "1" is written to SUBCKSEL, the PLL output clock is selected as the source clock (subsys_clk) for the bus clock and LCDC clock generators in the extended block. When "0" is written, the OSC3 clock is selected.
At initial reset, SUBCKSEL is set to "0" (OSC3 clock).

BCLGEN: Bus clock generator enable (D7) / Bus clock generator control register (0x300F33)

Enables/disables the bus clock generator.

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

When "1" is written to BCLGEN, the source clock supply is enabled and the bus clock generator starts outputting the halt2run_clk clock for the SDRAM and internal VRAM. When "0" is written, the bus clock generator stops outputting the halt2run_clk clock.
At initial reset, BCLGEN is set to "0" (disabled).

BCLGDT1–BCLGDT0: Bus clock setup (D[1:0]) / Bus clock generator control register (0x300F33)

Configures the frequency divider in the bus clock generator to set the halt2run_clk clock frequency.

Table VIII.2.13 Setting the Division Ratio for Bus Clock

BCLGDT1	BCLGDT0	BCLG source clock
1	1	subsys_clk / 4
1	0	subsys_clk / 2
0	1	subsys_clk
0	0	

At initial reset, BCLGDT is set to "0" (subsys_clk).

SDON: SDRAM controller enable (D3) / SDRAM control register (0x3A0200)

Enable the SDRAM controller.

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

When SDON is set to "1", the SDRAM controller activates and outputs the SDRAM clock from the SDCLK (BCLK) pin. Before setting SDON to "1", the halt2run_clk clock must be supplied to the SDRAM controller.
At initial reset, SDON is set to "0" (disabled).

ADDRC2–ADDRC0: SDRAM address configuration (D[2:0]) / SDRAM control register (0x3A0200)

Selects an SDRAM size and chip configuration. This selection also sets up the bank size, column address size (page size), and row address size.

Table VIII.2.14 Selecting SDRAM Size

ADDRC2	ADDRC1	ADDRC0	Bank	Row	Column	SDRAM configuration	Memory size
1	1	1	–	–	–	reserved	–
1	1	0	4	4K	1K	16M × 8-bit × 2	32M bytes
1	0	1	4	4K	512	8M × 8-bit × 2	16M bytes
1	0	0	2	2K	512	2M × 8-bit × 2	4M bytes
0	1	1	4	8K	512	16M × 16-bit × 1	32M bytes
0	1	0	4	4K	512	8M × 16-bit × 1	16M bytes
0	0	1	4	4K	256	4M × 16-bit × 1	8M bytes
0	0	0	2	2K	256	1M × 16-bit × 1	2M bytes

At initial reset, ADDRC is set to "0" (1M × 16-bit × 1).

SDEN: SDRAM initialize flag (D3) / SDRAM initial register (0x3A0202)

Indicates that the SDRAM has finished initialization (Mode Register Set).

Read "1": Initialized
Read "0": Not initialized
Write: Invalid

SDEN is reset to "0" after power-on, and is set to "1" upon completion of the initialization sequence. Make sure that SDEN is set to "1" before the SDRAM is accessed.
In addition to being reset at power-on, SDEN is reset to "0" by writing "0" to SDON (D3/0x3A0200).
At initial reset, SDEN is set to "0" (Not initialized).

INIMRS: MRS trigger (D2) / SDRAM initial register (0x3A0202)

Executes the MRS (Mode Register Set) command.

Write "1": Trigger
Write "0": Invalid
Read: Always "0"

Writing "1" to INIMRS sends the MRS (Mode Register Set) command to initialize the SDRAM. In this SDRAM interface, the value that is set to the SDRAM mode register is fixed as "00100_010_0_001" by the hardware.

In order to initialize the SDRAM, the PALL (Precharge All), REF (Auto Refresh), and MRS (Mode Register Set) commands must be executed sequentially. Note that the initialization sequence depends on the SDRAM.
Type A: 1. PALL → 2. REF → 3. MRS
Type B: 1. PALL → 2. MRS → 3. REF

Refer to the specifications of the SDRAM to be used for the initialization sequence.
The initialization sequence must be performed after holding the SDRAM in an NOP state for at least 100 µs (this varies with each SDRAM) after powering up the SDRAM.
Writing "0" results in No Operation.
Since INIMRS is a write-only bit, its content when read is always "0".

INIPRE: PRE trigger (D1) / SDRAM initial register (0x3A0202)

Executes the PALL (Precharge All) command.

Write "1": Trigger
Write "0": Invalid
Read: Always "0"

Writing "1" to INIPRE sends the PALL (Precharge All) command for initializing the SDRAM. See INIMRS command for the SDRAM initializing sequence.
Writing "0" results in No Operation.
Since INIPRE is a write-only bit, its content when read is always "0".

INIREF: REF trigger (D0) / SDRAM initial register (0x3A0202)

Executes the REF (Auto Refresh) command.

Write "1": Trigger
Write "0": Invalid
Read: Always "0"

Writing "1" to INIREF sends the REF (Auto Refresh) command for initializing the SDRAM. See INIMRS command for the SDRAM initializing sequence.
Writing "0" results in No Operation.
Since INIREF is a write-only bit, its content when read is always "0".

AURCO11–AURCO0: SDRAM auto-refresh counter (D[B:0]) / SDRAM auto-refresh counter register (0x3A0204)

Set the auto refresh counter value.
The auto-refresh counter counts up on the halt2run_clk clock edges beginning with 0, and when the count specified here is reached, the SDRAM controller sends an auto-refresh command. The counter is reset at that point, and starts counting the next refresh period. The counter is also reset by self-refresh.
The value calculated from the equation below is the maximum count that can be set.

$$AURCO \leq \frac{RFP}{ROWS} \times f_{CLK} - BL - CL - 2 \times t_{RP} - t_{RCD} - 3$$

RFP: Maximum refresh period [s]
ROWS: Row address size
f_{CLK}: halt2run_clk clock frequency [Hz]
BL: Burst length (= 2)
CL: CAS latency (= 2)
t_{RP}: PRECHARGE command period [Number of cycles]
t_{RCD}: ACTIVE to READ or WRITE delay time [Number of cycles]

At initial reset, AURCO is set to "0x0".

SELDO: SDRAM self-refresh status (D9) / SDRAM self-refresh counter register (0x3A0206)

Indicates the SDRAM self-refresh status.

Read "1": In self-refresh mode
Read "0": Not in self-refresh mode
Write: Invalid

SELDO is "1" while the SDRAM controller holds the SDCKE pin low (i.e., the SDRAM is in self-refresh mode). Otherwise, SELDO = "0".
Before entering SLEEP mode, always be sure to read this bit using a program stored elsewhere (i.e., not in the SDRAM) to confirm that the SDRAM is in self-refresh mode.
At initial reset, SELDO is set to "0" (Not in self-refresh mode).

SCKON: SDRAM clock during self-refresh (D8) / SDRAM self-refresh counter register (0x3A0206)

Select whether to stop the SDRAM clock during self-refresh or not.

Write "1": Enabled (not stopped)
Write "0": Disabled (stopped)
Read: Valid

Writing "0" to SCKON causes the SDRAM clock output from the SDCLK pin to stop and to remain off while the SDRAM is self-refreshed. This helps to reduce the chip's current consumption.
If SCKON = "1", the SDRAM clock is always output from the SDCLK pin even while the SDRAM is self-refreshed.
At initial reset, SCKON is set to "0" (disabled).

SELEN: SDRAM self-refresh enable (D7) / SDRAM self-refresh counter register (0x3A0206)

Enable the SDRAM's self-refresh control function.

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

Writing "1" to SELEN enables the SDRAM controller to start self-refreshing the SDRAM (by setting SDCKE output low). Note that self-refreshing of the SDRAM actually begins a certain time after accessing or auto-refreshing the SDRAM. The duration of this elapsed time is defined by the number of clock cycles in SELCO[6:0] (D[6:0]/0x3A0206).
SELEN = "0" disables the self-refresh function.
At initial reset, SELEN is set to "0" (disabled).

SELCO6–SELCO0: SDRAM self-refresh counter (D[6:0]) / SDRAM self-refresh counter register (0x3A0206)

Set the self-refresh counter value.
If SELEN (D7/0x3A0206) is set to "1" (self-refresh-enabled), the self-refresh counter starts counting up on the halt2run_clk clock edges beginning with 0 after accessing or auto-refreshing the SDRAM. When the count specified here is reached, the SDCKE output is pulled low, causing the SDRAM to start self-refreshing. If an access to the SDRAM occurs during self-refresh mode, SDCKE is returned high, thereby taking the SDRAM out of self-refresh mode.
At initial reset, SELCO is set to "0x7F".

T80NS1–T80NS0: SDRAM t_{RC}, t_{RFC}, and t_{XSR} cycles (D[2:1]) / SDRAM timing setup register (0x3A0208)

Sets the following SDRAM timing parameters.
- t_{RC} ACTIVE to ACTIVE command period
- t_{RFC} REFRESH command period
- t_{XSR} Exit SELF REFRESH to ACTIVE command period

Table VIII.2.15 Setting t_{RC}, t_{RFC}, and t_{XSR} Timing Parameters

T80NS1	T80NS0	t_{RC}, t_{RFC}, t_{XSR}
1	1	5 cycles
1	0	4 cycles
0	1	3 cycles
0	0	2 cycles

At initial reset, T80NS is set to "11" (5 cycles).

T24NS: SDRAM tRP and tRCD cycles (D0) / SDRAM timing setup register (0x3A0208)

Set the tRP and tRCD SDRAM timing parameters.

Write "1": 2 cycles
Write "0": 1 cycle
Read: Valid

- tRP PRECHARGE command period
- tRCD ACTIVE to READ or WRITE delay time

At initial reset, T24NS is set to "1" (2 cycles).

SDHP: SDRAM high-performance access (D3) / SDRAM application configuration register (0x3A0210)

Selects a high-performance access condition.

Write "1": High performance
Write "0": Low performance
Read: Valid

When SDHP is set to "0", the read/write signals from the C33 STD are used without a reshaping.
When SDHP is set to "1", the read/write signals are generated in the SDRAMC block according to the bus mode and bus size. So the SDRAMC operating speed will be relatively higher than in the case of "SDHP = 0". However when the SDRAM is used as the external VRAM and the bcuclk = halt2run_clk / (2, 4 or 8), the SDHP must be set to "0".
At initial reset, SDHP is set to "0" (low performance).

AR14C: Area 8/14 configuration (D2) / SDRAM application configuration register (0x3A0210)
AR13C: Area 7/13 configuration (D1) / SDRAM application configuration register (0x3A0210)

Selects whether the area is used for SDRAM or not.

Write "1": Used for SDRAM
Write "0": Not used for SDRAM
Read: Valid

A 16-bit SDRAM or two 8-bit SDRAMs can be connected to Areas 7–8 or Areas 13–14. Write "1" to AR13C to set Area 7 or Area 13 for SDRAM use. Similarly, write "1" to AR14C to set Area 8 or Area 14 for SDRAM use. Either or both areas (e.g. Area 7 + Area 8, Area 13 + Area 14) can be configured for SDRAM. Note, however, that only one chip enable is available even if both areas are configured for SDRAM.
When the bit is set to "0", the corresponding area is not used for SDRAM.
At initial reset, these bits are set to "0" (not used for SDRAM).

Programming Notes

(1) Set the area used for an SDRAM for external access (A8IO or A14IO (DA or DD/0x48132) = "0").

(2) Be sure to set SWAITE (D0/0x4812E) to "1" to enable the #WAIT signal control when the SDRAM controller is used. In this case, it is not necessary to switch the P30 pin function to a #WAIT input pin, note, however, that the P30 port function is disabled. If the P30 pin must be used as an I/O port, switch the pin function and use it as the PA2 port.

(3) Before entering SLEEP mode, be sure to place the SDRAM in self-refresh mode, because the SDRAM cannot be auto-refreshed while in this mode. In that case, confirm that SELDO (D9/0x3A0206) = "1" (i.e., that the SDRAM is in self-refresh mode) before executing the slp instruction.
If an access to the SDRAM occurs while being self-refreshed, the SDRAM is taken out of self-refresh mode; thus always make sure the SELDO check and the slp instruction execution are performed from devices other than the SDRAM.

VIII-3 INSTRUCTION/DATA QUEUE BUFFERS

Overview

The SDRAMC Block contains the SDRAMC Application Unit that is an interface module to connect between the C33 STD core and the SDRAM interface module described in Section VIII-2. It generates the read, write, address, data, and handshake signals to drive the SDRAM interface module. Besides generating these signals, it also includes a Data Queue Buffer and an Instruction Queue Buffer to realize the instruction pre-fetch function and to increase the C33 STD memory performance.

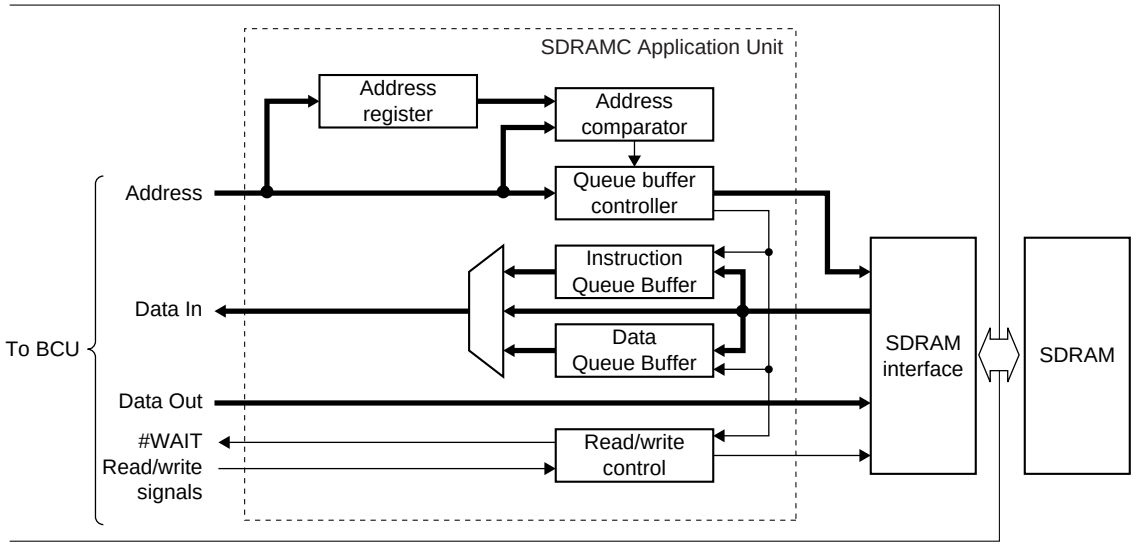


Figure VIII.3.1 Instruction/Data Queue Buffers

IQB (Instruction Queue Buffer)

This is a queue buffer to pre-fetch instructions and consists of a 16 × 16-bit dual port synchronous SRAM. It is organized in 2 slots × 8 × 16 bits as shown in the figure below.

A[24:14]	A[13:4]	A[3:1]							
IQB address	Slot 1 address	Slot 1							
		Buf 0	Buf 1	Buf 2	Buf 3	Buf 4	Buf 5	Buf 6	Buf 7
	Slot 2 address	Slot 2							
		Buf 0	Buf 1	Buf 2	Buf 3	Buf 4	Buf 5	Buf 6	Buf 7

Figure VIII.3.2 Structure of IQB

IQB acts as an instruction cache located between the CPU and SDRAM when it is enabled by setting the IQBON (D0) / SDRAM application configuration register (0x3A0210) to "1". When the CPU attempts to fetch the first instruction from the SDRAM after IQB is enabled, the SDRAMC Application Unit pre-fetches 8 instructions from the SDRAM (including CPU aimed instructions) and stores them in IQB's 1st slot. The CPU then gets the needed instruction from IQB. After that, the CPU can get the subsequent instructions to be executed from IQB if IQB contains them (called as IQB Hit). If IQB does not contain the instruction to be executed next (called as IQB Not Hit), the SDRAMC Application Unit pre-fetches another 8 instructions (including CPU aimed instructions) from the SDRAM and stores them in the IQB's 2nd slot, then the CPU gets the needed instruction from the IQB's 2nd slot. The two slots are used alternately like this and the CPU continues fetching instructions from IQB while the routine to be executed is located in the SDRAM.

Each slot can store 8 instructions, so IQB always stores the pre-fetched data beginning with a 4-word (128 bits) boundary address.

Note: When the CPU aimed instruction's address A[3:2] = "11" and when IQB is not hit, instructions will be fetched from Data Queue Buffer, rather than the IQB.

CPU aimed instruction address A[3:1]	Slot 1, 2							
	Buf 0	Buf 1	Buf 2	Buf 3	Buf 4	Buf 5	Buf 6	Buf 7
000	■							
001		■						
010			■					
011				■				
100					■			
101						■		
110	Fetched from DQB							
111								

■ CPU aimed instruction IQB pre-fetched instruction

Figure VIII.3.3 Instructions in Slot

When IQB is disabled (IQBON (D0/0x3A0210) = "0"), the CPU can only fetch the instruction or data from the SDRAM through the Data Queue Buffer described below.

DQB (Data Queue Buffer)

The DQB consists of two-stage 16-bit buffers and is mainly used to store data read from the SDRAM and to decrease the SDRAM read latency. Provided two stages, defined as Buffer 0 and Buffer 1, correspond to two-burst reading for the SDRAM.
When IQBON (D0/0x3A0210) is set to "1" to enable IQB, DQB acts as a data buffer in which Data in the program is stored (Instructions will also be stored in the special case as described in Note on the previous page). When IQBON (D0/0x3A0210) is set to "0" to disable IQB, DQB acts as a pure read buffer in which all data read from the SDRAM are stored without distinction between Instructions and Data. Note that DQB cannot be disabled.
Table VIII.3.1 lists the DQB status corresponding to the bus operation for the SDRAM.

Table VIII.3.1 DQB Status Corresponding to Bus Operation

Bus operation	DQB status (active or inactive)	
	When IQB is enabled	When IQB is disabled
CPU Instruction Fetch	Active only when fetching the instruction at the address of A[3:2] = "11" and IQB is not hit	Active
CPU Vector Fetch	Active	Active
CPU Data Read	Active	Active
CPU Data Write	Inactive	Inactive
CPU Stack Read	Active	Active
CPU Stack Write	Inactive	Inactive
DMA Data Read	Active	Active
DMA Data Write	Inactive	Inactive

The SDRAM interface always reads two successive half-word data beginning with a 32-bit boundary address in burst reading. Therefore, 16-bit data at a word-boundary address (A[1:0] is corrected to "00") is always stored in Buffer 0, and the next read data at the subsequent half-word boundary address (A[1:0] = "10") is always stored in Buffer 1.
If DQB contains the needed data when the CPU reads data from the SDRAM, no SDRAM read cycle is generated and data is read from DQB.

Operations using IQB/DQB

Reading SDRAM Data

In order to judge the IQB/DQB Hit or IQB/DQB Not Hit, the SDRAMC Application Unit has an address register that holds the address in which data is buffered into IQB/DQB and an address comparator for comparing between the CPU read address and the address register.
The address comparator compares the SDRAM address being accessed from the CPU with the address data in the address register, and issues Hit if they are matched or Not Hit if they are not matched.
After an initial reset, the IQB/DQB and address register statuses are reset as Empty. Therefore, the CPU's first fetching/reading of the SDRAM always causes Not Hit.

In the case of Not Hit:

The internal wait signal is output to the C33 STD core and the BCU bus cycle enters a wait state. The SDRAMC Application Unit reads data from the SDRAM through the SDRAM interface and stores it into IQB or DQB. When the buffer is ready to read, the internal wait signal is negated and the BCU bus cycle reads the data from the IQB or DQB.

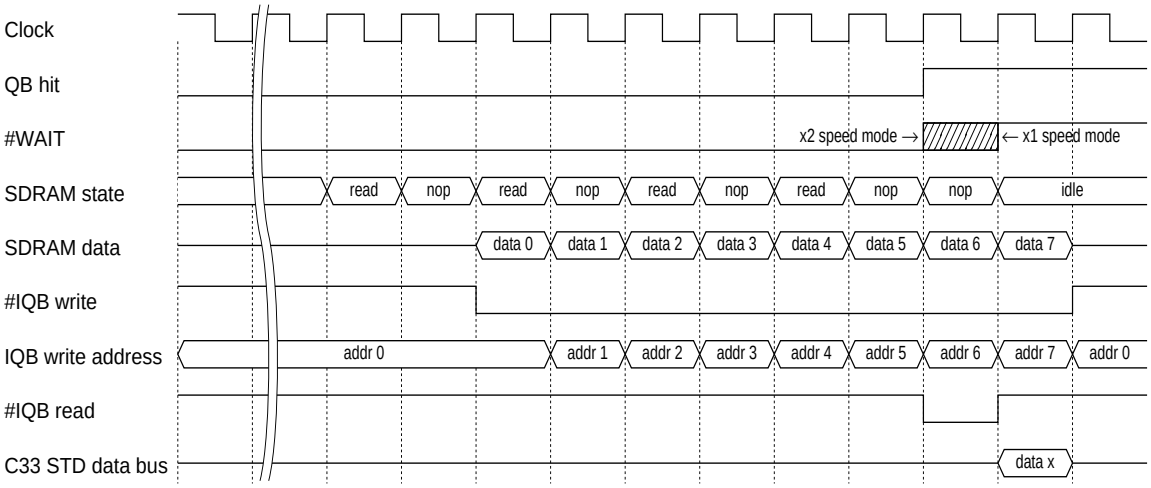


Figure VIII.3.4 CPU Instruction Fetch Cycle for IQB Not Hit

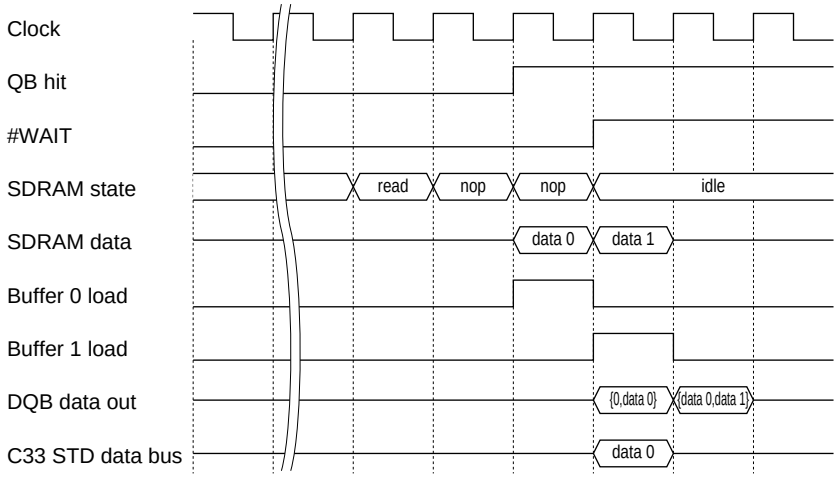


Figure VIII.3.5 CPU Data Read Cycle for DQB Not Hit

In the case of Hit:

No SDRAM access is generated and the CPU can fetch or read the instruction or data from IQB or DQB with no wait state inserted.

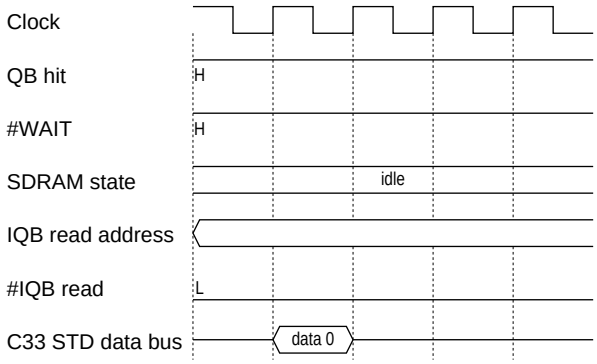


Figure VIII.3.6 CPU Read Cycle with Hit

Writing Data to SDRAM

When the CPU writes data to the SDRAM, the internal wait signal input to the C33 STD core is asserted until the SDRAM interface has finished writing to the SDRAM.
If data is written to the address in which data has buffered in IQB or DQB, the related buffer data is flushed.

I/O Memory of IQB/DQB

Table VIII.3.2 shows the control bit of the IQB/DQB.

Table VIII.3.2 Control Bits of IQB/DQB

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SDRAM application configuration register	**03A0210 (HW)**	DF–4	**–**	reserved	–				–	–	0 when being read.
		D3	**SDHP**	SDRAM high-performance access	1	High perform.	0	Low perform.	0	R/W	
		D2	**AR14C**	Area 8/14 configuration	1	SDRAM	0	Other	0	R/W	
		D1	**AR13C**	Area 7/13 configuration	1	SDRAM	0	Other	0	R/W	
		D0	**IQBON**	Instruction queue buffer enable	1	Enabled	0	Disabled	0	R/W	

IQBON: Instruction queue buffer enable (D0) / SDRAM application configuration register (0x3A0210)

Enables the IQB (Instruction Queue Buffer).

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

Setting IQBON to "1" enables IQB and instructions stored in the SDRAM are pre-fetched into the IQB. The IQB acts as a high-speed instruction cache for the CPU. In this case, DQB is used as a data read buffer. When IQBON is set to "0", IQB is disabled and DQB is used as an instruction/data buffer.
At initial reset, IQBON is set to "0" (disabled).

VIII-4 BUS ARBITER

Overview

The LCDC may use the first 1MB (max.) area in the SDRAM as the VRAM. To prevent a reduction in LCD display quality, the LCDC can access the SDRAM (VRAM) directly while the C33 STD accesses another device. This is realized with the bus arbiter embedded in the UMA (Unified Memory Access) system. The bus arbiter arbitrates the ownership of the external bus between the LCDC and the BCU.
The following lists the features of the bus arbiter:

- Supports only
  Non-SDRAM area: 32 MHz operation frequency in x1 speed mode
  (one read hold cycle and one wait cycle must be inserted)
  Non-SDRAM area: 48 MHz operation frequency in x2 speed mode
  (one wait cycle and one output disable cycle must be inserted)
  SDRAM area: 32 MHz operation frequency in x1 speed mode
  (one wait cycle and one output disable cycle must be inserted)
  SDRAM area: 48 MHz operation frequency in x2 speed mode
  (a zero wait cycle can be set)
- Single clock edge operation
- Arbitrates among up to four bus masters.
- Park/grant logic
  When no master requires the external bus, one default master (selected by using a register) will be parked to the external bus.
- Bus lock function
  The ownership of the external bus cannot be interrupted even if another master module with a higher priority asserts the bus request signal.
- Supports a BCU interface called BCU bridge.
- The following functions in the C33 STD core cannot be used while the bus arbiter is enabled:
  - External burst-ROM function
  - Bus request function using the #BUSREQ/#BUSACK signals
  - EDO-DRAM

  If one of these functions is used in the system, the bus arbiter should be disabled, otherwise a malfunction will occur.

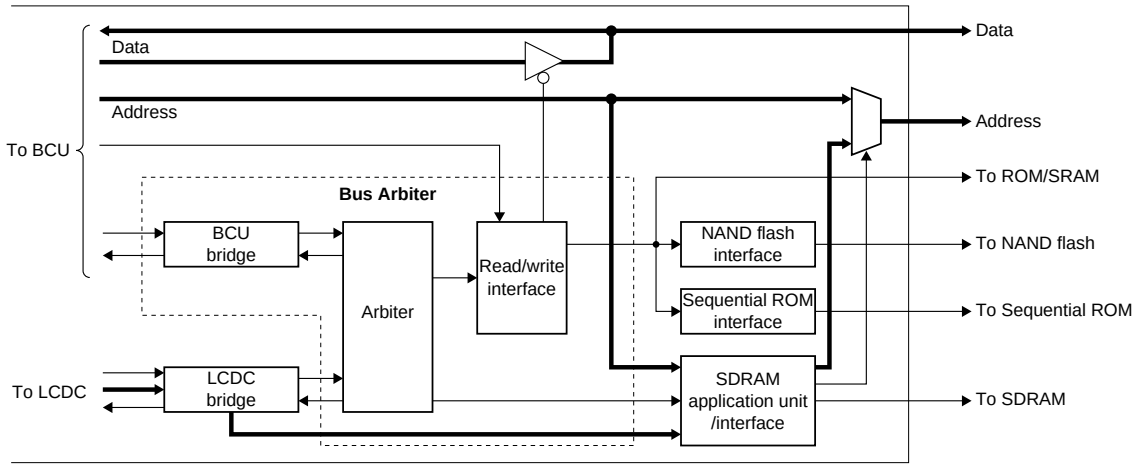


Figure VIII.4.1 Bus Arbiter Block Diagram

Controlling the Bus Arbiter

When using the external SDRAM as the VRAM, the bus arbiter must be enabled so that both the LCDC and BCU can access the SDRAM. Before enabling the bus arbiter, it is necessary to set the conditions shown below.

Setting the BCU

The control registers for the bus arbiter are mapped into Area 6. Therefore, in order for the control registers to be accessed, the BCU must be set up in accordance with the procedure described below.

1. A6IO (D9) / Access control register (0x48132) = "1"
   Configure Area 6 for an internal device area.
2. A6WT[2:0] (D[A:8]) / Areas 6–4 setup register (0x4812A)
   The number of wait cycles for Area 6 can be set to 0 when the CPU runs with a 48 MHz clock in x2 speed mode or a 32 MHz clock in x1 speed mode.
3. SWAITE (D0) / Bus control register (0x4812E) = "1"
   Enable the #WAIT signal.
4. A6EC (D1) / Access control register (0x48132) = "0"
   Select little endian as the Area 6 data format.
5. AxxRH (D[7:0]) / Read cycle hold time control register (0x4813C)
   When using the bus arbiter and accessing non-SDRAM type external devices (external ROM/SRAM, NAND flash, Sequential ROM, etc) in x1 speed mode, configure the Read cycle hold time control register (0x4813C) so that one read hold cycle will be inserted when the areas allocated to the external devices are accessed.
6. AxxWT[2:0] and AxxDF[1:0] / Area xx setup registers
   When using the bus arbiter and accessing the external devices (SDRAM, external ROM/SRAM, NAND flash, Sequential ROM, etc.), configure the wait cycle control bits (AxxWT[2:0]) and output disable delay cycle control bits (AxxDF[1:0]) in the Area xx setup register so that wait cycles and output disable delay cycles will be inserted when the areas allocated to the related external devices are accessed. (The number of wait cycles and output disable delay cycles must be set to at least the minimum numbers required according to the operating speed, bus speed mode and device type (SDRAM or other).)
7. Only the "ld.h" or "ld.uh" instructions can be used to access the bus arbiter control registers located in Area 6.
8. CFP30 (D0) / P3 function select register (0x402DC) = "1"
   Set the P30 port for the #WAIT function.
9. SBUSST (D3) / Bus control register (0x4812E) = BSLSEL (D0) / A0/#BSL select register (0x300F38)
   Be sure to set the same external interface method between the SBUSST (D3/0x4812E) in a BCU register and the BSLSEL (D0/0x300F38) in a Misc register.
   Before writing to the A0/#BSL select register (0x300F38), Misc register write protection should be removed.

Controlling the Operating Clock

The bus arbiter operates with the sbcuclk clock equivalent to the BCU clock, so it is not necessary to set up the clock independently. This clock stops when the CPU enters HALT2 or SLEEP mode. Furthermore, the software can stop supplying the clock to the bus arbiter in order to reduce current consumption when the bus arbiter is not used. To stop supplying the clock, set the ARBCKOF (D1) / Module clock control register 0 (0x300F35) to "1". ARBCKOF is initialized to "0" after an initial reset and the clock is supplied to the bus arbiter. Do not set it to "1" while the bus arbiter is enabled. Otherwise, a malfunction will occur.

Refer to Section III-17, "S1C33L05 Clock System and Miscellaneous Registers", for details on the clock control.

Selecting the Default Bus Master

The bus arbiter grants the ownership of the external bus to a bus master (BCU or LCDC) according to the bus request from bus masters. When no bus masters request the ownership of the external bus, the default bus master is parked on the external bus. The default bus master can be selected using the PARKID[1:0] (D[3:2]) / Bus arbiter control register (0x390000).

Table VIII.4.1 Selecting the Default Bus Master

PARKID1	PARKID0	Bus master ID
1	1	Master 3 (reserved)
1	0	Master 2 (reserved)
0	1	Master 1 (LCDC)
0	0	Master 0 (BCU)

At initial reset, Master 0 (BCU) is set as the default bus master.

Note: When setting the CPU in HALT2 mode, be sure to set PARKID to "01" (LCDC) before executing the halt instruction. Otherwise a malfunction will occur. After the CPU returns to normal mode, the PARKID can be reset to "00" (BCU).

When the bus arbiter is not used (disabled), only the BCU controls the external bus.

Setting the Number of Wait Cycles

In order to prevent a malfunction, additional wait cycles will be inserted into the first bus cycle every time the ownership of the external bus shifts from other bus masters to the BCU. Use the ARBWT[3:0] (D[3:0]) / Bus arbiter wait control register (0x390002) to set the number of wait cycles to be inserted. The ARBWT value must be specified according to the bus speed mode (x1 speed mode or x2 speed mode).
x1 speed mode: Number of wait cycles = ARBWT[3:0] (0 to 7 cycles)
x2 speed mode: Number of wait cycles = (ARBWT[3:0] + 1) / 2 (0 to 15 cycles)

Table VIII.4.2 ARBWT[3:0] Settings

Number of wait cycles	ARBWT[3:0]	
	x1 speed mode	x2 speed mode
0	0	0
1	1	1
2	2	3
3	3	5
4	4	7
5	5	9
6	6	11
7	7	13

Enabling the Bus Arbiter

After setting the above conditions, write "1" to the ARBIEN (D0) / Bus arbiter control register (0x390000) to enable the bus arbiter.
The following functions in the C33 STD core cannot be used while the bus arbiter is enabled:
- External burst-ROM function
- Bus request function using the #BUSREQ/#BUSACK signals
- EDO-DRAM

If one of these functions is used in the system, the bus arbiter should be disabled, otherwise a malfunction will occur.

Timing Charts

When the CPU accesses an internal device or is in HALT status, and the LCDC reads data from the SDRAM:

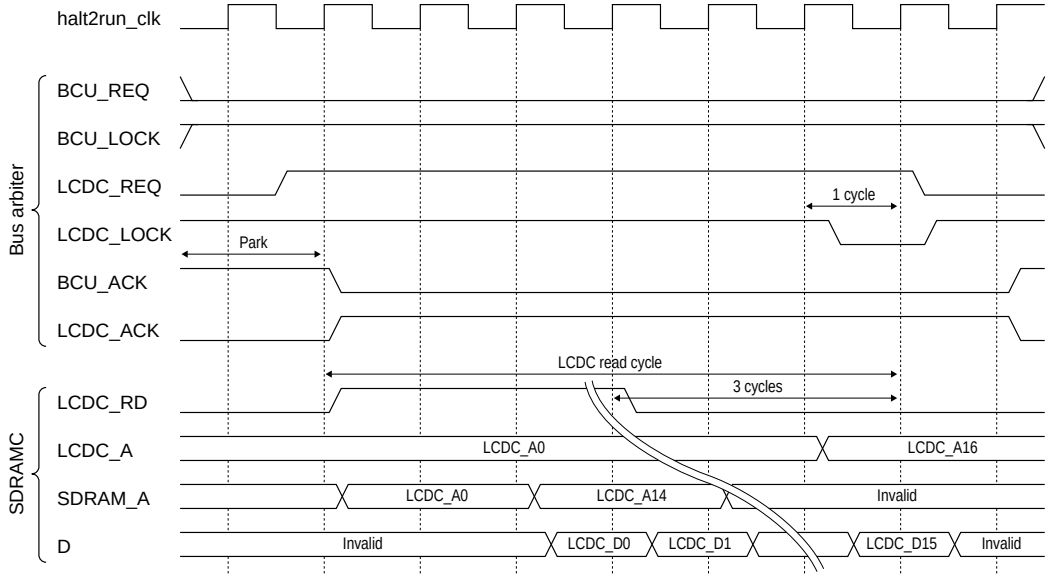


Figure VIII.4.2 Timing Chart 1 (BCU: idle, LCDC: reads SDRAM)

When the BCU and LCDC read data from the SDRAM:

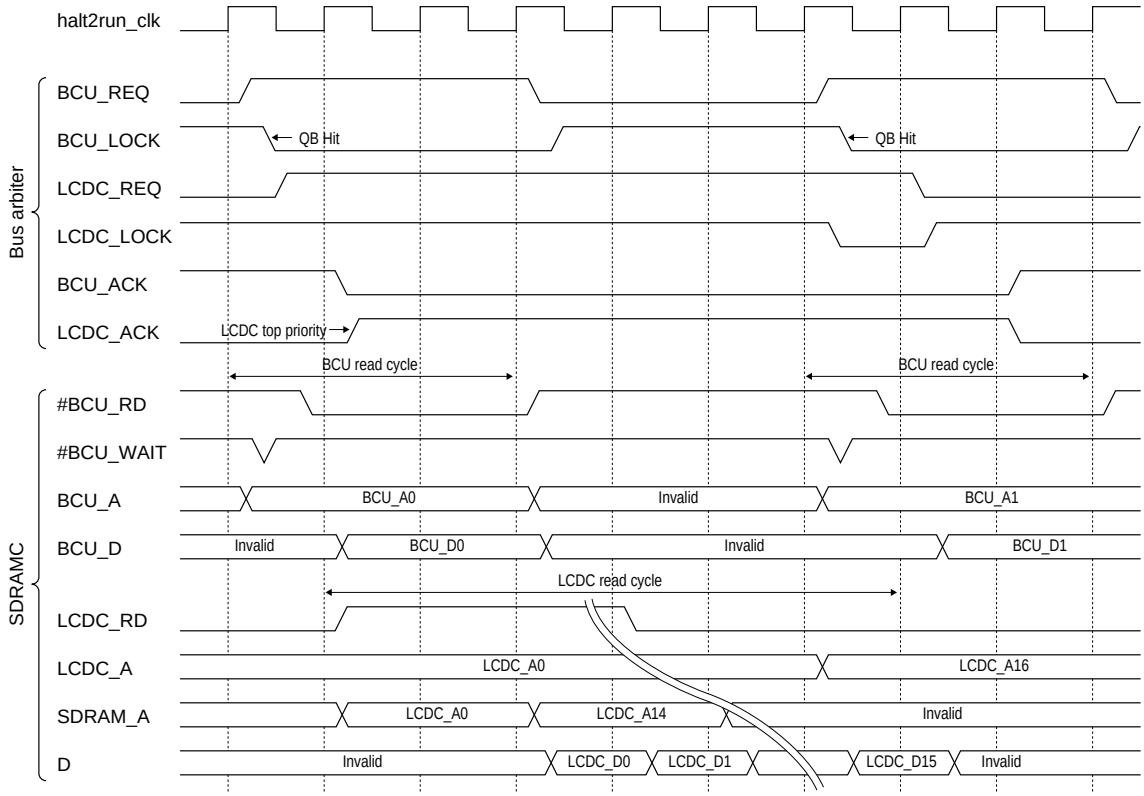


Figure VIII.4.3 Timing Chart 2 (BCU: reads SDRAM with QB Hit, LCDC: reads SDRAM)

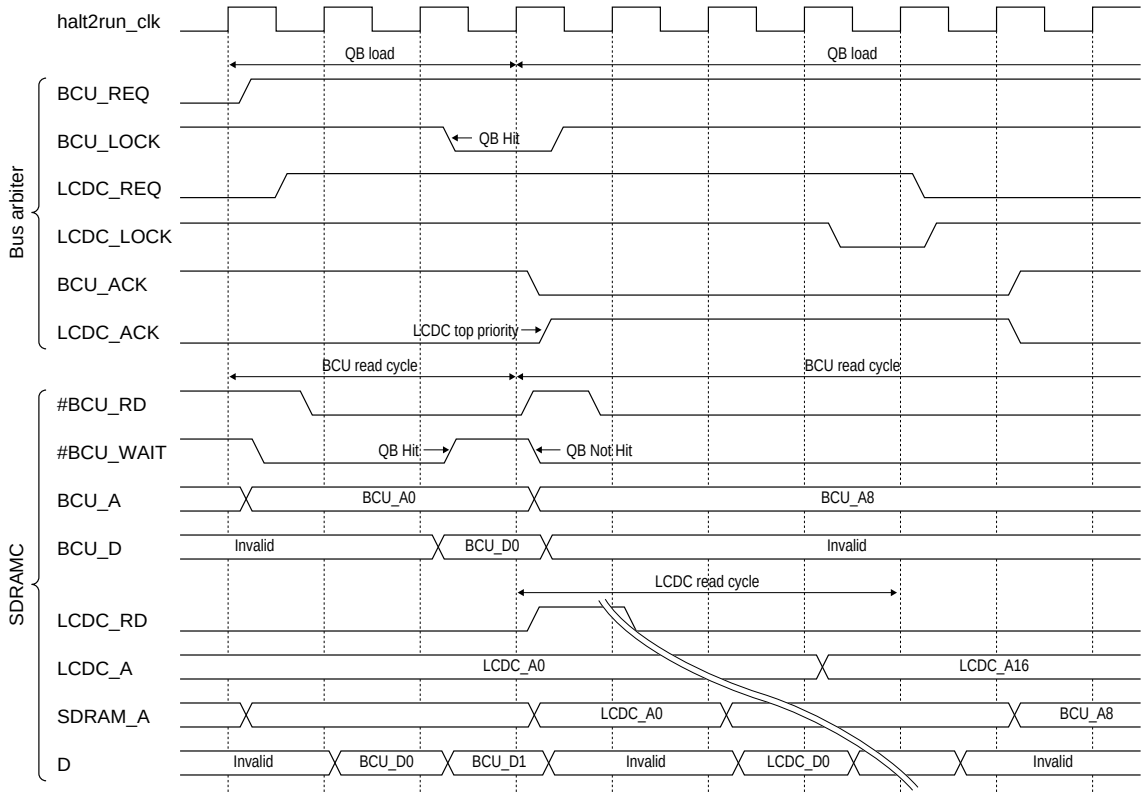


Figure VIII.4.4 Timing Chart 3 (BCU: reads SDRAM with QB Not Hit, LCDC: reads SDRAM)

When the BCU reads data from an external device and the LCDC reads data from the SDRAM:

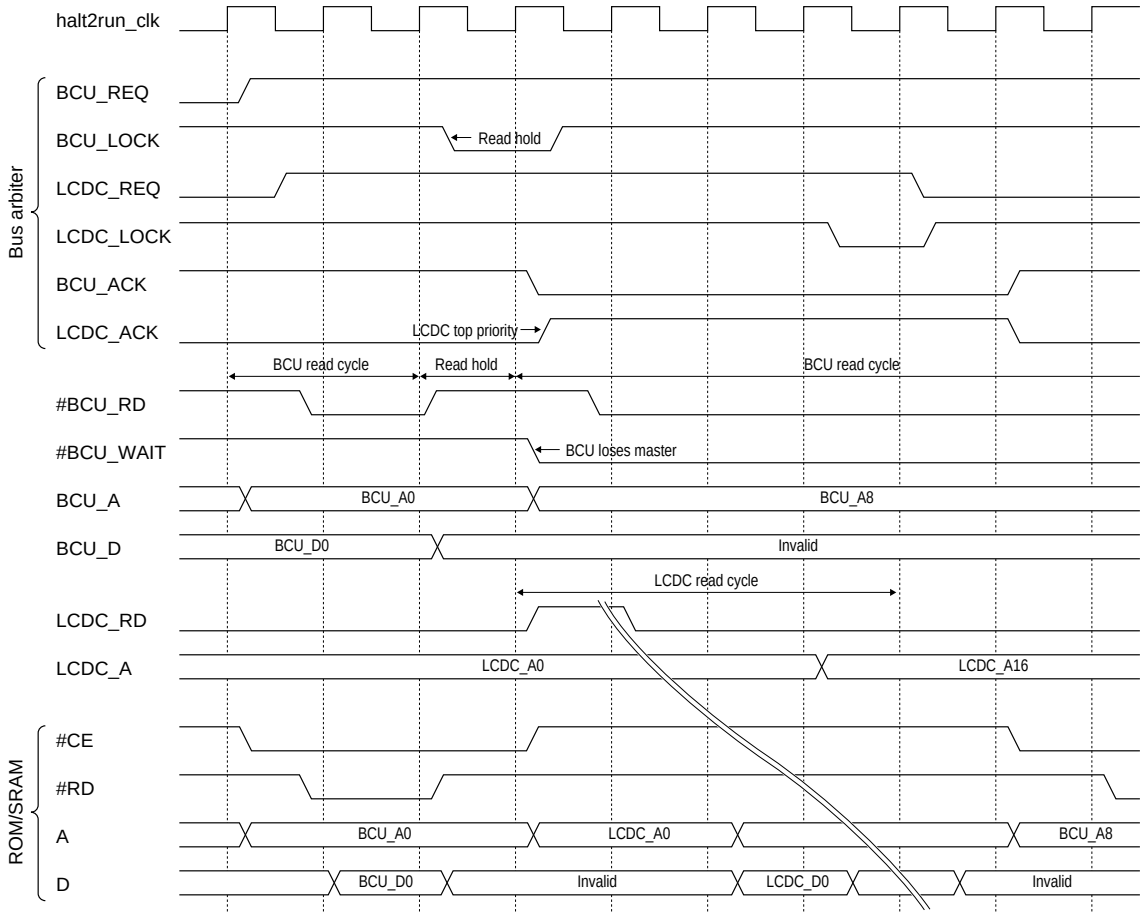


Figure VIII.4.5 Timing Chart 4 (BCU: reads an external device in x1 speed mode, LCDC: reads SDRAM)

When the BCU writes data to an external device and the LCDC reads data from the SDRAM:

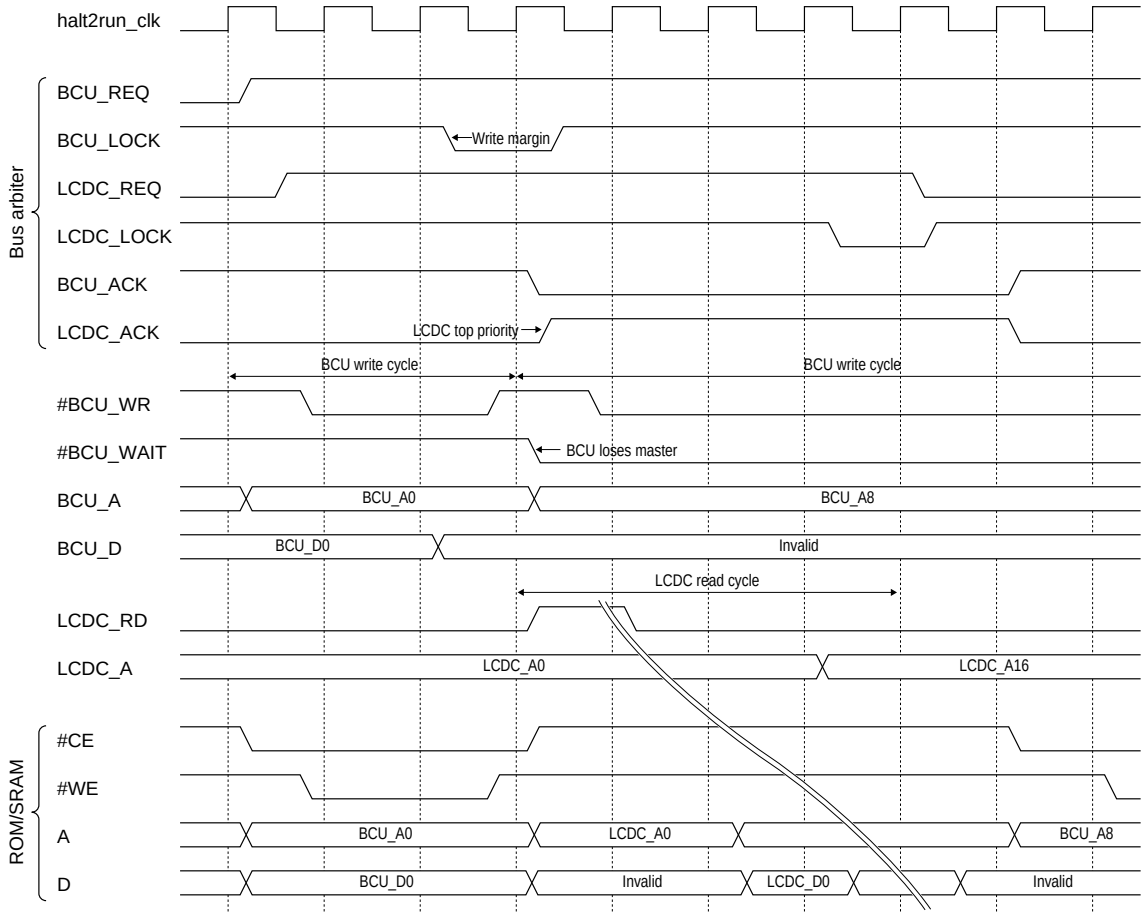


Figure VIII.4.6 Timing Chart 5 (BCU: writes to an external device, LCDC: reads SDRAM)

I/O Memory of Bus Arbiter

Table VIII.4.3 shows the control bits of the bus arbiter.

Table VIII.4.3 Control Bits of Bus Arbiter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Bus arbiter control register	**0390000 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**PARKID1**	Parked bus master ID	PARKID[1:0] Bus master ID	0	R/W	
		D2	**PARKID0**		1 1 3 (reserved)	0		
					1 0 2 (reserved)			
					0 1 1 (LCDC)			
					0 0 0 (BCU)			
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**ARBIEN**	Bus arbiter enable	1 Enabled 0 Disabled	0	R/W	
Bus arbiter wait control register	**0390002 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**ARBWT3**	Bus arbiter wait control	0 to 15	1	R/W	
		D2	**ARBWT2**			1		
		D1	**ARBWT1**			1		
		D0	**ARBWT0**			1		

PARKID1–PARKID0: Parked bus master ID (D[3:2]) / Bus arbiter control register (0x390000)

Sets the default bus master that will get the ownership of the external bus when it is not requested from any bus master.

Table VIII.4.4 Setting the Default Bus Master

PARKID1	PARKID0	Bus master ID
1	1	Master 3 (reserved)
1	0	Master 2 (reserved)
0	1	Master 1 (LCDC)
0	0	Master 0 (BCU)

PARKID is only effective when the bus arbiter is enabled. When the bus arbiter is disabled, Master 0 is parked on the external bus.
At initial reset, PARKID is set to "00" (Master 0).

ARBIEN: Bus arbiter enable (D0) / Bus arbiter control register (0x390000)

Enables the bus arbiter.

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

When ARBIEN is set to "1", the bus arbiter is enabled and can arbitrate the external bus requests from the bus masters. When ARBIEN is set to "0", the bus arbiter is disabled. In this case, only the BCU can use the bus.
Be sure to set ARBIEN to "1" when using the SDRAM as the VRAM.
At initial reset, ARBIEN is set to "0" (disabled).

ARBWT3–ARBWT0: Bus arbiter wait control (D[3:0]) / Bus arbiter wait control register (0x390002)

Specifies the number of wait cycles that will be inserted in the external bus operation by the BCU.
The wait cycles set using ARBWT will be inserted only when the bus arbiter grants the ownership of the external bus from other bus masters to the BCU.
The number of wait cycles to be inserted should be set as follows according to the bus speed mode (x1 speed mode or x2 speed mode) that has been set:
x1 speed mode: Number of wait cycles = ARBWT[3:0] (0 to 7 cycles)
x2 speed mode: Number of wait cycles = (ARBWT[3:0] + 1) / 2 (0 to 15 cycles)

At initial reset, ARBWT is set to "0xF".

Programming Notes

(1) The following functions in the C33 STD core cannot be used while the bus arbiter is enabled:
- External burst-ROM function
- Bus request function using the #BUSREQ/#BUSACK signals
- EDO-DRAM

If one of these functions is used in the system, the bus arbiter should be disabled, otherwise a malfunction will occur.

(2) The ARBWT value that specifies the number of wait cycles for BCU's bus operation must be changed according to the bus speed mode (x1 speed mode or x2 speed mode).
x1 speed mode: Number of wait cycles = ARBWT[3:0]
x2 speed mode: Number of wait cycles = (ARBWT[3:0] + 1) / 2

(3) When setting the CPU in HALT2 mode, be sure to set PARKID to "01" (LCDC) before executing the halt instruction. Otherwise a malfunction will occur. After the CPU returns to normal mode, the PARKID can be reset to "00" (BCU).

THIS PAGE IS BLANK.

S1C33L05 Technical Manual

Appendix

A 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 θ/1 0 Divided clk.	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D0	**P8TPCK4**	8-bit timer 4 clock selection	1 θ/1 0 Divided clk.	0	R/W	
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 θ/256	0	R/W	θ: selected by Prescaler clock select register (0x40181) 8-bit timer 5 can generate the clock for the serial I/F Ch.3.
		D5	**P8TS51**	clock division ratio selection	1 1 0 θ/128	0	R/W	
		D4	**P8TS50**		1 0 1 θ/64	0	R/W	
					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	**P8TON4**	8-bit timer 4 clock control	1 On 0 Off	0	R/W	
		D2	**P8TS42**	8-bit timer 4	1 1 1 θ/4096	0	R/W	θ: selected by Prescaler clock select register (0x40181) 8-bit timer 4 can generate the clock for the serial I/F Ch.2.
		D1	**P8TS41**	clock division ratio selection	1 1 0 θ/2048	0	R/W	
		D0	**P8TS40**		1 0 1 θ/64	0	R/W	
					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			
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 θ/1 0 Divided clk.	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D2	**P8TPCK2**	8-bit timer 2 clock selection	1 θ/1 0 Divided clk.	0	R/W	
		D1	**P8TPCK1**	8-bit timer 1 clock selection	1 θ/1 0 Divided clk.	0	R/W	
		D0	**P8TPCK0**	8-bit timer 0 clock selection	1 θ/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	θ: selected by Prescaler clock select register (0x40181) 16-bit timer 0 can be used as a watchdog timer.
		D1	**P16TS01**	clock division ratio selection	1 1 1 θ/4096	0		
		D0	**P16TS00**		1 1 0 θ/1024	0		
					1 0 1 θ/256			
					1 0 0 θ/64			
					0 1 1 θ/16			
					0 1 0 θ/4			
					0 0 1 θ/2			
					0 0 0 θ/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	θ: selected by Prescaler clock select register (0x40181)
		D1	**P16TS11**	clock division ratio selection	1 1 1 θ/4096	0		
		D0	**P16TS10**		1 1 0 θ/1024	0		
					1 0 1 θ/256			
					1 0 0 θ/64			
					0 1 1 θ/16			
					0 1 0 θ/4			
					0 0 1 θ/2			
					0 0 0 θ/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	θ: selected by Prescaler clock select register (0x40181)
		D1	**P16TS21**	clock division ratio selection	1 1 1 θ/4096	0		
		D0	**P16TS20**		1 1 0 θ/1024	0		
					1 0 1 θ/256			
					1 0 0 θ/64			
					0 1 1 θ/16			
					0 1 0 θ/4			
					0 0 1 θ/2			
					0 0 0 θ/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.

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]: 111 θ/4096; 110 θ/1024; 101 θ/256; 100 θ/64; 011 θ/16; 010 θ/4; 001 θ/2; 000 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**P16TS31**			0		
		D0	**P16TS30**			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 clock division ratio selection	P16TS4[2:0]: 111 θ/4096; 110 θ/1024; 101 θ/256; 100 θ/64; 011 θ/16; 010 θ/4; 001 θ/2; 000 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**P16TS41**			0		
		D0	**P16TS40**			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 clock division ratio selection	P16TS5[2:0]: 111 θ/4096; 110 θ/1024; 101 θ/256; 100 θ/64; 011 θ/16; 010 θ/4; 001 θ/2; 000 θ/1	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**P16TS51**			0		
		D0	**P16TS50**			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 clock division ratio selection	P8TS1[2:0]: 111 θ/4096; 110 θ/2048; 101 θ/1024; 100 θ/512; 011 θ/256; 010 θ/128; 001 θ/64; 000 θ/32	0	R/W	θ: selected by Prescaler clock select register (0x40181) 8-bit timer 1 can generate the OSC3 oscillation-stabilize waiting period.
		D5	**P8TS11**			0		
		D4	**P8TS10**			0		
		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]: 111 θ/256; 110 θ/128; 101 θ/64; 100 θ/32; 011 θ/16; 010 θ/8; 001 θ/4; 000 θ/2	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**P8TS01**			0		
		D0	**P8TS00**			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: 111 θ/256; 110 θ/128; 101 θ/64; 100 θ/32; 011 θ/16; 010 θ/8; 001 θ/4; 000 θ/2	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**			0		
		D4	**P8TS30**			0		
		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: 111 θ/4096; 110 θ/2048; 101 θ/64; 100 θ/32; 011 θ/16; 010 θ/8; 001 θ/4; 000 θ/2	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**			0		
		D0	**P8TS20**			0		
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: 111 θ/256; 110 θ/128; 101 θ/64; 100 θ/32; 011 θ/16; 010 θ/8; 001 θ/4; 000 θ/2	0	R/W	θ: selected by Prescaler clock select register (0x40181)
		D1	**PSAD1**			0		
		D0	**PSAD0**			0		
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: 111 None; 110 Day; 101 Hour; 100 Minute; 011 1 Hz; 010 2 Hz; 001 8 Hz; 000 32 Hz	X	R/W	
		D6	**TCISE1**			X		
		D5	**TCISE0**			X		
		D4	**TCASE2**	Clock timer alarm factor selection	TCASE[2:0] Alarm factor: 1XX Day; X1X Hour; XX1 Minute; 000 None	X	R/W	
		D3	**TCASE1**			X		
		D2	**TCASE0**			X		
		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		

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 (low-order 8 bits)	0 to 65535 days (low-order 8 bits)	X	R/W	
		D6	**TCND6**			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 (high-order 8 bits)	0 to 65535 days (high-order 8 bits)	X	R/W	
		D6	**TCND14**			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 (Note) Can be set within 0–63.	X	R/W	
		D4	**TCCH4**			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 (Note) Can be set within 0–31.	X	R/W	
		D3	**TCCD3**	TCCD4 = MSB		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.

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	0 to 255	X	R/W	
		D6	**RLD46**	RLD47 = MSB		X		
		D5	**RLD45**	RLD40 = LSB		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	0 to 255	X	R	
		D6	**PTD46**	PTD47 = MSB		X		
		D5	**PTD45**	PTD40 = LSB		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	0 to 255	X	R/W	
		D6	**RLD56**	RLD57 = MSB		X		
		D5	**RLD55**	RLD50 = LSB		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	0 to 255	X	R	
		D6	**PTD56**	PTD57 = MSB		X		
		D5	**PTD55**	PTD50 = LSB		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	CLKDT[1:0] Division ratio: 1 1 1/8; 1 0 1/4; 0 1 1/2; 0 0 1/1	0	R/W	
		D6	**CLKDT0**			0		
		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.	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	0x0 to 0xFF(0x7F)	X	R/W	7-bit asynchronous mode does not use TXD07.
		D6	**TXD06**	TXD07(06) = MSB		X		
		D5	**TXD05**	TXD00 = LSB		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	0x0 to 0xFF(0x7F)	X	R	7-bit asynchronous mode does not use RXD07 (fixed at 0).
		D6	**RXD06**	RXD07(06) = MSB		X		
		D5	**RXD05**	RXD00 = LSB		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	**–**	–	–	–	–	0 when being read.
		D5	**TEND0**	Ch.0 transmit-completion flag	1 Transmitting 0 End	0	R	
		D4	**FER0**	Ch.0 framing 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	**SSCK0**	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	0x0 to 0xFF(0x7F)	X	R/W	7-bit asynchronous mode does not use TXD17.
		D6	**TXD16**	TXD17(16) = MSB		X		
		D5	**TXD15**	TXD10 = LSB		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	0x0 to 0xFF(0x7F)	X	R	7-bit asynchronous mode does not use RXD17 (fixed at 0).
		D6	**RXD16**	RXD17(16) = MSB		X		
		D5	**RXD15**	RXD10 = LSB		X		
		D4	**RXD14**			X		
		D3	**RXD13**			X		
		D2	**RXD12**			X		
		D1	**RXD11**			X		
		D0	**RXD10**			X		
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 framing 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	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.1 control register	**00401E8 (B)**	D7	**TXEN1**	Ch.1 transmit enable	1 Enabled / 0 Disabled	0	R/W	
		D6	**RXEN1**	Ch.1 receive enable	1 Enabled / 0 Disabled	0	R/W	
		D5	**EPR1**	Ch.1 parity enable	1 With parity / 0 No parity	X	R/W	Valid only in asynchronous mode.
		D4	**PMD1**	Ch.1 parity mode selection	1 Odd / 0 Even	X	R/W	
		D3	**STPB1**	Ch.1 stop bit selection	1 2 bits / 0 1 bit	X	R/W	
		D2	**SSCK1**	Ch.1 input clock selection	1 #SCLK1 / 0 Internal clock	X	R/W	
		D1	**SMD11**	Ch.1 transfer mode selection	SMD1[1:0] Transfer mode	X	R/W	
		D0	**SMD10**		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	**DIVMD1**	Ch.1 async. clock division ratio	1 1/8 / 0 1/16	X	R/W	
		D3	**IRTL1**	Ch.1 IrDA I/F output logic inversion	1 Inverted / 0 Direct	X	R/W	Valid only in asynchronous mode.
		D2	**IRRL1**	Ch.1 IrDA I/F input logic inversion	1 Inverted / 0 Direct	X	R/W	
		D1	**IRMD11**	Ch.1 interface mode selection	IRMD1[1:0] I/F mode	X	R/W	
		D0	**IRMD10**		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	**TXD27**	Serial I/F Ch.2 transmit data	0x0 to 0xFF(0x7F)	X	R/W	
		D6	**TXD26**	TXD27(26) = MSB		X		
		D5	**TXD25**	TXD20 = LSB		X		
		D4	**TXD24**			X		
		D3	**TXD23**			X		
		D2	**TXD22**			X		
		D1	**TXD21**			X		
		D0	**TXD20**			X		
Serial I/F Ch.2 receive data register	**00401F1 (B)**	D7	**RXD27**	Serial I/F Ch.2 receive data	0x0 to 0xFF(0x7F)	X	R	
		D6	**RXD26**	RXD27(26) = MSB		X		
		D5	**RXD25**	RXD20 = LSB		X		
		D4	**RXD24**			X		
		D3	**RXD23**			X		
		D2	**RXD22**			X		
		D1	**RXD21**			X		
		D0	**RXD20**			X		
Serial I/F Ch.2 status register	**00401F2 (B)**	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**TEND2**	Ch.2 transmit-completion flag	1 Transmitting / 0 End	0	R	
		D4	**FER2**	Ch.2 framing error flag	1 Error / 0 Normal	0	R/W	Reset by writing 0.
		D3	**PER2**	Ch.2 parity error flag	1 Error / 0 Normal	0	R/W	Reset by writing 0.
		D2	**OER2**	Ch.2 overrun error flag	1 Error / 0 Normal	0	R/W	Reset by writing 0.
		D1	**TDBE2**	Ch.2 transmit data buffer empty	1 Empty / 0 Buffer full	1	R	
		D0	**RDBF2**	Ch.2 receive data buffer full	1 Buffer full / 0 Empty	0	R	
Serial I/F Ch.2 control register	**00401F3 (B)**	D7	**TXEN2**	Ch.2 transmit enable	1 Enabled / 0 Disabled	0	R/W	
		D6	**RXEN2**	Ch.2 receive enable	1 Enabled / 0 Disabled	0	R/W	
		D5	**EPR2**	Ch.2 parity enable	1 With parity / 0 No parity	X	R/W	Valid only in asynchronous mode.
		D4	**PMD2**	Ch.2 parity mode selection	1 Odd / 0 Even	X	R/W	
		D3	**STPB2**	Ch.2 stop bit selection	1 2 bits / 0 1 bit	X	R/W	
		D2	**SSCK2**	Ch.2 input clock selection	1 #SCLK2 / 0 Internal clock	X	R/W	
		D1	**SMD21**	Ch.2 transfer mode selection	SMD2[1:0] Transfer mode	X	R/W	
		D0	**SMD20**		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	**DIVMD2**	Ch.2 async. clock division ratio	1 1/8 / 0 1/16	X	R/W	
		D3	**IRTL2**	Ch.2 IrDA I/F output logic inversion	1 Inverted / 0 Direct	X	R/W	Valid only in asynchronous mode.
		D2	**IRRL2**	Ch.2 IrDA I/F input logic inversion	1 Inverted / 0 Direct	X	R/W	
		D1	**IRMD21**	Ch.2 interface mode selection	IRMD2[1:0] I/F mode	X	R/W	
		D0	**IRMD20**		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	R	
		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	–	–	–	0 when being read.
		D5	**TEND3**	Ch.3 transmit-completion flag	1 Transmitting / 0 End	0	R	
		D4	**FER3**	Ch.3 framing 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
		D4	**PMD3**	Ch.3 parity mode selection	1 Odd / 0 Even	X	R/W	asynchronous mode.
		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
		D2	**IRRL3**	Ch.3 IrDA I/F input logic inversion	1 Inverted / 0 Direct	X	R/W	asynchronous mode.
		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			
A/D conversion result (low-order) register	**0040240 (B)**	D7	**ADD7**	A/D converted data	0x0 to 0x3FF	0	R	
		D6	**ADD6**	(low-order 8 bits)	(low-order 8 bits)	0		
		D5	**ADD5**	ADD0 = LSB		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	0x0 to 0x3FF	0	R	
		D0	**ADD8**	(high-order 2 bits) ADD9 = MSB	(high-order 2 bits)	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			

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
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: 111 AD7; 110 AD6; 101 AD5; 100 AD4; 011 AD3; 010 AD2; 001 AD1; 000 AD0	0	R/W	
		D4	**CE1**			0		
		D3	**CE0**			0		
		D2	**CS2**	A/D converter start channel selection	CS[2:0] Start channel: 111 AD7; 110 AD6; 101 AD5; 100 AD4; 011 AD3; 010 AD2; 001 AD1; 000 AD0	0	R/W	
		D1	**CS1**			0		
		D0	**CS0**			0		
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 ADCADV = "1".
		D5	**CMPINTEN**	Upper/lower limit int. enable	1 Enabled / 0 Disabled	0	R/W	
		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.
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: 11 9 clocks; 10 7 clocks; 01 5 clocks; 00 3 clocks	1	R/W	Use with 9 clocks.
		D0	**ST0**			1		
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) AD0BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	**AD0BUF8**			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**			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) AD1BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	**AD1BUF8**			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) AD2BUF9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R	Can be used when ADCADV = "1".
		D0	**AD2BUF8**			0		
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**			0		
		D5	**AD3BUF5**			0		
		D4	**AD3BUF4**			0		
		D3	**AD3BUF3**			0		
		D2	**AD3BUF2**			0		
		D1	**AD3BUF1**			0		
		D0	**AD3BUF0**			0		
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**			0		
		D5	**AD4BUF5**			0		
		D4	**AD4BUF4**			0		
		D3	**AD4BUF3**			0		
		D2	**AD4BUF2**			0		
		D1	**AD4BUF1**			0		
		D0	**AD4BUF0**			0		
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**			0		
		D5	**AD5BUF5**			0		
		D4	**AD5BUF4**			0		
		D3	**AD5BUF3**			0		
		D2	**AD5BUF2**			0		
		D1	**AD5BUF1**			0		
		D0	**AD5BUF0**			0		
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**			0		
		D5	**AD6BUF5**			0		
		D4	**AD6BUF4**			0		
		D3	**AD6BUF3**			0		
		D2	**AD6BUF2**			0		
		D1	**AD6BUF1**			0		
		D0	**AD6BUF0**			0		

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) 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**			0		
		D5	**AD7BUF5**			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	–	–	–	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**			0		
		D5	**ADUPR5**			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	–	–	–	0 when being read.
		D1	**ADUPR9**	A/D conversion upper limit value (high-order 2 bits) ADUPR9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R/W	Can be used when ADCADV = "1".
		D0	**ADUPR8**			0		
A/D lower limit value (low-order) register	**004025A (B)**	D7	**ADLWR7**	A/D conversion lower limit value (low-order 8 bits) ADLWR0 = LSB	0x0 to 0x3FF (low-order 8 bits)	0	R/W	Can be used when ADCADV = "1".
		D6	**ADLWR6**			0		
		D5	**ADLWR5**			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	–	–	–	0 when being read.
		D1	**ADLWR9**	A/D conversion lower limit value (high-order 2 bits) ADLWR9 = MSB	0x0 to 0x3FF (high-order 2 bits)	0	R/W	Can be used when ADCADV = "1".
		D0	**ADLWR8**			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	
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**			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	0 to 7	X	R/W	
		D5	**PHSD1L1**	interrupt level		X		
		D4	**PHSD1L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PHSD0L2**	High-speed DMA Ch.0	0 to 7	X	R/W	
		D1	**PHSD0L1**	interrupt level		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	0 to 7	X	R/W	
		D5	**PHSD3L1**	interrupt level		X		
		D4	**PHSD3L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PHSD2L2**	High-speed DMA Ch.2	0 to 7	X	R/W	
		D1	**PHSD2L1**	interrupt level		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		
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		
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	0 to 7	X	R/W	
		D5	**PSIO01**	interrupt level		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	0 to 7	X	R/W	
		D1	**PSIO11**	interrupt level		X		
		D0	**PSIO10**			X		

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**			X		
		D0	**PCTM0**			X		
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**			X		
		D4	**PP5L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PP4L2**	Port input 4 interrupt level	0 to 7	X	R/W	
		D1	**PP4L1**			X		
		D0	**PP4L0**			X		
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**			X		
		D4	**PP7L0**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PP6L2**	Port input 6 interrupt level	0 to 7	X	R/W	
		D1	**PP6L1**			X		
		D0	**PP6L0**			X		
Serial I/F Ch.2/3 interrupt priority register	**004026E (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PSIO32**	Serial interface Ch.3	0 to 7	X	R/W	
		D5	**PSIO31**	interrupt level		X		
		D4	**PSIO30**			X		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**PSIO22**	Serial interface Ch.2	0 to 7	X	R/W	
		D1	**PSIO21**	interrupt level		X		
		D0	**PSIO20**			X		
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	
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)**	D7–4	**–**	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	
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	

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	–				–	–	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.
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	**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	–				–	–	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	

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	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	
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	

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			
		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			
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			
		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	

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	0 to 7				X	R/W	
		D5	**–**	function (reserved)					X		
		D4	**–**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**–**	Interrupt level of extended	0 to 7				X	R/W	
		D1	**–**	function (reserved)					X		
		D0	**–**						X		
Interrupt priority register for functions extended	**00402A1 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**–**	Interrupt level of extended	0 to 7				X	R/W	
		D5	**–**	function (reserved)					X		
		D4	**–**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**–**	Interrupt level of extended	0 to 7				X	R/W	
		D1	**–**	function (reserved)					X		
		D0	**–**						X		
LCDC interrupt priority register	**00402A2 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**PLCDCI2**	LCD controller interrupt level	0 to 7				X	R/W	
		D5	**PLCDCI1**						X		
		D4	**PLCDCI0**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**–**	Interrupt level of extended	0 to 7				X	R/W	
		D1	**–**	function (reserved)					X		
		D0	**–**						X		
USB, SPI interrupt priority register	**00402A3 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**PSPII2**	SPI interrupt level	0 to 7				X	R/W	
		D5	**PSPII1**						X		
		D4	**PSPII0**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**PUSBI2**	USB interrupt level	0 to 7				X	R/W	
		D1	**PUSBI1**						X		
		D0	**PUSBI0**						X		
Interrupt priority register for functions extended	**00402A4 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**–**	Interrupt level of extended	0 to 7				X	R/W	
		D5	**–**	function (reserved)					X		
		D4	**–**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**–**	Interrupt level of extended	0 to 7				X	R/W	
		D1	**–**	function (reserved)					X		
		D0	**–**						X		
Interrupt priority register for functions extended	**00402A5 (B)**	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**–**	Interrupt level of extended	0 to 7				X	R/W	
		D5	**–**	function (reserved)					X		
		D4	**–**						X		
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**–**	Interrupt level of extended	0 to 7				X	R/W	
		D1	**–**	function (reserved)					X		
		D0	**–**						X		

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCDC, USB, SPI interrupt enable register	**00402A6 (B)**	D7	**ESPII**	SPI interrupts	1	Enabled	0	Disabled	0	R/W	
		D6	**EUSBI**	USB interrupts					0	R/W	
		D5	**ELCDCI**	LCDC interrupts					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	Do not write 1.
		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	
LCDC, USB, SPI interrupt factor flag register	**00402A9 (B)**	D7	**FSPII**	SPI interrupts	1	Factor is generated	0	No factor is generated	X	R/W	
		D6	**FUSBI**	USB interrupts					X	R/W	
		D5	**FLCDCI**	LCDC interrupts					X	R/W	
		D4	–	Extended interrupt (reserved)					X	R/W	Do not write 1.
		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	
LCDC, USB, SPI IDMA request register	**00402AC (B)**	D7	**RSPII**	SPI interrupts	1	IDMA request	0	Interrupt request	0	R/W	
		D6	**RUSBI**	USB interrupts					0	R/W	
		D5	**RLCDCI**	LCDC interrupts					0	R/W	
		D4	–	Extended interrupt (reserved)					0	R/W	Do not write 1.
		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	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCDC, USB, SPI IDMA enable register	**00402AE (B)**	D7	**DESPII**	SPI interrupts	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D6	**DEUSBI**	USB interrupts					0	R/W	
		D5	**DELCDCI**	LCDC interrupts					0	R/W	
		D4	**–**	Extended interrupt (reserved)					0	R/W	Do not write 1.
		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	0	K67	0	R/W	
		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**	FPK1 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	–				–	–	0 when being read.
		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	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	

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 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	–				–	–	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 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	1	Output	0	Input	0	R/W	This register
		D6	**IOC26**	P26 I/O control					0	R/W	indicates the values
		D5	**IOC25**	P25 I/O control					0	R/W	of the I/O control
		D4	**IOC24**	P24 I/O control					0	R/W	signals of the ports
		D3	**IOC23**	P23 I/O control					0	R/W	when it is read. (See
		D2	**IOC22**	P22 I/O control					0	R/W	detailed explanation.)
		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	–				–	–	0 when being read.
		D5	**IOC35**	P35 I/O control	1	Output	0	Input	0	R/W	This register
		D4	**IOC34**	P34 I/O control					0	R/W	indicates the values
		D3	**IOC33**	P33 I/O control					0	R/W	of the I/O control
		D2	**IOC32**	P32 I/O control					0	R/W	signals of the ports
		D1	**IOC31**	P31 I/O control					0	R/W	when it is read. (See
		D0	**IOC30**	P30 I/O control					0	R/W	detailed explanation.)
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	

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]: 1 1 → 3.5; 1 0 → 2.5; 0 1 → 1.5; 0 0 → 0.5	1	R/W	
		DC	**A18DF0**			1		
		DB	**–**	reserved	–	–	–	0 when being read.
		DA	**A18WT2**	Areas 18–17 wait control	A18WT[2:0]: 1 1 1 → 7; 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	1	R/W	
		D9	**A18WT1**			1		
		D8	**A18WT0**			1		
		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]: 1 1 → 3.5; 1 0 → 2.5; 0 1 → 1.5; 0 0 → 0.5	1	R/W	
		D4	**A16DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A16WT2**	Areas 16–15 wait control	A16WT[2:0]: 1 1 1 → 7; 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	1	R/W	
		D1	**A16WT1**			1		
		D0	**A16WT0**			1		
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]: 1 1 → 3.5; 1 0 → 2.5; 0 1 → 1.5; 0 0 → 0.5	1	R/W	
		D4	**A14DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A14WT2**	Areas 14–13 wait control	A14WT[2:0]: 1 1 1 → 7; 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	1	R/W	
		D1	**A14WT1**			1		
		D0	**A14WT0**			1		
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]: 1 1 → 3.5; 1 0 → 2.5; 0 1 → 1.5; 0 0 → 0.5	1	R/W	
		D4	**A12DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A12WT2**	Areas 12–11 wait control	A12WT[2:0]: 1 1 1 → 7; 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	1	R/W	
		D1	**A12WT1**			1		
		D0	**A12WT0**			1		

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 1 1 2MB 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	1	R/W	
		DD	**A10IR1**			1		
		DC	**A10IR0**			1		
		DB	**–**	reserved	–	–	–	0 when being read.
		DA	**A10BW1**	Areas 10–9 burst ROM burst read cycle wait control	A10BW[1:0] Wait cycles 1 1 3 1 0 2 0 1 1 0 0 0	0	R/W	
		D9	**A10BW0**			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 1 3.5 1 0 2.5 0 1 1.5 0 0 0.5	1	R/W	
		D4	**A10DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A10WT2**	Areas 10–9 wait control	A10WT[2:0] Wait cycles 1 1 1 7 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	1	R/W	
		D1	**A10WT1**			1		
		D0	**A10WT0**			1		
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 1 3.5 1 0 2.5 0 1 1.5 0 0 0.5	1	R/W	
		D4	**A8DF0**			1		
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**A8WT2**	Areas 8–7 wait control	A8WT[2:0] Wait cycles 1 1 1 7 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	1	R/W	
		D1	**A8WT1**			1		
		D0	**A8WT0**			1		

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	1		
		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 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	1		
		D8	**A6WT0**			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 0 0 0.5	1		
		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 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	1		
		D0	**A5WT0**			1		
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	0		
		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	RRA[1:0] Number of cycles	0	R/W	
		D5	**RRA0**	selection	1 1 5 1 0 4 0 1 3 0 0 2	0		
		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]	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	RPRC[1:0]	0	R/W	
		D6	**RPRC0**	RAS precharge cycles selection	1 1: 4 cycles	0		
					1 0: 3 cycles			
					0 1: 2 cycles			
					0 0: 1 cycle			
		D5	**–**	reserved	–	–	–	0 when being read.
		D4	**CASC1**	DRAM	CASC[1:0]	0	R/W	
		D3	**CASC0**	CAS cycles selection	1 1: 4 cycles	0		
					1 0: 3 cycles			
					0 1: 2 cycles			
					0 0: 1 cycle			
		D2	**–**	reserved	–	–	–	0 when being read.
		D1	**RASC1**	DRAM	RASC[1:0]	0	R/W	
		D0	**RASC0**	RAS cycles selection	1 1: 4 cycles	0		
					1 0: 3 cycles			
					0 1: 2 cycles			
					0 0: 1 cycle			
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	
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.
		D8	**TTBR08**			0		Writing 1 not allowed.
		D7	**TTBR07**			0		
		D6	**TTBR06**			0		
		D5	**TTBR05**			0		
		D4	**TTBR04**			0		
		D3	**TTBR03**			0		
		D2	**TTBR02**			0		
		D1	**TTBR01**			0		
		D0	**TTBR00**			0		

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.
		DE	**TTBR32**			0		Writing 1 not allowed.
		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		
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: 1 1 PLL_CLK; 1 0 OSC3_CLK; 0 1 BUS_CLK; 0 0 CPU_CLK	0	R/W	
		D0	**BCLKSEL0**			0		
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		0	R/W	
		D5	**A14RH**	Areas 14–13 read hold cycle		0	R/W	
		D4	**A12RH**	Areas 12–11 read hold cycle		0	R/W	
		D3	**A10RH**	Areas 10–9 read hold cycle		0	R/W	
		D2	**A8RH**	Areas 8–7 read hold cycle		0	R/W	
		D1	**A6RH**	Area 6 read hold cycle		0	R/W	
		D0	**A5RH**	Areas 5–4 read hold cycle		0	R/W	
Bus speed setting register	**004813E (B)**	D7	**A18BS**	Areas 18–17 bus speed selection	1 ×1 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		0	R/W	
		D5	**A14BS**	Areas 14–13 bus speed selection		0	R/W	
		D4	**A12BS**	Areas 12–11 bus speed selection		0	R/W	
		D3	**A10BS**	Areas 10– 9 bus speed selection		0	R/W	
		D2	**A8BS**	Areas 8–7 bus speed selection		0	R/W	
		D1	**A6BS**	Area 6 bus speed selection		0	R/W	
		D0	**A5BS**	Areas 5–4 bus speed selection		0	R/W	

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	0 to 65535	X	R/W	
		DE	**CR0A14**	CR0A15 = MSB		X		
		DD	**CR0A13**	CR0A0 = LSB		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	0 to 65535	X	R/W	
		DE	**CR0B14**	CR0B15 = MSB		X		
		DD	**CR0B13**	CR0B0 = LSB		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	0 to 65535	X	R	
		DE	**TC014**	TC015 = MSB		X		
		DD	**TC013**	TC00 = LSB		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	**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	

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	0 to 65535	X	R/W	
		DE	**CR1A14**	CR1A15 = MSB		X		
		DD	**CR1A13**	CR1A0 = LSB		X		
		DC	**CR1A12**			X		
		DB	**CR1A11**			X		
		DA	**CR1A10**			X		
		D9	**CR1A9**			X		
		D8	**CR1A8**			X		
		D7	**CR1A7**			X		
		D6	**CR1A6**			X		
		D5	**CR1A5**			X		
		D4	**CR1A4**			X		
		D3	**CR1A3**			X		
		D2	**CR1A2**			X		
		D1	**CR1A1**			X		
		D0	**CR1A0**			X		
16-bit timer 1 comparison data B set-up register	**004818A (HW)**	DF	**CR1B15**	16-bit timer 1 comparison data B	0 to 65535	X	R/W	
		DE	**CR1B14**	CR1B15 = MSB		X		
		DD	**CR1B13**	CR1B0 = LSB		X		
		DC	**CR1B12**			X		
		DB	**CR1B11**			X		
		DA	**CR1B10**			X		
		D9	**CR1B9**			X		
		D8	**CR1B8**			X		
		D7	**CR1B7**			X		
		D6	**CR1B6**			X		
		D5	**CR1B5**			X		
		D4	**CR1B4**			X		
		D3	**CR1B3**			X		
		D2	**CR1B2**			X		
		D1	**CR1B1**			X		
		D0	**CR1B0**			X		
16-bit timer 1 counter data register	**004818C (HW)**	DF	**TC115**	16-bit timer 1 counter data	0 to 65535	X	R	
		DE	**TC114**	TC115 = MSB		X		
		DD	**TC113**	TC10 = LSB		X		
		DC	**TC112**			X		
		DB	**TC111**			X		
		DA	**TC110**			X		
		D9	**TC19**			X		
		D8	**TC18**			X		
		D7	**TC17**			X		
		D6	**TC16**			X		
		D5	**TC15**			X		
		D4	**TC14**			X		
		D3	**TC13**			X		
		D2	**TC12**			X		
		D1	**TC11**			X		
		D0	**TC10**			X		
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	

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	0 to 65535	X	R/W	
		DE	**CR2A14**	CR2A15 = MSB		X		
		DD	**CR2A13**	CR2A0 = LSB		X		
		DC	**CR2A12**			X		
		DB	**CR2A11**			X		
		DA	**CR2A10**			X		
		D9	**CR2A9**			X		
		D8	**CR2A8**			X		
		D7	**CR2A7**			X		
		D6	**CR2A6**			X		
		D5	**CR2A5**			X		
		D4	**CR2A4**			X		
		D3	**CR2A3**			X		
		D2	**CR2A2**			X		
		D1	**CR2A1**			X		
		D0	**CR2A0**			X		
16-bit timer 2 comparison data B set-up register	**0048192 (HW)**	DF	**CR2B15**	16-bit timer 2 comparison data B	0 to 65535	X	R/W	
		DE	**CR2B14**	CR2B15 = MSB		X		
		DD	**CR2B13**	CR2B0 = LSB		X		
		DC	**CR2B12**			X		
		DB	**CR2B11**			X		
		DA	**CR2B10**			X		
		D9	**CR2B9**			X		
		D8	**CR2B8**			X		
		D7	**CR2B7**			X		
		D6	**CR2B6**			X		
		D5	**CR2B5**			X		
		D4	**CR2B4**			X		
		D3	**CR2B3**			X		
		D2	**CR2B2**			X		
		D1	**CR2B1**			X		
		D0	**CR2B0**			X		
16-bit timer 2 counter data register	**0048194 (HW)**	DF	**TC215**	16-bit timer 2 counter data	0 to 65535	X	R	
		DE	**TC214**	TC215 = MSB		X		
		DD	**TC213**	TC20 = LSB		X		
		DC	**TC212**			X		
		DB	**TC211**			X		
		DA	**TC210**			X		
		D9	**TC29**			X		
		D8	**TC28**			X		
		D7	**TC27**			X		
		D6	**TC26**			X		
		D5	**TC25**			X		
		D4	**TC24**			X		
		D3	**TC23**			X		
		D2	**TC22**			X		
		D1	**TC21**			X		
		D0	**TC20**			X		
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	

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	0 to 65535	X	R/W	
		DE	**CR3A14**	CR3A15 = MSB		X		
		DD	**CR3A13**	CR3A0 = LSB		X		
		DC	**CR3A12**			X		
		DB	**CR3A11**			X		
		DA	**CR3A10**			X		
		D9	**CR3A9**			X		
		D8	**CR3A8**			X		
		D7	**CR3A7**			X		
		D6	**CR3A6**			X		
		D5	**CR3A5**			X		
		D4	**CR3A4**			X		
		D3	**CR3A3**			X		
		D2	**CR3A2**			X		
		D1	**CR3A1**			X		
		D0	**CR3A0**			X		
16-bit timer 3 comparison data B set-up register	**004819A (HW)**	DF	**CR3B15**	16-bit timer 3 comparison data B	0 to 65535	X	R/W	
		DE	**CR3B14**	CR3B15 = MSB		X		
		DD	**CR3B13**	CR3B0 = LSB		X		
		DC	**CR3B12**			X		
		DB	**CR3B11**			X		
		DA	**CR3B10**			X		
		D9	**CR3B9**			X		
		D8	**CR3B8**			X		
		D7	**CR3B7**			X		
		D6	**CR3B6**			X		
		D5	**CR3B5**			X		
		D4	**CR3B4**			X		
		D3	**CR3B3**			X		
		D2	**CR3B2**			X		
		D1	**CR3B1**			X		
		D0	**CR3B0**			X		
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		X		
		DD	**TC313**	TC30 = LSB		X		
		DC	**TC312**			X		
		DB	**TC311**			X		
		DA	**TC310**			X		
		D9	**TC39**			X		
		D8	**TC38**			X		
		D7	**TC37**			X		
		D6	**TC36**			X		
		D5	**TC35**			X		
		D4	**TC34**			X		
		D3	**TC33**			X		
		D2	**TC32**			X		
		D1	**TC31**			X		
		D0	**TC30**			X		
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	

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		X		
		DD	**CR4A13**	CR4A0 = LSB		X		
		DC	**CR4A12**			X		
		DB	**CR4A11**			X		
		DA	**CR4A10**			X		
		D9	**CR4A9**			X		
		D8	**CR4A8**			X		
		D7	**CR4A7**			X		
		D6	**CR4A6**			X		
		D5	**CR4A5**			X		
		D4	**CR4A4**			X		
		D3	**CR4A3**			X		
		D2	**CR4A2**			X		
		D1	**CR4A1**			X		
		D0	**CR4A0**			X		
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		X		
		DD	**CR4B13**	CR4B0 = LSB		X		
		DC	**CR4B12**			X		
		DB	**CR4B11**			X		
		DA	**CR4B10**			X		
		D9	**CR4B9**			X		
		D8	**CR4B8**			X		
		D7	**CR4B7**			X		
		D6	**CR4B6**			X		
		D5	**CR4B5**			X		
		D4	**CR4B4**			X		
		D3	**CR4B3**			X		
		D2	**CR4B2**			X		
		D1	**CR4B1**			X		
		D0	**CR4B0**			X		
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		X		
		DD	**TC413**	TC40 = LSB		X		
		DC	**TC412**			X		
		DB	**TC411**			X		
		DA	**TC410**			X		
		D9	**TC49**			X		
		D8	**TC48**			X		
		D7	**TC47**			X		
		D6	**TC46**			X		
		D5	**TC45**			X		
		D4	**TC44**			X		
		D3	**TC43**			X		
		D2	**TC42**			X		
		D1	**TC41**			X		
		D0	**TC40**			X		
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	

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		X		
		DD	**CR5A13**	CR5A0 = LSB		X		
		DC	**CR5A12**			X		
		DB	**CR5A11**			X		
		DA	**CR5A10**			X		
		D9	**CR5A9**			X		
		D8	**CR5A8**			X		
		D7	**CR5A7**			X		
		D6	**CR5A6**			X		
		D5	**CR5A5**			X		
		D4	**CR5A4**			X		
		D3	**CR5A3**			X		
		D2	**CR5A2**			X		
		D1	**CR5A1**			X		
		D0	**CR5A0**			X		
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		X		
		DD	**CR5B13**	CR5B0 = LSB		X		
		DC	**CR5B12**			X		
		DB	**CR5B11**			X		
		DA	**CR5B10**			X		
		D9	**CR5B9**			X		
		D8	**CR5B8**			X		
		D7	**CR5B7**			X		
		D6	**CR5B6**			X		
		D5	**CR5B5**			X		
		D4	**CR5B4**			X		
		D3	**CR5B3**			X		
		D2	**CR5B2**			X		
		D1	**CR5B1**			X		
		D0	**CR5B0**			X		
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		X		
		DD	**TC513**	TC50 = LSB		X		
		DC	**TC512**			X		
		DB	**TC511**			X		
		DA	**TC510**			X		
		D9	**TC59**			X		
		D8	**TC58**			X		
		D7	**TC57**			X		
		D6	**TC56**			X		
		D5	**TC55**			X		
		D4	**TC54**			X		
		D3	**TC53**			X		
		D2	**TC52**			X		
		D1	**TC51**			X		
		D0	**TC50**			X		
16-bit timer 5 control register	**00481AE (B)**	D7	**–**	reserved	–	0	–	0 when being read.
		D6	**SELFM5**	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		

Note:
D) Dual address mode
S) Single address mode

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
High-speed DMA Ch.0 low-order source address set-up register Note: D) Dual address mode S) Single address mode	**0048224 (HW)**	DF	**S0ADRL15**	D) Ch.0 source address[15:0] S) Ch.0 memory address[15:0]		X	R/W	
		DE	**S0ADRL14**			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 Note: D) Dual address mode S) Single address mode	**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 S) Ch.0 memory address control	S0IN[1:0] Inc/dec 1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0	R/W	
		DC	**S0IN0**			0		
		DB	**S0ADRH11**	D) Ch.0 source address[27:16] S) Ch.0 memory address[27:16]		X	R/W	
		DA	**S0ADRH10**			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 Note: D) Dual address mode S) Single address mode	**0048228 (HW)**	DF	**D0ADRL15**	D) Ch.0 destination address[15:0] S) Invalid		X	R/W	
		DE	**D0ADRL14**			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		
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 1 1 Invalid 1 0 Block 0 1 Successive 0 0 Single	0	R/W	
		DE	**D0MOD0**			0		
		DD	**D0IN1**	D) Ch.0 destination address control S) Invalid	D0IN[1:0] Inc/dec 1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0	R/W	
		DC	**D0IN0**			0		
		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		

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)	1 Clear 0 No operation	0	R/W	
				Ch.0 trigger flag status (reading)	1 Set 0 Cleared			
High-speed DMA Ch.1 transfer counter register	**0048230 (HW)**	DF	**TC1_L7**	Ch.1 transfer counter[7:0]		X	R/W	
		DE	**TC1_L6**	(block transfer mode)		X		
		DD	**TC1_L5**			X		
		DC	**TC1_L4**	Ch.1 transfer counter[15:8]		X		
		DB	**TC1_L3**	(single/successive transfer mode)		X		
		DA	**TC1_L2**			X		
		D9	**TC1_L1**			X		
		D8	**TC1_L0**			X		
		D7	**BLKLEN17**	Ch.1 block length		X	R/W	
		D6	**BLKLEN16**	(block transfer mode)		X		
		D5	**BLKLEN15**			X		
		D4	**BLKLEN14**	Ch.1 transfer counter[7:0]		X		
		D3	**BLKLEN13**	(single/successive transfer mode)		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.
Note: D) Dual address mode S) Single address mode		D7	**TC1_H7**	Ch.1 transfer counter[15:8]		X	R/W	
		D6	**TC1_H6**	(block transfer mode)		X		
		D5	**TC1_H5**			X		
		D4	**TC1_H4**	Ch.1 transfer counter[23:16]		X		
		D3	**TC1_H3**	(single/successive transfer mode)		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]		X	R/W	
		DE	**S1ADRL14**	S) Ch.1 memory address[15:0]		X		
		DD	**S1ADRL13**			X		
		DC	**S1ADRL12**			X		
		DB	**S1ADRL11**			X		
		DA	**S1ADRL10**			X		
Note: D) Dual address mode S) Single address mode		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)	0		
					1 0 Inc.(init)			
					0 1 Dec.(no init)			
Note: D) Dual address mode S) Single address mode					0 0 Fixed			
		DB	**S1ADRH11**	D) Ch.1 source address[27:16]		X	R/W	
		DA	**S1ADRH10**	S) Ch.1 memory address[27:16]		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		

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]		X	R/W	
		DE	**D1ADRL14**	S) Invalid		X		
		DD	**D1ADRL13**			X		
		DC	**D1ADRL12**			X		
		DB	**D1ADRL11**			X		
		DA	**D1ADRL10**			X		
		D9	**D1ADRL9**			X		
		D8	**D1ADRL8**			X		
		D7	**D1ADRL7**			X		
		D6	**D1ADRL6**			X		
		D5	**D1ADRL5**			X		
		D4	**D1ADRL4**			X		
		D3	**D1ADRL3**			X		
		D2	**D1ADRL2**			X		
		D1	**D1ADRL1**			X		
		D0	**D1ADRL0**			X		
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]: 1 1 Invalid; 1 0 Block; 0 1 Successive; 0 0 Single	0	R/W	
		DE	**D1MOD0**			0		
		DD	**D1IN1**	D) Ch.1 destination address control S) Invalid	D1IN[1:0]: 1 1 Inc/dec; 1 0 Inc.(no init); 0 1 Inc.(init); 0 0 Dec.(no init); Fixed	0	R/W	
		DC	**D1IN0**			0		
		DB	**D1ADRH11**	D) Ch.1 destination address[27:16] S) Invalid		X	R/W	
		DA	**D1ADRH10**			X		
		D9	**D1ADRH9**			X		
		D8	**D1ADRH8**			X		
		D7	**D1ADRH7**			X		
		D6	**D1ADRH6**			X		
		D5	**D1ADRH5**			X		
		D4	**D1ADRH4**			X		
		D3	**D1ADRH3**			X		
		D2	**D1ADRH2**			X		
		D1	**D1ADRH1**			X		
		D0	**D1ADRH0**			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	**TC2_L7**	Ch.2 transfer counter[7:0] (block transfer mode)		X	R/W	
		DE	**TC2_L6**			X		
		DD	**TC2_L5**			X		
		DC	**TC2_L4**	Ch.2 transfer counter[15:8] (single/successive transfer mode)		X		
		DB	**TC2_L3**			X		
		DA	**TC2_L2**			X		
		D9	**TC2_L1**			X		
		D8	**TC2_L0**			X		
		D7	**BLKLEN27**	Ch.2 block length (block transfer mode) Ch.2 transfer counter[7:0] (single/successive transfer mode)		X	R/W	
		D6	**BLKLEN26**			X		
		D5	**BLKLEN25**			X		
		D4	**BLKLEN24**			X		
		D3	**BLKLEN23**			X		
		D2	**BLKLEN22**			X		
		D1	**BLKLEN21**			X		
		D0	**BLKLEN20**			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	1 Memory WR; 0 Memory RD	0	R/W	
		DD–8	**–**	reserved	–	–	–	Undefined in read.
Note: D) Dual address mode S) Single address mode		D7	**TC2_H7**	Ch.2 transfer counter[15:8] (block transfer mode) Ch.2 transfer counter[23:16] (single/successive transfer mode)		X	R/W	
		D6	**TC2_H6**			X		
		D5	**TC2_H5**			X		
		D4	**TC2_H4**			X		
		D3	**TC2_H3**			X		
		D2	**TC2_H2**			X		
		D1	**TC2_H1**			X		
		D0	**TC2_H0**			X		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
High-speed DMA Ch.2 low-order source address set-up register Note: D) Dual address mode S) Single address mode	**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 Note: D) Dual address mode S) Single address mode	**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]	0	R/W	
		DC	**S2IN0**	S) Ch.2 memory address control	1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0		
		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 Note: D) Dual address mode S) Single address mode	**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 Note: D) Dual address mode S) Single address mode	**004824A (HW)**	DF	**D2MOD1**	Ch.2 transfer mode	D2MOD[1:0] Mode 1 1 Invalid 1 0 Block 0 1 Successive 0 0 Single	0	R/W	
		DE	**D2MOD0**			0		
		DD	**D2IN1**	D) Ch.2 destination address control S) Invalid	D2IN[1:0] Inc/dec 1 1 Inc.(no init) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0	R/W	
		DC	**D2IN0**			0		
		DB	**D2ADRH11**	D) Ch.2 destination address[27:16] S) Invalid		X	R/W	
		DA	**D2ADRH10**			X		
		D9	**D2ADRH9**			X		
		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		

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)	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	**TC3_L7**	Ch.3 transfer counter[7:0] (block transfer mode) (single/successive transfer mode)		X	R/W	
		DE	**TC3_L6**			X		
		DD	**TC3_L5**			X		
		DC	**TC3_L4**	Ch.3 transfer counter[15:8]		X		
		DB	**TC3_L3**			X		
		DA	**TC3_L2**			X		
		D9	**TC3_L1**			X		
		D8	**TC3_L0**			X		
		D7	**BLKLEN37**	Ch.3 block length (block transfer mode) Ch.3 transfer counter[7:0] (single/successive transfer mode)		X	R/W	
		D6	**BLKLEN36**			X		
		D5	**BLKLEN35**			X		
		D4	**BLKLEN34**			X		
		D3	**BLKLEN33**			X		
		D2	**BLKLEN32**			X		
		D1	**BLKLEN31**			X		
		D0	**BLKLEN30**			X		
High-speed DMA Ch.3 control register Note: D) Dual address mode S) Single address mode	**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) Ch.3 transfer counter[23:16] (single/successive transfer mode)		X	R/W	
		D6	**TC3_H6**			X		
		D5	**TC3_H5**			X		
		D4	**TC3_H4**			X		
		D3	**TC3_H3**			X		
		D2	**TC3_H2**			X		
		D1	**TC3_H1**			X		
		D0	**TC3_H0**			X		
High-speed DMA Ch.3 low-order source address set-up register Note: D) Dual address mode S) Single address mode	**0048254 (HW)**	DF	**S3ADRL15**	D) Ch.3 source address[15:0] S) Ch.3 memory address[15:0]		X	R/W	
		DE	**S3ADRL14**			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 Note: D) Dual address mode S) Single address mode	**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) 1 0 Inc.(init) 0 1 Dec.(no init) 0 0 Fixed	0		
		DB	**S3ADRH11**	D) Ch.3 source address[27:16] S) Ch.3 memory address[27:16]		X	R/W	
		DA	**S3ADRH10**			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		

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		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		
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]: 1 1 Invalid; 1 0 Block; 0 1 Successive; 0 0 Single	0	R/W	
		DE	**D3MOD0**			0		
		DD	**D3IN1**	D) Ch.3 destination address control S) Invalid	D3IN[1:0]: 1 1 Inc.(no init); 1 0 Inc.(init); 0 1 Dec.(no init); 0 0 Fixed	0	R/W	
		DC	**D3IN0**			0		
		DB	**D3ADRH11**	D) Ch.3 destination address[27:16] S) Invalid		X	R/W	
		DA	**D3ADRH10**			X		
		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			

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	0x15	0	R	
		D6	**MID6**	0x03: S1C333xx		0		
		D5	**MID5**	0x15: S1C33Lxx		0		
		D4	**MID4**			1		
		D3	**MID3**			0		
		D2	**MID2**			1		
		D1	**MID1**			0		
		D0	**MID0**			1		
Model ID register	**0300002 (B)**	D7	**NAME7**	Model ID	0x05	0	R	
		D6	**NAME6**	0x04: S1C33L04		0		
		D5	**NAME5**	0x05: S1C33L05		0		
		D4	**NAME4**	0x11: S1C33L11		0		
		D3	**NAME3**			0		
		D2	**NAME2**			1		
		D1	**NAME1**			0		
		D0	**NAME0**			1		
Version register	**0300003 (B)**	D7	**VER3**	Version code	0x00	0	R	
		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	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	**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	–	–	–	0 when being read.
		D5	**P55D**	P55 I/O port data	1 High 0 Low	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	–	–	–	0 when being read.
		D5	**IOC55**	P55 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	**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	–	–	–	0 when being read.
		D3	**P63D**	P63 I/O port data	1 High 0 Low	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	–	–	–	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	

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
P00–P03 port function extension register	**0300040 (B)**	D7	**EFP031**	P03 port extended function	EFP03[1:0]		Function	0	R/W	
		D6	**EFP030**		1	*	reserved	0		
					0	1	#FSRDY0			
					0	0	P03/#SRDY0			
		D5	**EFP021**	P02 port extended function	EFP02[1:0]		Function	0	R/W	
		D4	**EFP020**		1	*	reserved	0		
					0	1	#FSCLK0			
					0	0	P02/#SCLK0			
		D3	**EFP011**	P01 port extended function	EFP01[1:0]		Function	0	R/W	
		D2	**EFP010**		1	*	reserved	0		
					0	1	FSOUT0			
					0	0	P01/SOUT0			
		D1	**EFP001**	P00 port extended function	EFP00[1:0]		Function	0	R/W	
		D0	**EFP000**		1	*	reserved	0		
					0	1	FSIN0			
					0	0	P00/SIN0			
P20–P21 port function extension register	**0300044 (B)**	D7–4	**–**	reserved	–			–	–	0 when being read.
		D3	**EFP211**	P21 port extended function	EFP21[1:0]		Function	0	R/W	
		D2	**EFP210**		1	*	reserved	0		
					0	1	#SDWE			
					0	0	P21/#DWE			
		D1	**EFP201**	P20 port extended function	EFP20[1:0]		Function	0	R/W	
		D0	**EFP200**		1	*	reserved	0		
					0	1	SDCKE			
					0	0	P20/#DRD			
P31–P33 port function extension register	**0300046 (B)**	D7	**EFP331**	P33 port extended function	EFP33[1:0]		Function	0	R/W	
		D6	**EFP330**		1	*	reserved	0		
					0	1	SDI			
					0	0	P33, etc.			
		D5	**EFP321**	P32 port extended function	EFP32[1:0]		Function	0	R/W	
		D4	**EFP320**		1	*	reserved	0		
					0	1	SPICLK			
					0	0	P32, etc.			
		D3	**EFP311**	P31 port extended function	EFP31[1:0]		Function	0	R/W	
		D2	**EFP310**		1	*	reserved	0		
					0	1	SDO			
					0	0	P31, etc.			
		D1–0	**–**	reserved	–			–	–	0 when being read.
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.			
P40–P43 port function extension register	**0300048 (B)**	D7	**EFP431**	P43 port extended function	EFP43[1:0]		Function	0	R/W	
		D6	**EFP430**		1	*	reserved	0		
					0	1	P43			
					0	0	A22			
		D5	**EFP421**	P42 port extended function	EFP42[1:0]		Function	0	R/W	
		D4	**EFP420**		1	*	reserved	0		
					0	1	P42			
					0	0	A23			
		D3	**EFP411**	P41 port extended function	EFP41[1:0]		Function	0	R/W	
		D2	**EFP410**		1	*	reserved	0		
					0	1	P41			
					0	0	A24			
		D1	**EFP401**	P40 port extended function	EFP40[1:0]		Function	0	R/W	
		D0	**EFP400**		1	*	reserved	0		
					0	1	P40			
					0	0	A25			

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
P44–P47 port function extension register	**0300049 (B)**	D7	**EFP471**	P47 port extended function	EFP47[1:0]		Function	0	R/W	
		D6	**EFP470**		1	*	reserved	0		
					0	1	P47			
					0	0	A18			
		D5	**EFP461**	P46 port extended function	EFP46[1:0]		Function	0	R/W	
		D4	**EFP460**		1	*	reserved	0		
					0	1	P46			
					0	0	A19			
		D3	**EFP451**	P45 port extended function	EFP45[1:0]		Function	0	R/W	
		D2	**EFP450**		1	*	reserved	0		
					0	1	P45			
					0	0	A20			
		D1	**EFP441**	P44 port extended function	EFP44[1:0]		Function	0	R/W	
		D0	**EFP440**		1	*	reserved	0		
					0	1	P44			
					0	0	A21			
P50–P53 port function extension register	**030004A (B)**	D7	**EFP531**	P53 port extended function	EFP53[1:0]		Function	0	R/W	
		D6	**EFP530**		1	1	reserved	0		
					1	0	#SDCE			
					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–P63 port function extension register	**030004C (B)**	D7	**EFP631**	P63 port extended function	EFP63[1:0]		Function	0	R/W	
		D6	**EFP630**		1	*	reserved	0		
					0	1	UDQM			
					0	0	P63			
		D5	**EFP621**	P62 port extended function	EFP62[1:0]		Function	0	R/W	
		D4	**EFP620**		1	*	reserved	0		
					0	1	LDQM			
					0	0	P62			
		D3	**EFP611**	P61 port extended function	EFP61[1:0]		Function	0	R/W	
		D2	**EFP610**		1	*	reserved	0		
					0	1	SDA10			
					0	0	P61			
		D1	**EFP601**	P60 port extended function	EFP60[1:0]		Function	0	R/W	
		D0	**EFP600**		1	1	SDCLK	0		
					1	0	FOSC1			
					0	1	P60			
					0	0	BCLK			
NAND flash I/F #CE area select register	**0300100 (B)**	D7	**SMCODE**	Booting code enable	1	Enabled	0 Disabled	1	R/W	
		D6–2	**–**	reserved	–			–	–	0 when being read.
		D1	**SMCES1**	NAND flash I/F	SMCES[1:0]		#CE area	0	R/W	
		D0	**SMCES0**	#CE area selection	1	1	#CE9/#CE17(+18)	0		
				(according of CEFUNC)	1	0	#CE8/#CE14			
					0	1	#CE7/#CE13			
					0	0	#CE5/#CE15(+16)			

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
ECC reset/ready register	**0300101 (B)**	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**ECCRST**	ECC circuit reset	1 Reset 0 No effect	–	W	
			ECCRDY	Parity data ready status	1 Ready 0 Busy	0	R	
ECC enable register	**0300102 (B)**	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**ECCEN**	ECC circuit enable	1 Enabled 0 Disabled	0	R/W	
Mode register	**0300103 (B)**	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**MODE**	Flash device mode	1 16 bits 0 8 bits	0	R/W	
Area 0 ECC column parity data register	**0300104 (B)**	D7	**CP05**	Area 0 column parity data	0x0 to 0x3F	1	R	
		D6	**CP04**			1		
		D5	**CP03**			1		
		D4	**CP02**			1		
		D3	**CP01**			1		
		D2	**CP00**			1		
		D1	**–**	Unused bit	–	1	R	1 when being read.
		D0	**–**	Unused bit	–	1	R	
Area 0 ECC line parity register 0 (low-order byte)	**0300105 (B)**	D7	**LP07**	Area 0 ECC line parity low-order byte	0x0 to 0xFF	1	R	
		D6	**LP06**			1		
		D5	**LP05**			1		
		D4	**LP04**			1		
		D3	**LP03**			1		
		D2	**LP02**			1		
		D1	**LP01**			1		
		D0	**LP00**			1		
Area 0 ECC line parity register 1 (high-order byte)	**0300106 (B)**	D7	**LP015**	Area 0 ECC line parity high-order byte	0x0 to 0xFF	1	R	
		D6	**LP014**			1		
		D5	**LP013**			1		
		D4	**LP012**			1		
		D3	**LP011**			1		
		D2	**LP010**			1		
		D1	**LP09**			1		
		D0	**LP08**			1		
Area 1 ECC column parity data register	**0300107 (B)**	D7	**CP15**	Area 1 column parity data	0x0 to 0x3F	1	R	
		D6	**CP14**			1		
		D5	**CP13**			1		
		D4	**CP12**			1		
		D3	**CP11**			1		
		D2	**CP10**			1		
		D1	**–**	Unused bit	–	1	R	1 when being read.
		D0	**–**	Unused bit	–	1	R	
Area 1 ECC line parity register 0 (low-order byte)	**0300108 (B)**	D7	**LP17**	Area 1 ECC line parity low-order byte	0x0 to 0xFF	1	R	
		D6	**LP16**			1		
		D5	**LP15**			1		
		D4	**LP14**			1		
		D3	**LP13**			1		
		D2	**LP12**			1		
		D1	**LP11**			1		
		D0	**LP10**			1		
Area 1 ECC line parity register 1 (high-order byte)	**0300109 (B)**	D7	**LP115**	Area 1 ECC line parity high-order byte	0x0 to 0xFF	1	R	
		D6	**LP114**			1		
		D5	**LP113**			1		
		D4	**LP112**			1		
		D3	**LP111**			1		
		D2	**LP110**			1		
		D1	**LP19**			1		
		D0	**LP18**			1		
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		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
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		X		
		D5	**RXD05**	RXD00 = LSB		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]: 1 1 = 4; 1 0 = 3; 0 1 = 2; 0 0 = 1 or 0	0	R/W	
		D6	**RXD0NUM0**			0		
		D5	**TEND0**	Ch.0 transmit-completion flag	1 Transmitting / 0 End	0	R	
		D4	**FER0**	Ch.0 framing 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]: 1 1 = 8-bit asynchronous; 1 0 = 7-bit asynchronous; 0 1 = Clock sync. Slave; 0 0 = Clock sync. Master	X	R/W	
		D0	**SMD00**			X		
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]: 1 1 = 4; 1 0 = 3; 0 1 = 2; 0 0 = 1	0	R/W	
		D5	**FIFOINT00**			0	R/W	
		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]: 1 1 = reserved; 1 0 = IrDA 1.0; 0 1 = reserved; 0 0 = General I/F	X	R/W	
		D0	**IRMD00**			X		
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		
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 baud-rate timer reload data [9:8]	0x0 to 0x3 (BRTRD0[9:0] = 0x0 to 0x3FF)	0	R/W	
		D0	**BRTRD08**			0		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FIFO serial I/F Ch.0 baud-rate timer count data register (LSB)	**0300208 (B)**	D7	**BRTCD07**	FIFO serial I/F Ch.0 baud-rate timer count data [7:0]	0x0 to 0xFF (BRTCD0[9:0] = 0x0 to 0x3FF)	0	R	
		D6	**BRTCD06**			0		
		D5	**BRTCD05**			0		
		D4	**BRTCD04**			0		
		D3	**BRTCD03**			0		
		D2	**BRTCD02**			0		
		D1	**BRTCD01**			0		
		D0	**BRTCD00**			0		
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 baud-rate timer count data [9:8]	0x0 to 0x3 (BRTCD0[9:0] = 0x0 to 0x3FF)	0	R	
		D0	**BRTCD08**			0		
Bus signal pull-up control register	**0300F00 (B)**	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**PUPCAS**	#HCAS, #LCAS, BCLK (PA1, PA0, P60) pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D2	**PUPCE**	#CE10EX–#CE4 (P55–P50) pull-up		1	R/W	
		D1	**PUPAD**	A25–A0 (P40–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**	#HCAS, #LCAS, BCLK low drive		0	R/W	
		D2	**LDRVCE**	#CE10EX–#CE4 low drive		0	R/W	
		D1	**LDRVAD**	A25–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**	K64, BOOT pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D1	**PUPK6L**	K63–K60 pull-up		1	R/W	
		D0	**PUPK5**	K53–K50 pull-up		1	R/W	
I/O port pull-up control register	**0300F04 (B)**	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**PUPP6**	P63–P61 pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D5–4	**–**	reserved	–	–	–	0 when being read.
		D3	**PUPP3**	P35–P30 (PA2) 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	
PB, PC port pull-up control register	**0300F06 (B)**	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**PUPPBH**	PB7–PB4 pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D3	**PUPPBL**	PB3–PB0 pull-up		1	R/W	
		D2	**PUPPCL**	PC3–PC0 pull-up		1	R/W	
		D1–0	**–**	reserved	–	–	–	0 when being read.
PD port pull-up control register	**0300F08 (B)**	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PUPPDH**	PD7–PD4 pull-up	1 Pulled up 0 No pull-up	1	R/W	
		D0	**PUPPDL**	PD3–PD0 pull-up		1	R/W	
Macro control register	**0300F20 (B)**	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**USBSNZ**	USB snooze control	1 Enabled 0 Disabled	0	R/W	
		D4	**USBCLKEN**	USB clock enable	1 Enabled 0 Disabled	0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**USBWT2**	USB bus wait count	0 to 7	1	R/W	
		D1	**USBWT1**			1		
		D0	**USBWT0**			1		
Misc write protect register	**0300F2F (B)**	D7	**WRPROT7**	Misc register write protect	Writing 10010110 (0x96) removes the write protection of the Misc registers (0x300F20, 0x300F30–0x300F39). Writing another value to this register or writing to a Misc register (0x300F20, 0x300F30–0x300F39) will set the write protection again.	0	R/W	
		D6	**WRPROT6**			0		
		D5	**WRPROT5**			0		
		D4	**WRPROT4**			0		
		D3	**WRPROT3**			0		
		D2	**WRPROT2**			0		
		D1	**WRPROT1**			0		
		D0	**WRPROT0**			0		
System clock division register	**0300F30 (B)**	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**SYSCLK1**	PLL input clock setup (OSC3 clock division ratio)	SYSCLK[1:0]: 1 1 OSC3 / 3; 1 0 OSC3 / 2; 0 * OSC3 / 1	1	R/W	
		D0	**SYSCLK0**			0		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PLL control register	0300F31 (B)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	PLLPDWN	PLL software power down (when PLLS0 is set to "1")	1 PLL not used (CLK=OSC3) 0 PLL used	0	R/W	
		D0	PLLMODE	PLL multiple mode select	1 x2 0 x1	0	R/W	
Sub system clock select register	0300F32 (B)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SUBCKSEL	Sub system source clock select	1 PLL clock 0 OSC3 clock	0	R/W	
Bus clock generator control register	0300F33 (B)	D7	BCLGEN	Bus clock generator enable	1 Enabled 0 Disabled	0	R/W	
		D6–2	–	reserved	–	–	–	0 when being read.
		D1	BCLGDT1	Bus clock setup (Sub system clock division ratio)	BCLGDT[1:0] Division ratio 1 1 subsys_clk / 4 1 0 subsys_clk / 2 0 * subsys_clk / 1	0	R/W	
		D0	BCLGDT0			0		
LCDC clock generator control register	0300F34 (B)	D7	LCLGEN	LCDC clock generator enable	1 Enabled 0 Disabled	0	R/W	
		D6–4	–	reserved	–	–	–	0 when being read.
		D3	LCLGDT3	LCDC clock setup (Sub system clock division ratio)	LCDC clock frequency = $\frac{subsys\_clk}{LCLGDT[3:0] + 1}$ [Hz]	0	R/W	
		D2	LCLGDT2			0		
		D1	LCLGDT1			0		
		D0	LCLGDT0			0		
Module clock control register 0	0300F35 (B)	D7	USBCKOF	USB clock (sbcuclk)	1 Disabled 0 Enabled	0	R/W	
		D6	FSIFCK1OF	FSIF clock (bcuclk)		0	R/W	
		D5	FSIFCK2OF	FSIF clock (periclk)		0	R/W	
		D4	SPICKOF	SPI clock (sbcuclk)		0	R/W	
		D3	SMIFCKOF	SMIF clock (sbcuclk)		0	R/W	
		D2	SQCKOF	SQROM clock (sbcuclk)		0	R/W	
		D1	ARBCKOF	Bus arbiter clock (sbcuclk)		0	R/W	
		D0	LHIFCKOF	LCDC host I/F clock (sbcuclk)		0	R/W	
Module clock control register 1	0300F36 (B)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	SDRAMCKOF	SDRAMC clock (halt2run_clk)	1 Disabled 0 Enabled	0	R/W	
		D0	VRAMCKOF	Internal VRAM clock (halt2run_clk)		0	R/W	
Sub system clock control register	0300F37 (B)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SUBCKOF	OSC3/PLL clock disable	1 Disabled 0 Enabled	0	R/W	
A0/#BSL select register	0300F38 (B)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	BSLSEL	A0/#BSL mode select	1 #BSL mode 0 A0 mode	0	R/W	
x2 speed mode control register	0300F39 (B)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	SDRAMX2	SDRAM x2 speed mode disable	1 x1 speed mode 0 x2 speed mode	0	R/W	
		D0	IVRAMX2	IVRAM x2 speed mode disable		0	R/W	
PA I/O port data register	0300F40 (B)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	PA2D	PA2 I/O port data	1 High 0 Low	0	R/W	
		D1	PA1D	PA1 I/O port data		0	R/W	
		D0	PA0D	PA0 I/O port data		0	R/W	
PA I/O control register	0300F41 (B)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	IOCA2	PA2 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.
		D1	IOCA1	PA1 I/O control		0	R/W	
		D0	IOCA0	PA0 I/O control		0	R/W	
PB I/O port data register	0300F42 (B)	D7	PB7D	PB7 I/O port data	1 High 0 Low	0	R/W	
		D6	PB6D	PB6 I/O port data		0	R/W	
		D5	PB5D	PB5 I/O port data		0	R/W	
		D4	PB4D	PB4 I/O port data		0	R/W	
		D3	PB3D	PB3 I/O port data		0	R/W	
		D2	PB2D	PB2 I/O port data		0	R/W	
		D1	PB1D	PB1 I/O port data		0	R/W	
		D0	PB0D	PB0 I/O port data		0	R/W	
PB I/O control register	0300F43 (B)	D7	IOCB7	PB7 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.
		D6	IOCB6	PB6 I/O control		0	R/W	
		D5	IOCB5	PB5 I/O control		0	R/W	
		D4	IOCB4	PB4 I/O control		0	R/W	
		D3	IOCB3	PB3 I/O control		0	R/W	
		D2	IOCB2	PB2 I/O control		0	R/W	
		D1	IOCB1	PB1 I/O control		0	R/W	
		D0	IOCB0	PB0 I/O control		0	R/W	
PC I/O port data register	0300F44 (B)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	PC3D	PC3 I/O port data	1 High 0 Low	0	R/W	
		D2	PC2D	PC2 I/O port data		0	R/W	
		D1	PC1D	PC1 I/O port data		0	R/W	
		D0	PC0D	PC0 I/O port data		0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PC I/O control register	**0300F45 (B)**	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**IOCC3**	PC3 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.
		D2	**IOCC2**	PC2 I/O control		0	R/W	
		D1	**IOCC1**	PC1 I/O control		0	R/W	
		D0	**IOCC0**	PC0 I/O control		0	R/W	
PD I/O port data register	**0300F46 (B)**	D7	**PD7D**	PD7 I/O port data	1 High 0 Low	0	R/W	
		D6	**PD6D**	PD6 I/O port data		0	R/W	
		D5	**PD5D**	PD5 I/O port data		0	R/W	
		D4	**PD4D**	PD4 I/O port data		0	R/W	
		D3	**PD3D**	PD3 I/O port data		0	R/W	
		D2	**PD2D**	PD2 I/O port data		0	R/W	
		D1	**PD1D**	PD1 I/O port data		0	R/W	
		D0	**PD0D**	PD0 I/O port data		0	R/W	
PD I/O control register	**0300F47 (B)**	D7	**IOCD7**	PD7 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.
		D6	**IOCD6**	PD6 I/O control		0	R/W	
		D5	**IOCD5**	PD5 I/O control		0	R/W	
		D4	**IOCD4**	PD4 I/O control		0	R/W	
		D3	**IOCD3**	PD3 I/O control		0	R/W	
		D2	**IOCD2**	PD2 I/O control		0	R/W	
		D1	**IOCD1**	PD1 I/O control		0	R/W	
		D0	**IOCD0**	PD0 I/O control		0	R/W	
PA0–PA2 port function extension register	**0300F60 (B)**	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**FPA21**	PA2 port extended function	FPA2[1:0] Function	0	R/W	
		D4	**FPA20**		1 * reserved	0		
					0 1 PA2			
					0 0 P30, etc.			
		D3	**FPA11**	PA1 port extended function	FPA1[1:0] Function	0	R/W	
		D2	**FPA10**		1 1 reserved	0		
					1 0 #SDRAS			
					0 1 PA1			
					0 0 #HCAS			
		D1	**FPA01**	PA0 port extended function	FPA0[1:0] Function	0	R/W	
		D0	**FPA00**		1 1 reserved	0		
					1 0 #SDCAS			
					0 1 PA0			
					0 0 #LCAS			
PB0–PB3 port function extension register	**0300F62 (B)**	D7	**FPB31**	PB3 port extended function	FPB3[1:0] Function	0	R/W	
		D6	**FPB30**		1 * reserved	0		
					0 1 FPDAT3			
					0 0 PB3			
		D5	**FPB21**	PB2 port extended function	FPB2[1:0] Function	0	R/W	
		D4	**FPB20**		1 * reserved	0		
					0 1 FPDAT2			
					0 0 PB2			
		D3	**FPB11**	PB1 port extended function	FPB1[1:0] Function	0	R/W	
		D2	**FPB10**		1 * reserved	0		
					0 1 FPDAT1			
					0 0 PB1			
		D1	**FPB01**	PB0 port extended function	FPB0[1:0] Function	0	R/W	
		D0	**FPB00**		1 * reserved	0		
					0 1 FPDAT0			
					0 0 PB0			
PB4–PB7 port function extension register	**0300F63 (B)**	D7	**FPB71**	PB7 port extended function	FPB7[1:0] Function	0	R/W	
		D6	**FPB70**		1 * reserved	0		
					0 1 FPDAT7			
					0 0 PB7			
		D5	**FPB61**	PB6 port extended function	FPB6[1:0] Function	0	R/W	
		D4	**FPB60**		1 * reserved	0		
					0 1 FPDAT6			
					0 0 PB6			
		D3	**FPB51**	PB5 port extended function	FPB5[1:0] Function	0	R/W	
		D2	**FPB50**		1 * reserved	0		
					0 1 FPDAT5			
					0 0 PB5			
		D1	**FPB41**	PB4 port extended function	FPB4[1:0] Function	0	R/W	
		D0	**FPB40**		1 * reserved	0		
					0 1 FPDAT4			
					0 0 PB4			

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
PC0–PC3 port function extension register	**0300F64 (B)**	D7	**FPC31**	PC3 port extended function	FPC3[1:0]		Function	0	R/W	
		D6	**FPC30**		1	*	reserved	0		
					0	1	DRDY			
					0	0	PC3			
		D5	**FPC21**	PC2 port extended function	FPC2[1:0]		Function	0	R/W	
		D4	**FPC20**		1	*	reserved	0		
					0	1	FPSHIFT			
					0	0	PC2			
		D3	**FPC11**	PC1 port extended function	FPC1[1:0]		Function	0	R/W	
		D2	**FPC10**		1	*	reserved	0		
					0	1	FPLINE			
					0	0	PC1			
		D1	**FPC01**	PC0 port extended function	FPC0[1:0]		Function	0	R/W	
		D0	**FPC00**		1	*	reserved	0		
					0	1	FPFRAME			
					0	0	PC0			
PD0–PD2 port function extension register	**0300F66 (B)**	D7–6	**–**	reserved	–			–	–	0 when being read.
		D5	**FPD21**	PD2 port extended function	FPD2[1:0]		Function	0	R/W	
		D4	**FPD20**		1	*	reserved	0		
					0	1	SQUALE			
					0	0	PD2			
		D3	**FPD11**	PD1 port extended function	FPD1[1:0]		Function	0	R/W	
		D2	**FPD10**		1	*	reserved	0		
					0	1	SQLALE			
					0	0	PD1			
		D1	**FPD01**	PD0 port extended function	FPD0[1:0]		Function	0	R/W	
		D0	**FPD00**		1	*	reserved	0		
					0	1	#SQRD			
					0	0	PD0			

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Frame interrupt register	**0380010 (HW)**	DF	**FRINTEN**	Frame interrupt enable	1 Enabled 0 Disabled	0	R/W	
		DE–0	**–**	reserved	–	–	–	0 when being read.
Status and power save register	**0380014 (HW)**	DF	**FRINTF**	Frame interrupt flag	1 Generated 0 Not generated	0	R/W	Reset by writing 1.
		DE–A	**–**	reserved	–	–	–	0 when being read.
		D9	**FIFOEF**	Display FIFO empty flag	1 Empty 0 Not empty	0	R/W	Reset by writing 1.
		D8	**WRBUFF**	LUT/IO write buffer status	1 Full 0 Empty	0	R	
		D7	**VNDPF**	Vertical display status	1 VNDP 0 VDP	1	R	
		D6	**VRAMF**	VRAM status	1 Idle 0 Busy	1	R	
		D5	**TOERRF**	CPU access timeout error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D4	**–**	reserved	–	–	–	0 when being read.
		D3	**FIFOHP**	Display FIFO flexible-burst read	1 Flex-burst read 0 16 burst read	0	R/W	
		D2	**–**	reserved	–	–	–	0 when being read.
		D1	**LPSAVE1**	Power save mode	LPSAVE[1:0] Mode	0	R/W	
		D0	**LPSAVE0**		1 1 Normal operation	0		
					1 0 Doze mode			
					0 1 reserved			
					0 0 Power save mode			
Horizontal non-display period register	**0380040 (HW)**	DF–5	**–**	reserved	–	–	–	0 when being read.
		D4	**HNDP4**	Horizontal non-display period	Non-display period (pixels) / 8 - 4	0	R/W	
		D3	**HNDP3**	HNDP4 = MSB		0		
		D2	**HNDP2**	HNDP0 = LSB		0		
		D1	**HNDP1**			0		
		D0	**HNDP0**			0		
Horizontal panel size register	**0380042 (HW)**	DF–7	**–**	reserved	–	–	–	0 when being read.
		D6	**HSIZE6**	Horizontal panel size	H panel resolution (pixels) / 8 - 1	0	R/W	
		D5	**HSIZE5**	HSIZE6 = MSB		0		
		D4	**HSIZE4**	HSIZE0 = LSB		0		
		D3	**HSIZE3**			0		
		D2	**HSIZE2**			0		
		D1	**HSIZE1**			0		
		D0	**HSIZE0**			0		
Vertical non-display period register	**038004A (HW)**	DF–6	**–**	reserved	–	–	–	0 when being read.
		D5	**VNDP5**	Vertical non-display period	Non-display period (lines)	0	R/W	
		D4	**VNDP4**	VNDP5 = MSB		0		
		D3	**VNDP3**	VNDP0 = LSB		0		
		D2	**VNDP2**			0		
		D1	**VNDP1**			0		
		D0	**VNDP0**			0		
Vertical panel size register	**038004C (HW)**	DF–A	**–**	reserved	–	–	–	0 when being read.
		D9	**VSIZE9**	Vertical panel size	Vertical panel resolution (lines) -1	0	R/W	
		D8	**VSIZE8**	VSIZE9 = MSB		0		
		D7	**VSIZE7**	VSIZE0 = LSB		0		
		D6	**VSIZE6**			0		
		D5	**VSIZE5**			0		
		D4	**VSIZE4**			0		
		D3	**VSIZE3**			0		
		D2	**VSIZE2**			0		
		D1	**VSIZE1**			0		
		D0	**VSIZE0**			0		
MOD rate register	**0380052 (HW)**	DF–6	**–**	reserved	–	–	–	0 when being read.
		D5	**MOD5**	LCD MOD rate	0x0 to 0x3F	0	R/W	
		D4	**MOD4**	MOD5 = MSB		0		
		D3	**MOD3**	MOD0 = LSB		0		
		D2	**MOD2**			0		
		D1	**MOD1**			0		
		D0	**MOD0**			0		
LCDC mode register 0	**0380200 (HW)**	DF–8	**–**	reserved	–	–	–	0 when being read.
		D7	**FRMRPT**	Frame repeat for EL panel	1 Repeated 0 Not repeated	0	R/W	
		D6	**DITHEN**	Dither mode enable	1 Enabled 0 Disabled	0	R/W	
		D5	**–**	reserved	–	–	–	0 when being read.
		D4	**LUTPASS**	LUT bypass mode	1 Bypassed 0 Used	0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**BPP2**	Bit-per-pixel select	BPP[2:0] bpp (color/gray)	0	R/W	
		D1	**BPP1**		1 1 * reserved	0		
		D0	**BPP0**		1 0 1 16 bpp (64Kc/64gr)	0		
					1 0 0 12 bpp (4Kc/16gr)			
					0 1 1 8 bpp (256c/64gr)			
					0 1 0 4 bpp (16c/16gr)			
					0 0 1 2 bpp (4c/4gr)			
					0 0 0 1 bpp (2c/2gr)			

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC mode register 1	**0380202 (HW)**	DF	**–**	reserved	–	–	–	0 when being read.
		DE	**COLOR**	Color/mono select	1 Color 0 Mono	0	R/W	
		DD	**MASK**	FPSHIFT mask enable	1 Enabled 0 Disabled	0	R/W	
		DC	**–**	reserved	–	–	–	0 when being read.
		DB	**DWT1**	LCD panel data width	DWT[1:0] Data format	0	R/W	
		DA	**DWT0**		1 1 8-bit (format2)	0		
					1 0 reserved			
					0 1 8-bit (format1)			
					0 0 4-bit			
		D9	**SWINV**	Software video invert	1 Inverted 0 Normal	0	R/W	
		D8	**BLANK**	Display blank enable	1 Blank 0 Normal	0	R/W	
		D7–0	**–**	reserved	–	–	–	0 when being read.
Screen start address register	**0380210 (HW)**	DF	**VRAMSEL**	VRAM select	1 External 0 Internal	0	R/W	
		DE	**SADDR14**	Screen start address	0x0 to 0x7FFF	0	R/W	
		DD	**SADDR13**	SADDR14 = MSB		0		
		DC	**SADDR12**	SADDR0 = LSB		0		
		DB	**SADDR11**			0		
		DA	**SADDR10**			0		
		D9	**SADDR9**			0		
		D8	**SADDR8**			0		
		D7	**SADDR7**			0		
		D6	**SADDR6**			0		
		D5	**SADDR5**			0		
		D4	**SADDR4**			0		
		D3	**SADDR3**			0		
		D2	**SADDR2**			0		
		D1	**SADDR1**			0		
		D0	**SADDR0**			0		
Look-up table R/G data register (Address 0)	**0380400 (HW)**	DF	**LUTG05**	Look-up table G data	0x0 to 0x3F	X	R/W	0 when being read.
		DE	**LUTG04**	LUTG05 = MSB		X		
		DD	**LUTG03**	LUTG00 = LSB		X		
		DC	**LUTG02**			X		
		DB	**LUTG01**			X		
		DA	**LUTG00**			X		
		D9–8	**–**	reserved	–	–	–	0 when being read.
		D7	**LUTR05**	Look-up table R data	0x0 to 0x3F	X	R/W	
		D6	**LUTR04**	LUTR05 = MSB		X		
		D5	**LUTR03**	LUTR00 = LSB		X		
		D4	**LUTR02**			X		
		D3	**LUTR01**			X		
		D2	**LUTR00**			X		
		D1–0	**–**	reserved	–	–	–	0 when being read.
Look-up table B data register (Address 0)	**0380402 (HW)**	DF–8	**–**	reserved	–	–	–	0 when being read.
		D7	**LUTB05**	Look-up table B data	0x0 to 0x3F	X	R/W	
		D6	**LUTB04**	LUTB05 = MSB		X		
		D5	**LUTB03**	LUTB00 = LSB		X		
		D4	**LUTB02**			X		
		D3	**LUTB01**			X		
		D2	**LUTB00**			X		
		D1–0	**–**	reserved	–	–	–	0 when being read.

– – – 0x380404 ... 0x3807FA – – –

Look-up table R/G data register (Address FF)	**03807FC (HW)**	DF	**LUTGFF5**	Look-up table G data	0x0 to 0x3F	X	R/W	0 when being read.
		DE	**LUTGFF4**	LUTGFF5 = MSB		X		
		DD	**LUTGFF3**	LUTGFF0 = LSB		X		
		DC	**LUTGFF2**			X		
		DB	**LUTGFF1**			X		
		DA	**LUTGFF0**			X		
		D9–8	**–**	reserved	–	–	–	0 when being read.
		D7	**LUTRFF5**	Look-up table R data	0x0 to 0x3F	X	R/W	
		D6	**LUTRFF4**	LUTRFF5 = MSB		X		
		D5	**LUTRFF3**	LUTRFF0 = LSB		X		
		D4	**LUTRFF2**			X		
		D3	**LUTRFF1**			X		
		D2	**LUTRFF0**			X		
		D1–0	**–**	reserved	–	–	–	0 when being read.
Look-up table B data register (Address FF)	**03807FE (HW)**	DF–8	**–**	reserved	–	–	–	0 when being read.
		D7	**LUTBFF5**	Look-up table B data	0x0 to 0x3F	X	R/W	
		D6	**LUTBFF4**	LUTBFF5 = MSB		X		
		D5	**LUTBFF3**	LUTBFF0 = LSB		X		
		D4	**LUTBFF2**			X		
		D3	**LUTBFF1**			X		
		D2	**LUTBFF0**			X		
		D1–0	**–**	reserved	–	–	–	0 when being read.

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Bus arbiter control register	**0390000 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**PARKID1**	Parked bus master ID	PARKID[1:0] Bus master ID	0	R/W	
		D2	**PARKID0**		1 1 3 (reserved)	0		
					1 0 2 (reserved)			
					0 1 1 (LCDC)			
					0 0 0 (BCU)			
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**ARBIEN**	Bus arbiter enable	1 Enabled 0 Disabled	0	R/W	
Bus arbiter wait control register	**0390002 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**ARBWT3**	Bus arbiter wait control	0 to 15	1	R/W	
		D2	**ARBWT2**			1		
		D1	**ARBWT1**			1		
		D0	**ARBWT0**			1		
SPI receive data register	**03A0000 (HW)**	DF	**SPIRDF**	Received data from serial input	0x0 to 0xFFFF	X	R	
		DE	**SPIRDE**	SPIRDF = MSB		X		
		DD	**SPIRDD**	SPIRD0 = LSB		X		
		DC	**SPIRDC**			X		
		DB	**SPIRDB**			X		
		DA	**SPIRDA**			X		
		D9	**SPIRD9**			X		
		D8	**SPIRD8**			X		
		D7	**SPIRD7**			X		
		D6	**SPIRD6**			X		
		D5	**SPIRD5**			X		
		D4	**SPIRD4**			X		
		D3	**SPIRD3**			X		
		D2	**SPIRD2**			X		
		D1	**SPIRD1**			X		
		D0	**SPIRD0**			X		
SPI transmit data register	**03A0002 (HW)**	DF	**SPITDF**	Transmit data to serial output	0x0 to 0xFFFF	X	W	
		DE	**SPITDE**	SPITDF = MSB		X		
		DD	**SPITDD**	SPITD0 = LSB		X		
		DC	**SPITDC**			X		
		DB	**SPITDB**			X		
		DA	**SPITDA**			X		
		D9	**SPITD9**			X		
		D8	**SPITD8**			X		
		D7	**SPITD7**			X		
		D6	**SPITD6**			X		
		D5	**SPITD5**			X		
		D4	**SPITD4**			X		
		D3	**SPITD3**			X		
		D2	**SPITD2**			X		
		D1	**SPITD1**			X		
		D0	**SPITD0**			X		
SPI control register 1	**03A0004 (HW)**	DF–E	**–**	reserved	–	–	–	0 when being read.
		DD	**BPT3**	Number of data bits per transfer	Number of data bits per transfer = BPT + 1	0	R/W	
		DC	**BPT2**			0		
		DB	**BPT1**			0		
		DA	**BPT0**			0		
		D9	**CPHA**	SPICLK phase selection	1 Phase 1 0 Phase 0	0	R/W	
		D8	**CPOL**	SPICLK polarity selection	1 Active low 0 Active high	0	R/W	
		D7	**–**	reserved	–	–	–	0 when being read.
		D6	**MCBR2**	Master clock bit rate (in master mode only)	Master clock divided value = 4×2^{MCBR}	0	R/W	
		D5	**MCBR1**			0		
		D4	**MCBR0**			0		
		D3	**CLKS**	Shift clock source selection	Fixed set "0"	0	R/W	
		D2	**–**	reserved	–	–	–	0 when being read.
		D1	**MODE**	SPI mode selection	Fixed set "1"	0	R/W	
		D0	**ENA**	SPI enable	1 Enabled 0 Disabled	0	R/W	
SPI control register 2	**03A0006 (HW)**	DF–B	**–**	reserved	–	–	–	0 when being read.
		DA	**SS**	#SCS control	reserved, fixed at "0"	0	R/W	Master mode
		D9	**SSP**	#SCS polarity selection	reserved, fixed at "0"	0	R/W	Slave mode
		D8	**SSC**	#SCS configuration	reserved, fixed at "0"	0	R/W	Master mode
		D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RDYS**	#SRDY type selection	reserved, fixed at "0"	0	R/W	
		D0	**RDYE**	#SRDY enable	reserved, fixed at "0"	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI wait register	**03A0008 (HW)**	DF	**SPIWF**	Wait cycle control (in master mode only) SPIWF = MSB SPIW0 = LSB	0x0 to 0xFFFF	0	R/W	
		DE	**SPIWE**			0		
		DD	**SPIWD**			0		
		DC	**SPIWC**			0		
		DB	**SPIWB**			0		
		DA	**SPIWA**			0		
		D9	**SPIW9**			0		
		D8	**SPIW8**			0		
		D7	**SPIW7**			0		
		D6	**SPIW6**			0		
		D5	**SPIW5**			0		
		D4	**SPIW4**			0		
		D3	**SPIW3**			0		
		D2	**SPIW2**			0		
		D1	**SPIW1**			0		
		D0	**SPIW0**			0		
SPI status register	**03A000A (HW)**	DF–7	**–**	reserved	–	–	–	0 when being read.
		D6	**BSYF**	Transfer busy flag	1 Busy 0 Idle	0	R	Master mode
		D5	**–**	reserved	–	–	–	0 when being read.
		D4	**TDEF**	Transmit data empty flag	1 Empty 0 Not empty	1	R	
		D3	**–**	reserved	–	–	–	
		D2	**RDFF**	Receive data full flag	1 Full 0 Not full	0	R	
		D1–0	**–**	reserved	–	–	–	0 when being read.
SPI interrupt control register	**03A000C (HW)**	DF–6	**–**	reserved	–	–	–	0 when being read.
		D5	**MFIE**	Mode-fault interrupt enable	reserved, fixed at "0"	0	R/W	
		D4	**TEIE**	Data transmit interrupt enable	1 Enabled 0 Disabled	0	R/W	
		D3	**ROIE**	Receive overflow interrupt enable	reserved, fixed at "0"	0	R/W	
		D2	**RFIE**	Data receive interrupt enable	1 Enabled 0 Disabled	0	R/W	
		D1	**MIRQ**	Manual IRQ set/clear	1 Set 0 Clear	0	R/W	
		D0	**IRQE**	Interrupt request enable	1 Enabled 0 Disabled	0	R/W	
Sequential ROM I/F #CE area select register	**03A0100 (HW)**	DF–2	**–**	reserved	–	–	–	0 when being read.
		D1	**SQCES1**	Sequential ROM I/F #CE area selection (depends on CEFUNC)	SQCES[1:0] = 1 1: #CE9/#CE17(+18)	0	R/W	
		D0	**SQCES0**		1 0: #CE8/#CE14	0		
					0 1: #CE7/#CE13			
					0 0: #CE5/#CE15(+16)			
SDRAM control register	**03A0200 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**SDON**	SDRAM controller enable	1 Enabled 0 Disabled	0	R/W	
		D2	**ADDRC2**	SDRAM address configuration	ADDRC[2:0] Configuration	0	R/W	
		D1	**ADDRC1**		1 1 1: reserved	0		
		D0	**ADDRC0**		1 1 0: 16M x 8 bits x 2	0		
					1 0 1: 8M x 8 bits x 2			
					1 0 0: 2M x 8 bits x 2			
					0 1 1: 16M x 16 bits x 1			
					0 1 0: 8M x 16 bits x 1			
					0 0 1: 4M x 16 bits x 1			
					0 0 0: 1M x 16 bits x 1			
SDRAM initial register	**03A0202 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**SDEN**	SDRAM initialize flag	1 Initialized 0 Not initialized	0	R	
		D2	**INIMRS**	MRS trigger	1 Trigger 0 Invalid	0	W	0 when being read.
		D1	**INIPRE**	PRE trigger	1 Trigger 0 Invalid	0	W	
		D0	**INIREF**	REF trigger	1 Trigger 0 Invalid	0	W	
SDRAM auto-refresh counter register	**03A0204 (HW)**	DF–C	**–**	reserved	–	–	–	0 when being read.
		DB	**AURCO11**	SDRAM auto-refresh counter	0x0 to 0xFFF	0	R/W	
		DA	**AURCO10**			0		
		D9	**AURCO9**			0		
		D8	**AURCO8**			0		
		D7	**AURCO7**			0		
		D6	**AURCO6**			0		
		D5	**AURCO5**			0		
		D4	**AURCO4**			0		
		D3	**AURCO3**			0		
		D2	**AURCO2**			0		
		D1	**AURCO1**			0		
		D0	**AURCO0**			0		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SDRAM self-refresh counter register	**03A0206 (HW)**	DF–A	**–**	reserved	–	–	–	0 when being read.
		D9	**SELDO**	SDRAM self-refresh status	1 Refresh mode 0 Done	0	R	
		D8	**SCKON**	SDRAM clock during self-refresh	1 Enabled 0 Disabled	0	R/W	
		D7	**SELEN**	SDRAM self-refresh enable	1 Enabled 0 Disabled	0	R/W	
		D6	**SELCO6**	SDRAM self-refresh counter	0x0 to 0x7F	1	R/W	
		D5	**SELCO5**			1		
		D4	**SELCO4**			1		
		D3	**SELCO3**			1		
		D2	**SELCO2**			1		
		D1	**SELCO1**			1		
		D0	**SELCO0**			1		
SDRAM timing setup register	**03A0208 (HW)**	DF–3	**–**	reserved	–	–	–	0 when being read.
		D2	**T80NS1**	SDRAM t_{RC}, t_{RFC} and t_{XSR} cycles	T80NS[1:0] Number of cycles	1	R/W	
		D1	**T80NS0**		1 1 5 cycles	1		
					1 0 4 cycles			
					0 1 3 cycles			
					0 0 2 cycles			
		D0	**T24NS**	SDRAM t_{RP} and t_{RCD} cycles	1 2 cycles 0 1 cycle	1	R/W	
SDRAM application configuration register	**03A0210 (HW)**	DF–4	**–**	reserved	–	–	–	0 when being read.
		D3	**SDHP**	SDRAM high-performance access	1 High perform. 0 Low perform.	0	R/W	
		D2	**AR14C**	Area 8/14 configuration	1 SDRAM 0 Other	0	R/W	
		D1	**AR13C**	Area 7/13 configuration	1 SDRAM 0 Other	0	R/W	
		D0	**IQBON**	Instruction queue buffer enable	1 Enabled 0 Disabled	0	R/W	

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
MainIntStat	**1000000**	D7	**SIE_IntStat**	1	SIE interrupts	0	None	0	R	
(Main interrupt	**(B)**	D6	**EPrIntStat**	1	EPr interrupts	0	None	0	R	
status)		D5	**DMA_IntStat**	1	DMA interrupts	0	None	0	R	
		D4	**FIFO_IntStat**	1	FIFO interrupts	0	None	0	R	
		D3–2	**–**	–				–	–	0 when being read.
		D1	**EP0IntStat**	1	EP0 interrupts	0	None	0	R	
		D0	**RcvEP0SETUP**	1	Receive EP0 SETUP	0	None	0	R(W)	
SIE_IntStat	**1000001**	D7	**VBUS_Changed**	1	VBUS is changed	0	None	0	R(W)	
(SIE interrupt	**(B)**	D6	**NonJ**	1	Detect non J state	0	None	0	R(W)	
status)		D5	**DetectReset**	1	Detect USB reset	0	None	0	R(W)	
		D4	**DetectSuspend**	1	Detect USB suspend	0	None	0	R(W)	
		D3	**RcvSOF**	1	Receive SOF token	0	None	0	R(W)	
		D2	**DetectJ**	1	Detect J state	0	None	0	R(W)	
		D1	**–**	–				–	–	0 when being read.
		D0	**SetAddressCmp**	1	AutoSetAddress complete	0	None	0	R(W)	
EPrIntStat	**1000002**	D7–4	**–**	–				–	–	0 when being read.
(EPr interrupt	**(B)**	D3	**EPdIntStat**	1	EPd interrupt	0	None	0	R	
status)		D2	**EPcIntStat**	1	EPc interrupt	0	None	0	R	
		D1	**EPbIntStat**	1	EPb interrupt	0	None	0	R	
		D0	**EPaIntStat**	1	EPa interrupt	0	None	0	R	
DMA_IntStat	**1000003**	D7–2	**–**	–				–	–	0 when being read.
(DMA interrupt	**(B)**	D1	**DMA_CountUp**	1	DMA counter overflow	0	None	0	R(W)	
status)		D0	**DMA_Cmp**	1	DMA complete	0	None	0	R(W)	
FIFO_IntStat	**1000004**	D7	**DescriptorCmp**	1	Descriptor complete	0	None	0	R(W)	
(FIFO interrupt	**(B)**	D6–2	**–**	–				–	–	0 when being read.
status)		D1	**FIFO_IN_Cmp**	1	IN FIFO Complete	0	None	0	R(W)	
		D0	**FIFO_OUT_Cmp**	1	OUT FIFO complete	0	None	0	R(W)	
EP0IntStat	**1000007**	D7–6	**–**	–				–	–	0 when being read.
(EP0 interrupt	**(B)**	D5	**IN_TranACK**	1	In transaction ACK	0	None	0	R(W)	
status)		D4	**OUT_TranACK**	1	Out transaction ACK	0	None	0	R(W)	
		D3	**IN_TranNAK**	1	In transaction NAK	0	None	0	R(W)	
		D2	**OUT_TranNAK**	1	Out transaction NAK	0	None	0	R(W)	
		D1	**IN_TranErr**	1	In transaction error	0	None	0	R(W)	
		D0	**OUT_TranErr**	1	Out transaction error	0	None	0	R(W)	
EPaIntStat	**1000008**	D7	**–**	–				–	–	0 when being read.
(EPa interrupt	**(B)**	D6	**OUT_ShortACK**	1	Out short packet ACK	0	None	0	R(W)	
status)		D5	**IN_TranACK**	1	In transaction ACK	0	None	0	R(W)	
		D4	**OUT_TranACK**	1	Out transaction ACK	0	None	0	R(W)	
		D3	**IN_TranNAK**	1	In transaction NAK	0	None	0	R(W)	
		D2	**OUT_TranNAK**	1	Out transaction NAK	0	None	0	R(W)	
		D1	**IN_TranErr**	1	In transaction error	0	None	0	R(W)	
		D0	**OUT_TranErr**	1	Out transaction error	0	None	0	R(W)	
EPbIntStat	**1000009**	D7	**–**	–				–	–	0 when being read.
(EPb interrupt	**(B)**	D6	**OUT_ShortACK**	1	Out short packet ACK	0	None	0	R(W)	
status)		D5	**IN_TranACK**	1	In transaction ACK	0	None	0	R(W)	
		D4	**OUT_TranACK**	1	Out transaction ACK	0	None	0	R(W)	
		D3	**IN_TranNAK**	1	In transaction NAK	0	None	0	R(W)	
		D2	**OUT_TranNAK**	1	Out transaction NAK	0	None	0	R(W)	
		D1	**IN_TranErr**	1	In transaction error	0	None	0	R(W)	
		D0	**OUT_TranErr**	1	Out transaction error	0	None	0	R(W)	
EPcIntStat	**100000A**	D7	**–**	–				–	–	0 when being read.
(EPc interrupt	**(B)**	D6	**OUT_ShortACK**	1	Out short packet ACK	0	None	0	R(W)	
status)		D5	**IN_TranACK**	1	In transaction ACK	0	None	0	R(W)	
		D4	**OUT_TranACK**	1	Out transaction ACK	0	None	0	R(W)	
		D3	**IN_TranNAK**	1	In transaction NAK	0	None	0	R(W)	
		D2	**OUT_TranNAK**	1	Out transaction NAK	0	None	0	R(W)	
		D1	**IN_TranErr**	1	In transaction error	0	None	0	R(W)	
		D0	**OUT_TranErr**	1	Out transaction error	0	None	0	R(W)	
EPdIntStat	**100000B**	D7	**–**	–				–	–	0 when being read.
(EPd interrupt	**(B)**	D6	**OUT_ShortACK**	1	Out short packet ACK	0	None	0	R(W)	
status)		D5	**IN_TranACK**	1	In transaction ACK	0	None	0	R(W)	
		D4	**OUT_TranACK**	1	Out transaction ACK	0	None	0	R(W)	
		D3	**IN_TranNAK**	1	In transaction NAK	0	None	0	R(W)	
		D2	**OUT_TranNAK**	1	Out transaction NAK	0	None	0	R(W)	
		D1	**IN_TranErr**	1	In transaction error	0	None	0	R(W)	
		D0	**OUT_TranErr**	1	Out transaction error	0	None	0	R(W)	

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
MainIntEnb (Main interrupt enable)	**1000010 (B)**	D7	En**SIE_IntStat**	1	Enabled	0	Disabled	0	R/W	
		D6	En**EPrIntStat**					0	R/W	
		D5	**EnDMA_IntStat**					0	R/W	
		D4	**EnFIFO_IntStat**					0	R/W	
		D3–2	**–**	–				–	–	0 when being read.
		D1	En**EP0IntStat**	1	Enabled	0	Disabled	0	R/W	
		D0	En**RcvEP0SETUP**					0	R/W	
SIE_IntEnb (SIE interrupt enable)	**1000011 (B)**	D7	**EnVBUS_Changed**	1	Enabled	0	Disabled	0	R/W	
		D6	**EnNonJ**					0	R/W	
		D5	**EnDetectReset**					0	R/W	
		D4	**EnDetectSuspend**					0	R/W	
		D3	**EnRcvSOF**					0	R/W	
		D2	**EnDetectJ**					0	R/W	
		D1	**–**	–				–	–	0 when being read.
		D0	**EnSetAddressCmp**	1	Enabled	0	Disabled	0	R/W	
EPrIntEnb (EPr interrupt enable)	**1000012 (B)**	D7–4	**–**	–				–	–	0 when being read.
		D3	**EnEPdIntStat**	1	Enabled	0	Disabled	0	R/W	
		D2	**EnEPcIntStat**					0	R/W	
		D1	**EnEPbIntStat**					0	R/W	
		D0	**EnEPaIntStat**					0	R/W	
DMA_IntEnb (DMA interrupt enable)	**1000013 (B)**	D7–2	**–**	–				–	–	0 when being read.
		D1	**EnDMA_CountUp**	1	Enabled	0	Disabled	0	R/W	
		D0	**EnDMA_Cmp**					0	R/W	
FIFO_IntEnb (FIFO interrupt enable)	**1000014 (B)**	D7	**EnDescriptorCmp**	1	Enabled	0	Disabled	0	R/W	
		D6–2	**–**	–				–	–	0 when being read.
		D1	**EnFIFO_IN_Cmp**	1	Enabled	0	Disabled	0	R/W	
		D0	**EnFIFO_OUT_Cmp**					0	R/W	
EP0IntEnb (EP0 interrupt enable)	**1000017 (B)**	D7–6	**–**	–				–	–	0 when being read.
		D5	**EnIN_TranACK**	1	Enabled	0	Disabled	0	R/W	
		D4	**EnOUT_TranACK**					0	R/W	
		D3	**EnIN_TranNAK**					0	R/W	
		D2	**EnOUT_TranNAK**					0	R/W	
		D1	**EnIN_TranErr**					0	R/W	
		D0	**EnOUT_TranErr**					0	R/W	
EPaIntEnb (EPa interrupt enable)	**1000018 (B)**	D7	**–**	–				–	–	0 when being read.
		D6	**EnOUT_ShortACK**	1	Enabled	0	Disabled	0	R/W	
		D5	**EnIN_TranACK**					0	R/W	
		D4	**EnOUT_TranACK**					0	R/W	
		D3	**EnIN_TranNAK**					0	R/W	
		D2	**EnOUT_TranNAK**					0	R/W	
		D1	**EnIN_TranErr**					0	R/W	
		D0	**EnOUT_TranErr**					0	R/W	
EPbIntEnb (EPb interrupt enable)	**1000019 (B)**	D7	**–**	–				–	–	0 when being read.
		D6	**EnOUT_ShortACK**	1	Enabled	0	Disabled	0	R/W	
		D5	**EnIN_TranACK**					0	R/W	
		D4	**EnOUT_TranACK**					0	R/W	
		D3	**EnIN_TranNAK**					0	R/W	
		D2	**EnOUT_TranNAK**					0	R/W	
		D1	**EnIN_TranErr**					0	R/W	
		D0	**EnOUT_TranErr**					0	R/W	
EPcIntEnb (EPc interrupt enable)	**100001A (B)**	D7	**–**	–				–	–	0 when being read.
		D6	**EnOUT_ShortACK**	1	Enabled	0	Disabled	0	R/W	
		D5	**EnIN_TranACK**					0	R/W	
		D4	**EnOUT_TranACK**					0	R/W	
		D3	**EnIN_TranNAK**					0	R/W	
		D2	**EnOUT_TranNAK**					0	R/W	
		D1	**EnIN_TranErr**					0	R/W	
		D0	**EnOUT_TranErr**					0	R/W	
EPdIntEnb (EPd interrupt enable)	**100001B (B)**	D7	**–**	–				–	–	0 when being read.
		D6	**EnOUT_ShortACK**	1	Enabled	0	Disabled	0	R/W	
		D5	**EnIN_TranACK**					0	R/W	
		D4	**EnOUT_TranACK**					0	R/W	
		D3	**EnIN_TranNAK**					0	R/W	
		D2	**EnOUT_TranNAK**					0	R/W	
		D1	**EnIN_TranErr**					0	R/W	
		D0	**EnOUT_TranErr**					0	R/W	

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
RevisionNum (Revision number)	1000020 (B)	D7	RevisionNum[7]	Revision number (0x12)	0	R	
		D6	RevisionNum[6]		0		
		D5	RevisionNum[5]		0		
		D4	RevisionNum[4]		1		
		D3	RevisionNum[3]		0		
		D2	RevisionNum[2]		0		
		D1	RevisionNum[1]		1		
		D0	RevisionNum[0]		0		
USB_Control (USB control register)	1000021 (B)	D7	DisBusDetect	1 Disable bus detect / 0 Enable bus detect	0	R/W	
		D6	EnAutoNego	1 Enable auto negotiation / 0 Disable auto negotiation	0	R/W	
		D5	InSUSPEND	1 Monitor NonJ / 0 Do nothing	0	R/W	
		D4	StartDetectJ	1 Start J-state detection / 0 Do nothing	0	R/W	
		D3	SendWakeup	1 Send remote wakeup signal / 0 Do nothing	0	R/W	
		D2–1	–	–	–	–	0 when being read.
		D0	ActiveUSB	1 Activate USB / 0 Disactivate USB	0	R/W	
USB_Status (USB status register)	1000022 (B)	D7	VBUS	1 VBUS=High / 0 VBUS=Low	X	R	
		D6	FS	1 FS mode (fixed) / 0 –	1	R	
		D5–2	–	–	–	–	
		D1	LineState[1]	LineState[1:0] = 1 1: SE1; 1 0: K; 0 1: J; 0 0: SE0	X	R	
		D0	LineState[0]		X		
XcvrControl (Xcvr control register)	1000023 (B)	D7	RpuEnb	1 Enable pull-up / 0 Disable pull-up	1	R/W	
		D6–2	–	–	–	–	0 when being read.
		D1	OpMode[1]	OpMode[1:0] = 1 1: reserved; 1 0: Disable bitstuffing and NRZI encoding; 0 1: Non-driving; 0 0: Normal operation	0	R/W	
		D0	OpMode[0]		1		
USB_Test (USB test)	1000024 (B)	D7	EnUSB_Test	1 Enable USB test / 0 Do nothing	0	R/W	
		D6–4	–	–	–	–	0 when being read.
		D3	Test_SE0_NAK	1 Test_SE0_NAK / 0 Do nothing	0	R/W	
		D2	Test_J	1 Test_J / 0 Do nothing	0	R/W	
		D1	Test_K	1 Test_K / 0 Do nothing	0	R/W	
		D0	Test_Packet	1 Test_Packet / 0 Do nothing	0	R/W	
EPnControl (Endpoint control)	1000025 (B)	D7	AllForceNAK	1 Set all ForceNAK / 0 Do nothing	0	W	0 when being read.
		D6	EPrForceSTALL	1 Set EP's ForceSTALL / 0 Do nothing	0	W	
		D5	AllFIFO_Clr	1 Clear all FIFO / 0 Do nothing	0	W	
		D4–1	–	–	–	–	
		D0	EP0FIFO_Clr	1 Clear EP0 FIFO / 0 Do nothing	0	W	
EPrFIFO_Clr (EPr FIFO clear)	1000026 (B)	D7–4	–	–	–	–	0 when being read.
		D3	EPdFIFO_Clr	1 Clear EPd FIFO / 0 Do nothing	0	W	
		D2	EPcFIFO_Clr	1 Clear EPc FIFO / 0 Do nothing	0	W	
		D1	EPbFIFO_Clr	1 Clear EPb FIFO / 0 Do nothing	0	W	
		D0	EPaFIFO_Clr	1 Clear EPa FIFO / 0 Do nothing	0	W	
FrameNumber_H (Frame number high)	100002E (B)	D7	FnInvalid	1 Invalid frame number / 0 Valid frame number	1	R	
		D6–3	–	–	–	–	0 when being read.
		D2	FrameNumber[10]	Frame number high	0	R	
		D1	FrameNumber[9]		0		
		D0	FrameNumber[8]		0		
FrameNumber_L (Frame number low)	100002F (B)	D7	FrameNumber[7]	Frame number low	0	R	
		D6	FrameNumber[6]		0		
		D5	FrameNumber[5]		0		
		D4	FrameNumber[4]		0		
		D3	FrameNumber[3]		0		
		D2	FrameNumber[2]		0		
		D1	FrameNumber[1]		0		
		D0	FrameNumber[0]		0		
EP0Setup_0 (EP0 set-up 0)	1000030 (B)	D7	EP0Setup_0[7]	Endpoint 0 set-up data 0	0	R	
		D6	EP0Setup_0[6]		0		
		D5	EP0Setup_0[5]		0		
		D4	EP0Setup_0[4]		0		
		D3	EP0Setup_0[3]		0		
		D2	EP0Setup_0[2]		0		
		D1	EP0Setup_0[1]		0		
		D0	EP0Setup_0[0]		0		

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EP0Setup_1	**1000031**	D7	**EP0Setup_1[7]**	Endpoint 0 set-up data 1	0	R	
(EP0 set-up 1)	**(B)**	D6	**EP0Setup_1[6]**		0		
		D5	**EP0Setup_1[5]**		0		
		D4	**EP0Setup_1[4]**		0		
		D3	**EP0Setup_1[3]**		0		
		D2	**EP0Setup_1[2]**		0		
		D1	**EP0Setup_1[1]**		0		
		D0	**EP0Setup_1[0]**		0		
EP0Setup_2	**1000032**	D7	**EP0Setup_2[7]**	Endpoint 0 set-up data 2	0	R	
(EP0 set-up 2)	**(B)**	D6	**EP0Setup_2[6]**		0		
		D5	**EP0Setup_2[5]**		0		
		D4	**EP0Setup_2[4]**		0		
		D3	**EP0Setup_2[3]**		0		
		D2	**EP0Setup_2[2]**		0		
		D1	**EP0Setup_2[1]**		0		
		D0	**EP0Setup_2[0]**		0		
EP0Setup_3	**1000033**	D7	**EP0Setup_3[7]**	Endpoint 0 set-up data 3	0	R	
(EP0 set-up 3)	**(B)**	D6	**EP0Setup_3[6]**		0		
		D5	**EP0Setup_3[5]**		0		
		D4	**EP0Setup_3[4]**		0		
		D3	**EP0Setup_3[3]**		0		
		D2	**EP0Setup_3[2]**		0		
		D1	**EP0Setup_3[1]**		0		
		D0	**EP0Setup_3[0]**		0		
EP0Setup_4	**1000034**	D7	**EP0Setup_4[7]**	Endpoint 0 set-up data 4	0	R	
(EP0 set-up 4)	**(B)**	D6	**EP0Setup_4[6]**		0		
		D5	**EP0Setup_4[5]**		0		
		D4	**EP0Setup_4[4]**		0		
		D3	**EP0Setup_4[3]**		0		
		D2	**EP0Setup_4[2]**		0		
		D1	**EP0Setup_4[1]**		0		
		D0	**EP0Setup_4[0]**		0		
EP0Setup_5	**1000035**	D7	**EP0Setup_5[7]**	Endpoint 0 set-up data 5	0	R	
(EP0 set-up 5)	**(B)**	D6	**EP0Setup_5[6]**		0		
		D5	**EP0Setup_5[5]**		0		
		D4	**EP0Setup_5[4]**		0		
		D3	**EP0Setup_5[3]**		0		
		D2	**EP0Setup_5[2]**		0		
		D1	**EP0Setup_5[1]**		0		
		D0	**EP0Setup_5[0]**		0		
EP0Setup_6	**1000036**	D7	**EP0Setup_6[7]**	Endpoint 0 set-up data 6	0	R	
(EP0 set-up 6)	**(B)**	D6	**EP0Setup_6[6]**		0		
		D5	**EP0Setup_6[5]**		0		
		D4	**EP0Setup_6[4]**		0		
		D3	**EP0Setup_6[3]**		0		
		D2	**EP0Setup_6[2]**		0		
		D1	**EP0Setup_6[1]**		0		
		D0	**EP0Setup_6[0]**		0		
EP0Setup_7	**1000037**	D7	**EP0Setup_7[7]**	Endpoint 0 set-up data 7	0	R	
(EP0 set-up 7)	**(B)**	D6	**EP0Setup_7[6]**		0		
		D5	**EP0Setup_7[5]**		0		
		D4	**EP0Setup_7[4]**		0		
		D3	**EP0Setup_7[3]**		0		
		D2	**EP0Setup_7[2]**		0		
		D1	**EP0Setup_7[1]**		0		
		D0	**EP0Setup_7[0]**		0		
USB_Address	**1000038**	D7	**AutoSetAddress**	1 Auto set address 0 Do nothing	0	R/W	
(USB address)	**(B)**	D6	**USB_Address[6]**	USB address	0	R/W	
		D5	**USB_Address[5]**		0		
		D4	**USB_Address[4]**		0		
		D3	**USB_Address[3]**		0		
		D2	**USB_Address[2]**		0		
		D1	**USB_Address[1]**		0		
		D0	**USB_Address[0]**		0		
EP0Control	**1000039**	D7	**INxOUT**	1 IN 0 OUT	0	R/W	
(EP0 control)	**(B)**	D6–1	–	–	–	–	0 when being read.
		D0	**ReplyDescriptor**	1 Reply descriptor 0 Do nothing	0	W	

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EP0ControlIN	100003A	D7	–	–				–	–	0 when being read.
(EP0 control	(B)	D6	EnShortPkt	1	Enable short packet	0	Do nothing	0	R/W	
IN)		D5	–	–				–	–	0 when being read.
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	R/W	
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	R/W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	
EP0ControlOUT	100003B	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	
(EP0 control	(B)	D6–5	–	–				–	–	0 when being read.
OUT)		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	
EP0MaxSize	100003F	D7	–	–				–	–	0 when being read.
(EP0 max	(B)	D6	EP0MaxSize[6]	Endpoint EP0 max packet size				0	R/W	
packet size)		D5	EP0MaxSize[5]					0		
		D4	EP0MaxSize[4]					0		
		D3	EP0MaxSize[3]					1		
		D2–0	–	–				–	–	0 when being read.
EPaControl	1000040	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	
(EPa control)	(B)	D6	EnShortPkt	1	Enable short packet	0	Do nothing	0	R/W	
		D5	DisAF_NAK_Short	1	Disable auto force	0	Auto force NAK short	0	R/W	
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	
EPbControl	1000041	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	
(EPb control)	(B)	D6	EnShortPkt	1	Enable short packet	0	Do nothing	0	R/W	
		D5	DisAF_NAK_Short	1	Disable auto force	0	Auto force NAK short	0	R/W	
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	
EPcControl	1000042	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	
(EPc control)	(B)	D6	EnShortPkt	1	Enable short packet	0	Do nothing	0	R/W	
		D5	DisAF_NAK_Short	1	Disable auto force	0	Auto force NAK short	0	R/W	
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	
EPdControl	1000043	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	
(EPd control)	(B)	D6	EnShortPkt	1	Enable short packet	0	Do nothing	0	R/W	
		D5	DisAF_NAK_Short	1	Disable auto force	0	Auto force NAK short	0	R/W	
		D4	ToggleStat	Toggle sequence bit				0	R	
		D3	ToggleSet	1	Set toggle sequence bit	0	Do nothing	0	W	0 when being read.
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W	
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W	
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W	
EPaMaxSize_H	1000050	D7–2	–	–				–	–	0 when being read.
(EPa max	(B)	D1	EPaMaxSize[9]	Endpoint EPa max packet size				0	R/W	
packet size		D0	EPaMaxSize[8]					0		
high)										
EPaMaxSize_L	1000051	D7	EPaMaxSize[7]	Endpoint EPa max packet size				0	R/W	
(EPa max	(B)	D6	EPaMaxSize[6]					0		
packet size		D5	EPaMaxSize[5]					0		
low)		D4	EPaMaxSize[4]					0		
		D3	EPaMaxSize[3]					0		
		D2	EPaMaxSize[2]					0		
		D1	EPaMaxSize[1]					0		
		D0	EPaMaxSize[0]					0		

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPaConfig_0 (EPa configuration 0)	1000052 (B)	D7	INxOUT	1	In	0	Out	0	R/W	
		D6	ToggleMode	1	Always toggle	0	Normal toggle	0	R/W	
		D5	EnEndPoint	1	Enable endpoint	0	Disable endpoint	0	R/W	
		D4	–	–				–	–	0 when being read.
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)				0	R/W	
		D2	EndPointNumber[2]					0		
		D1	EndPointNumber[1]					0		
		D0	EndPointNumber[0]					0		
EPaConfig_1 (EPa configuration 1)	1000053 (B)	D7	ISO	1	ISO	0	Non-ISO	0	R/W	
		D6	ISO_CRCmode	1	CRC mode	0	Normal ISO	0	R/W	
		D5–0	–	–				–	–	0 when being read.
EPbMaxSize_H (EPb max packet size high)	1000054 (B)	D7–2	–	–				–	–	0 when being read.
		D1	EPbMaxSize[9]	Endpoint EPb max packet size				0	R/W	
		D0	EPbMaxSize[8]					0		
EPbMaxSize_L (EPb max packet size low)	1000055 (B)	D7	EPbMaxSize[7]	Endpoint EPb max packet size				0	R/W	
		D6	EPbMaxSize[6]					0		
		D5	EPbMaxSize[5]					0		
		D4	EPbMaxSize[4]					0		
		D3	EPbMaxSize[3]					0		
		D2	EPbMaxSize[2]					0		
		D1	EPbMaxSize[1]					0		
		D0	EPbMaxSize[0]					0		
EPbConfig_0 (EPb configuration 0)	1000056 (B)	D7	INxOUT	1	In	0	Out	0	R/W	
		D6	ToggleMode	1	Always toggle	0	Normal toggle	0	R/W	
		D5	EnEndPoint	1	Enable endpoint	0	Disable endpoint	0	R/W	
		D4	–	–				–	–	0 when being read.
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)				0	R/W	
		D2	EndPointNumber[2]					0		
		D1	EndPointNumber[1]					0		
		D0	EndPointNumber[0]					0		
EPbConfig_1 (EPb configuration 1)	1000057 (B)	D7	ISO	1	ISO	0	Non-ISO	0	R/W	
		D6	ISO_CRCmode	1	CRC mode	0	Normal ISO	0	R/W	
		D5–0	–	–				–	–	0 when being read.
EPcMaxSize_H (EPc max packet size high)	1000058 (B)	D7–2	–	–				–	–	0 when being read.
		D1	EPcMaxSize[9]	Endpoint EPc max packet size				0	R/W	
		D0	EPcMaxSize[8]					0		
EPcMaxSize_L (EPc max packet size low)	1000059 (B)	D7	EPcMaxSize[7]	Endpoint EPc max packet size				0	R/W	
		D6	EPcMaxSize[6]					0		
		D5	EPcMaxSize[5]					0		
		D4	EPcMaxSize[4]					0		
		D3	EPcMaxSize[3]					0		
		D2	EPcMaxSize[2]					0		
		D1	EPcMaxSize[1]					0		
		D0	EPcMaxSize[0]					0		
EPcConfig_0 (EPc configuration 0)	100005A (B)	D7	INxOUT	1	In	0	Out	0	R/W	
		D6	ToggleMode	1	Always toggle	0	Normal toggle	0	R/W	
		D5	EnEndPoint	1	Enable endpoint	0	Disable endpoint	0	R/W	
		D4	–	–				–	–	0 when being read.
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)				0	R/W	
		D2	EndPointNumber[2]					0		
		D1	EndPointNumber[1]					0		
		D0	EndPointNumber[0]					0		
EPcConfig_1 (EPc configuration 1)	100005B (B)	D7	ISO	1	ISO	0	Non-ISO	0	R/W	
		D6	ISO_CRCmode	1	CRC mode	0	Normal ISO	0	R/W	
		D5–0	–	–				–	–	0 when being read.
EPdMaxSize_H (EPd max packet size high)	100005C (B)	D7–2	–	–				–	–	0 when being read.
		D1	EPdMaxSize[9]	Endpoint EPd max packet size				0	R/W	
		D0	EPdMaxSize[8]					0		
EPdMaxSize_L (EPd max packet size low)	100005D (B)	D7	EPdMaxSize[7]	Endpoint EPd max packet size				0	R/W	
		D6	EPdMaxSize[6]					0		
		D5	EPdMaxSize[5]					0		
		D4	EPdMaxSize[4]					0		
		D3	EPdMaxSize[3]					0		
		D2	EPdMaxSize[2]					0		
		D1	EPdMaxSize[1]					0		
		D0	EPdMaxSize[0]					0		

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
EPdConfig_0	**100005E**	D7	**INxOUT**	1	In	0	Out	0	R/W	
(EPd	**(B)**	D6	**ToggleMode**	1	Always toggle	0	Normal toggle	0	R/W	
configuration 0)		D5	**EnEndPoint**	1	Enable endpoint	0	Disable endpoint	0	R/W	
		D4	**–**	–				–	–	0 when being read.
		D3	**EndPointNumber[3]**	Endpoint number				0	R/W	
		D2	**EndPointNumber[2]**	(0x1 to 0xF)				0		
		D1	**EndPointNumber[1]**					0		
		D0	**EndPointNumber[0]**					0		
EPdConfig_1	**100005F**	D7	**ISO**	1	ISO	0	Non-ISO	0	R/W	
(EPd	**(B)**	D6	**ISO_CRCmode**	1	CRC mode	0	Normal ISO	0	R/W	
configuration 1)		D5–0	**–**	–				–	–	0 when being read.
EPaStartAdrs_H	**1000070**	D7–4	**–**	–				–	–	0 when being read.
(EPa FIFO start	**(B)**	D3	**EPaStartAdrs[11]**	Endpoint EPa start address				0	R/W	
address high)		D2	**EPaStartAdrs[10]**					0		
		D1	**EPaStartAdrs[9]**					0		
		D0	**EPaStartAdrs[8]**					0		
EPaStartAdrs_L	**1000071**	D7	**EPaStartAdrs[7]**	Endpoint EPa start address				0	R/W	
(EPa FIFO start	**(B)**	D6	**EPaStartAdrs[6]**					0		
address low)		D5	**EPaStartAdrs[5]**					0		
		D4	**EPaStartAdrs[4]**					0		
		D3	**EPaStartAdrs[3]**					0		
		D2	**EPaStartAdrs[2]**					0		
		D1–0	**–**	–				–	–	0 when being read.
EPbStartAdrs_H	**1000072**	D7–4	**–**	–				–	–	0 when being read.
(EPb FIFO start	**(B)**	D3	**EPbStartAdrs[11]**	Endpoint EPb start address				0	R/W	
address high)		D2	**EPbStartAdrs[10]**					0		
		D1	**EPbStartAdrs[9]**					0		
		D0	**EPbStartAdrs[8]**					0		
EPbStartAdrs_L	**1000073**	D7	**EPbStartAdrs[7]**	Endpoint EPb start address				0	R/W	
(EPb FIFO start	**(B)**	D6	**EPbStartAdrs[6]**					0		
address low)		D5	**EPbStartAdrs[5]**					0		
		D4	**EPbStartAdrs[4]**					0		
		D3	**EPbStartAdrs[3]**					0		
		D2	**EPbStartAdrs[2]**					0		
		D1–0	**–**	–				–	–	0 when being read.
EPcStartAdrs_H	**1000074**	D7–4	**–**	–				–	–	0 when being read.
(EPc FIFO start	**(B)**	D3	**EPcStartAdrs[11]**	Endpoint EPc start address				0	R/W	
address high)		D2	**EPcStartAdrs[10]**					0		
		D1	**EPcStartAdrs[9]**					0		
		D0	**EPcStartAdrs[8]**					0		
EPcStartAdrs_L	**1000075**	D7	**EPcStartAdrs[7]**	Endpoint EPc start address				0	R/W	
(EPc FIFO start	**(B)**	D6	**EPcStartAdrs[6]**					0		
address low)		D5	**EPcStartAdrs[5]**					0		
		D4	**EPcStartAdrs[4]**					0		
		D3	**EPcStartAdrs[3]**					0		
		D2	**EPcStartAdrs[2]**					0		
		D1–0	**–**	–				–	–	0 when being read.
EPdStartAdrs_H	**1000076**	D7–4	**–**	–				–	–	0 when being read.
(EPd FIFO start	**(B)**	D3	**EPdStartAdrs[11]**	Endpoint EPd start address				0	R/W	
address high)		D2	**EPdStartAdrs[10]**					0		
		D1	**EPdStartAdrs[9]**					0		
		D0	**EPdStartAdrs[8]**					0		
EPdStartAdrs_L	**1000077**	D7	**EPdStartAdrs[7]**	Endpoint EPd start address				0	R/W	
(EPd FIFO start	**(B)**	D6	**EPdStartAdrs[6]**					0		
address low)		D5	**EPdStartAdrs[5]**					0		
		D4	**EPdStartAdrs[4]**					0		
		D3	**EPdStartAdrs[3]**					0		
		D2	**EPdStartAdrs[2]**					0		
		D1–0	**–**	–				–	–	0 when being read.
CPU_JoinRd	**1000080**	D7–4	**–**	–				–	–	0 when being read.
(CPU join FIFO	**(B)**	D3	**JoinEPdRd**	1	Join EPd FIFO read	0	Do nothing	0	R/W	
read)		D2	**JoinEPcRd**	1	Join EPc FIFO read	0	Do nothing	0	R/W	
		D1	**JoinEPbRd**	1	Join EPb FIFO read	0	Do nothing	0	R/W	
		D0	**JoinEPaRd**	1	Join EPa FIFO read	0	Do nothing	0	R/W	

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
CPU_JoinWr (CPU join FIFO write)	1000081 (B)	D7–4	–	–	–	–	0 when being read.
		D3	JoinEPdWr	1 Join EPd FIFO write 0 Do nothing	0	R/W	
		D2	JoinEPcWr	1 Join EPc FIFO write 0 Do nothing	0	R/W	
		D1	JoinEPbWr	1 Join EPb FIFO write 0 Do nothing	0	R/W	
		D0	JoinEPaWr	1 Join EPa FIFO write 0 Do nothing	0	R/W	
EnEPnFIFO _Access (Enable EPn FIFO access)	1000082 (B)	D7–2	–	–	–	–	0 when being read.
		D1	EnEPnFIFO_Wr	1 Enable join EPn FIFO write 0 Do nothing	0	R/W	
		D0	EnEPnFIFO_Rd	1 Enable join EPn FIFO read 0 Do nothing	0	R/W	
EPnFIFOforCPU (EPn FIFO for CPU)	1000083 (B)	D7	EPnFIFOData[7]	Endpoint EP0 FIFO access from CPU	X	R/W	
		D6	EPnFIFOData[6]		X		
		D5	EPnFIFOData[5]		X		
		D4	EPnFIFOData[4]		X		
		D3	EPnFIFOData[3]		X		
		D2	EPnFIFOData[2]		X		
		D1	EPnFIFOData[1]		X		
		D0	EPnFIFOData[0]		X		
EPnRdRemain_H (EPn FIFO read remain high)	1000084 (B)	D7–4	–	–	–	–	0 when being read.
		D3	EPnRdRemain[11]	Endpoint n FIFO read remain high	0	R	
		D2	EPnRdRemain[10]		0		
		D1	EPnRdRemain[9]		0		
		D0	EPnRdRemain[8]		0		
EPnRdRemain_L (EPn FIFO read remain low)	1000085 (B)	D7	EPnRdRemain[7]	Endpoint n FIFO read remain low	0	R	
		D6	EPnRdRemain[6]		0		
		D5	EPnRdRemain[5]		0		
		D4	EPnRdRemain[4]		0		
		D3	EPnRdRemain[3]		0		
		D2	EPnRdRemain[2]		0		
		D1	EPnRdRemain[1]		0		
		D0	EPnRdRemain[0]		0		
EPnWrRemain_H (EPn FIFO write remain high)	1000086 (B)	D7–4	–	–	–	–	0 when being read.
		D3	EPnWrRemain[11]	Endpoint n FIFO write remain high	0	R	
		D2	EPnWrRemain[10]		0		
		D1	EPnWrRemain[9]		0		
		D0	EPnWrRemain[8]		0		
EPnWrRemain_L (EPn FIFO write remain low)	1000087 (B)	D7	EPnWrRemain[7]	Endpoint n FIFO write remain low	0	R	
		D6	EPnWrRemain[6]		0		
		D5	EPnWrRemain[5]		0		
		D4	EPnWrRemain[4]		0		
		D3	EPnWrRemain[3]		0		
		D2	EPnWrRemain[2]		0		
		D1	EPnWrRemain[1]		0		
		D0	EPnWrRemain[0]		0		
DescAdrs_H (Descriptor address high)	1000088 (B)	D7–4	–	–	–	–	0 when being read.
		D3	DescAdrs[11]	Descriptor address	0	R/W	
		D2	DescAdrs[10]		0		
		D1	DescAdrs[9]		0		
		D0	DescAdrs[8]		0		
DescAdr_L (Descriptor address low)	1000089 (B)	D7	DescAdrs[7]	Descriptor address	0	R/W	
		D6	DescAdrs[6]		0		
		D5	DescAdrs[5]		0		
		D4	DescAdrs[4]		0		
		D3	DescAdrs[3]		0		
		D2	DescAdrs[2]		0		
		D1	DescAdrs[1]		0		
		D0	DescAdrs[0]		0		
DescSize_H (Descriptor size high)	100008A (B)	D7–2	–	–	–	–	0 when being read.
		D1	DescSize[9]	Descriptor size	0	R/W	
		D0	DescSize[8]		0		
DescSize_L (Descriptor size low)	100008B (B)	D7	DescSize[7]	Descriptor size	0	R/W	
		D6	DescSize[6]		0		
		D5	DescSize[5]		0		
		D4	DescSize[4]		0		
		D3	DescSize[3]		0		
		D2	DescSize[2]		0		
		D1	DescSize[1]		0		
		D0	DescSize[0]		0		

Register name	Address	Bit	Name	Setting				Init.	R/W	Remarks
DescDoor (Descriptor door)	100008F (B)	D7	DescMode[7]	Descriptor door				0	R/W	
		D6	DescMode[6]					0		
		D5	DescMode[5]					0		
		D4	DescMode[4]					0		
		D3	DescMode[3]					0		
		D2	DescMode[2]					0		
		D1	DescMode[1]					0		
		D0	DescMode[0]					0		
DMA_FIFO_Control (DMA FIFO control)	1000090 (B)	D7	FIFO_Running	1	FIFO is running	0	FIFO is not running	0	R	
		D6	AutoEnShort	1	Auto enable short packet	0	Do nothing	0	R/W	
		D5–0	–	–				–	–	0 when being read.
DMA_Join (DMA join FIFO)	1000091 (B)	D7–4	–	–				–	–	0 when being read.
		D3	JoinEPdDMA	1	Join EPd to DMA	0	Do nothing	0	R/W	
		D2	JoinEPcDMA	1	Join EPc to DMA	0	Do nothing	0	R/W	
		D1	JoinEPbDMA	1	Join EPb to DMA	0	Do nothing	0	R/W	
		D0	JoinEPaDMA	1	Join EPa to DMA	0	Do nothing	0	R/W	
DMA_Control (DMA control)	1000092 (B)	D7	DMA_Running	1	DMA is running	0	DMA is not running	0	R	
		D6	PDREQ	PDREQ (#DMAREQ3) signal logic				0	R	
		D5	PDACK	PDACK (#CE4) signal logic				0	R	
		D4	–	–				–	–	0 when being read.
		D3	CounterClr	1	Clear DMA counter	0	Do nothing	0	W	
		D2	–	–				–	–	
		D1	DMA_Stop	1	Finish DMA	0	Do nothing	0	W	
		D0	DMA_Go	1	Start DMA	0	Do nothing	0	W	
DMA_Config_0 (DMA configuration 0)	1000094 (B)	D7	ActivePort	1	Activate DMA port	0	Disactivate DMA port	0	R/W	
		D6–4	–	–				–	–	0 when being read.
		D3	PDREQ_Level	1	Active-low	0	Active-high	0	R/W	
		D2	PDACK_Level	1	Active-low	0	Active-high	0	R/W	
		D1	PDRDWR_Level	1	Active-low	0	Active-high	0	R/W	
		D0	–	–				–	–	0 when being read.
DMA_Config_1 (DMA configuration 1)	1000095 (B)	D7	RcvLimitMode	1	Recive limit mode	0	Normal	0	R/W	
		D6–4	–	–				–	–	0 when being read.
		D3	SingleWord	1	Single word	0	Multi word	0	R/W	
		D2–1	–	–				–	–	0 when being read.
		D0	CountMode	1	Count-down mode	0	Free-run mode	0	R/W	
DMA_Latency (DMA latency)	1000097 (B)	D7–4	–	–				–	–	0 when being read.
		D3	DMA_Latency[3]	Latency				0	R/W	
		D2	DMA_Latency[2]					0		
		D1	DMA_Latency[1]					0		
		D0	DMA_Latency[0]					0		
DMA_Remain_H (DMA FIFO remain high)	1000098 (B)	D7–4	–	–				–	–	0 when being read.
		D3	DMA_Remain[11]	DMA FIFO remain high				0	R	
		D2	DMA_Remain[10]					0		
		D1	DMA_Remain[9]					0		
		D0	DMA_Remain[8]					0		
DMA_Remain_L (DMA FIFO remain low)	1000099 (B)	D7	DMA_Remain[7]	DMA FIFO remain low				0	R	
		D6	DMA_Remain[6]					0		
		D5	DMA_Remain[5]					0		
		D4	DMA_Remain[4]					0		
		D3	DMA_Remain[3]					0		
		D2	DMA_Remain[2]					0		
		D1	DMA_Remain[1]					0		
		D0	DMA_Remain[0]					0		
DMA_Count_HH (DMA transfer byte counter high/high)	100009C (B)	D7	DMA_Count[31]	DMA transfer byte counter				0	R/W	
		D6	DMA_Count[30]					0		
		D5	DMA_Count[29]					0		
		D4	DMA_Count[28]					0		
		D3	DMA_Count[27]					0		
		D2	DMA_Count[26]					0		
		D1	DMA_Count[25]					0		
		D0	DMA_Count[24]					0		
DMA_Count_HL (DMA transfer byte counter high/low)	100009D (B)	D7	DMA_Count[23]	DMA transfer byte counter				0	R/W	
		D6	DMA_Count[22]					0		
		D5	DMA_Count[21]					0		
		D4	DMA_Count[20]					0		
		D3	DMA_Count[19]					0		
		D2	DMA_Count[18]					0		
		D1	DMA_Count[17]					0		
		D0	DMA_Count[16]					0		

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Count_LH	**100009E**	D7	**DMA_Count[15]**	DMA transfer byte counter	0	R/W	
(DMA transfer	**(B)**	D6	**DMA_Count[14]**		0		
byte counter		D5	**DMA_Count[13]**		0		
low/high)		D4	**DMA_Count[12]**		0		
		D3	**DMA_Count[11]**		0		
		D2	**DMA_Count[10]**		0		
		D1	**DMA_Count[9]**		0		
		D0	**DMA_Count[8]**		0		
DMA_Count_LL	**100009F**	D7	**DMA_Count[7]**	DMA transfer byte counter	0	R/W	
(DMA transfer	**(B)**	D6	**DMA_Count[6]**		0		
byte counter		D5	**DMA_Count[5]**		0		
low/low)		D4	**DMA_Count[4]**		0		
		D3	**DMA_Count[3]**		0		
		D2	**DMA_Count[2]**		0		
		D1	**DMA_Count[1]**		0		
		D0	**DMA_Count[0]**		0		

THIS PAGE IS BLANK.

B REFERENCE: BUS CONDITION SETUP LIST

	Bus condition	48 MHz @x1 speed	48 MHz @x2 speed				32 MHz @x1 speed				24 MHz @x1 speed			
Bus Arbiter function	–	**Off**	**Off**	**Off**	**On**	**On**	**Off**	**Off**	**On**	**On**	**Off**	**Off**	**On**	**On**
USB function	–	**Off**	**Off**	**On**	**Off**	**On**	**Off**	**On**	**Off**	**On**	**Off**	**On**	**Off**	**On**
IVRAM (Internal Area 6)	Wait-cycle	NA	0	0	0	0	1	1	1	1	1	1	1	1
	Output-disable		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Read-hold		0	1	0	1	0	1	0	1	0	1	0	1
SDRAM initialization (Internal Area 6)	Wait-cycle	NA	0	0	0	0	2	2	2	2	2	2	2	2
	Output-disable		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Read-hold		0	1	0	1	0	1	0	1	0	1	0	1
Area 6 IO except SDRAM initialization (Internal Area 6)	Wait-cycle	NA	0	0	0	0	0	0	0	0	0	0	0	0
	Output-disable		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Read-hold		0	1	0	1	0	1	0	1	0	1	0	1
USB IO (Internal Area 11)	Wait-cycle	NA	NA		NA		NA		NA		NA		NA	
	Output-disable													
	Read-hold			1		1		1		1		1		1
USB DMA (Internal Area 4)	Wait-cycle	NA	NA		NA		NA		NA		NA		NA	
	Output-disable													
	Read-hold			1		1		1		1		1		1
SDRAM (External)	Wait-cycle	2	0	0	0 (Note 3)	0 (Note 3)	1	1	1	1	1	1	1	1
	Output-disable	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1.5	1.5	0.5	0.5	1.5	1.5
	Read-hold	0	0	0	0	0	0	0	0	0	0	0	0	0
SRAM/ROM (External)	Wait-cycle	Any	Any	Any	1	1	Any	Any	1	1	Any	Any	1	1
	Output-disable	Any	Any	Any	0.5	0.5	Any	Any	0.5	0.5	Any	Any	0.5	0.5
	Read-hold	Any	Any	Any	0	0	Any	Any	1	1	Any	Any	1	1
NAND flash (External)	Wait-cycle	NA	2	2	2	2	NA	NA	NA	NA	2	2	2	2
	Output-disable		0.5	0.5	0.5	0.5					0.5	0.5	0.5	0.5
	Read-hold		0	0	0	0					0	0	1	1
SQ ROM (External)	Wait-cycle	NA	5	5	5	5	6	6	6	6	4	4	4	4
	Output-disable		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Read-hold		1	1	1	1	1	1	1	1	1	1	1	1

Note 1: Besides the peripheral blocks and memories, the bus arbiter and USB functions affect the minimum access time.

Note 2: There are three setup conditions for Area 6 when using the bus in x1 speed mode; (1) 2-wait cycles for SDRAM initialization, (2) 1-wait cycle for IVRAM, and (3) 0-wait cycles for other I/O devices. To achieve high-performance, set the Area 6 wait-cycle condition to 2-wait cycles before initializing SDRAM and change it to 1-wait cycle after SDRAM initialization has completed.

Note 3: In this condition, the SDRAMC must be operated in low performance mode (SDHP [D3/0x3A0210] = "0").

THIS PAGE IS BLANK.

EPSON International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

- HEADQUARTERS -
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone: +1-800-228-3964 Fax: +1-408-922-0238

- SALES OFFICES -

West
1960 E.Grand Avenue Flr 2
El Segundo, CA 90245, U.S.A.
Phone: +1-800-249-7730 Fax: +1-310-955-5400

Central
101 Virginia Street, Suite 290
Crystal Lake, IL 60014, U.S.A.
Phone: +1-800-853-3588 Fax: +1-815-455-7633

Northeast
301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 Fax: +1-781-246-5443

Southeast
3010 Royal Blvd. South, Suite 170
Alpharetta, GA 30005, U.S.A.
Phone: +1-877-332-0020 Fax: +1-770-777-2637

EUROPE

EPSON EUROPE ELECTRONICS GmbH

- HEADQUARTERS -
Riesstrasse 15
80992 Munich, GERMANY
Phone: +49-89-14005-0 Fax: +49-89-14005-110

DÜSSELDORF BRANCH OFFICE
Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-2171-5045-0 Fax: +49-2171-5045-10

FRENCH BRANCH OFFICE
1 Avenue de l' Atlantique, LP 915 Les Conquerants
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE
Phone: +33-1-64862350 Fax: +33-1-64862355

BARCELONA BRANCH OFFICE
Barcelona Design Center
Edificio Testa, C/Alcalde Barnils 64-68, Modulo C 2a planta
E-08190 Sant Cugat del Vallès, SPAIN
Phone: +34-93-544-2490 Fax: +34-93-544-2491

UK & IRELAND BRANCH OFFICE
8 The Square, Stockley Park, Uxbridge
Middx UB11 1FW, UNITED KINGDOM
Phone: +44-1295-750-216/+44-1342-824451
Fax: +44-89-14005 446/447

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: +86-10-6410-6655 Fax: +86-10-6410-7320

SHANGHAI BRANCH
7F, High-Tech Bldg., 900, Yishan Road
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 Fax: +86-21-5423-5512

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: +886-2-8786-6688 Fax: +886-2-8786-6677

HSINCHU OFFICE
No. 99, Jiangong Road
Hsinchu City 300
Phone: +886-3-573-9900 Fax: +886-3-573-9169

EPSON SINGAPORE PTE., LTD.
401 Commonwealth Drive, #07-01
Haw Par Technocentre, SINGAPORE 149598
Phone: +65-6586-3100 Fax: +65-6472-4291

SEIKO EPSON CORPORATION KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: +82-2-784-6027 Fax: +82-2-767-3677

GUMI OFFICE
6F, Good Morning Securities Bldg., 56 Songjeong-Dong
Gumi-City, Seoul, 730-090, KOREA
Phone: +82-54-454-6027 Fax: +82-54-454-6093

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING & SALES DIV.

ED International Sales Dept.
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 Fax: +81-42-587-5117

S1C33L05
Technical Manual

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ EPSON Electronic Devices Website

http://www.epsondevice.com

Document code: 405262000
Issue September, 2004
Printed in Japan ⓁB