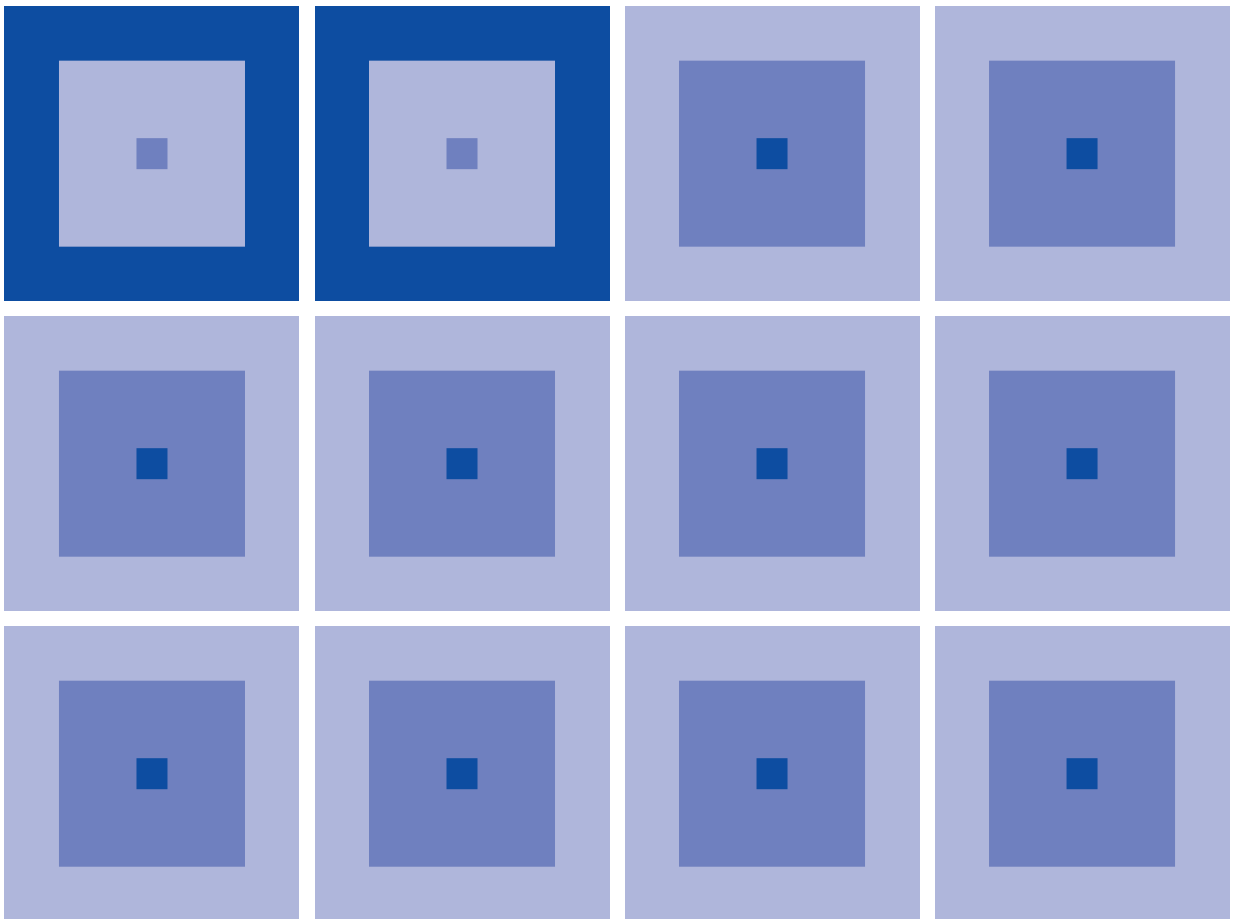


CMOS 32-BIT SINGLE CHIP MICROCOMPUTER

S1C33E08

Technical Manual



NOTICE

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

The EPSON S1C33E08 incorporates MP3 technology of which Thomson SA in France holds the patent. Manufacturers using the EPSON S1C33E08 to develop MP3 products must pay royalties to Thomson SA in order to procure the license for the MP3 technology.

Configuration of product number

Devices

S1 **C** **33209** **F** **00E1** **00**

Packing specifications

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

Specification

Package

[D: die form; F: QFP, B: BGA]

Model number

Model name

[C: microcomputer, digital products]

Product classification

[S1: semiconductor]

Development tools

S5U1 **C** **33000** **H2** **1** **00**

Packing specifications

[00: standard packing]

Version

[1: Version 1]

Tool type

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

Corresponding model number

[33L01: for S1C33L01]

Tool classification

[C: microcomputer use]

Product classification

[S5U1: development tool for semiconductor products]

S1C33E08 Technical Manual

I S1C33E08 SPECIFICATIONS	I
I.1 Overview	Overview
I.2 Block Diagram	Block
I.3 Pin Description	Pin
I.4 Power Supply	Power
I.5 CPU Core and Bus Architecture	CPU
I.6 Memory Map	Map
I.7 Electrical Characteristics	E char
I.8 Basic External Wiring Diagram	Wiring
I.9 Precautions on Mounting	Mount
II BUS MODULES	II
II.1 High-Speed DMA (HSDMA)	HSDMA
II.2 Intelligent DMA (IDMA)	IDMA
II.3 SRAM Controller (SRAMC)	SRAMC
II.4 SDRAM Controller (SDRAMC)	SDRAMC
III PERIPHERAL MODULES 1 (SYSTEM)	III
III.1 Clock Management Unit (CMU)	CMU
III.2 Interrupt Controller (ITC)	ITC
III.3 Real-Time Clock (RTC)	RTC
III.4 Misc Registers	MISC
IV PERIPHERAL MODULES 2 (TIMERS)	IV
IV.1 16-Bit Timers (T16)	T16
IV.2 Watchdog Timer (WDT)	WDT
V PERIPHERAL MODULES 3 (INTERFACE)	V
V.1 General-Purpose Serial Interface (EFSIO)	EFSIO
V.2 Serial Peripheral Interface (SPI)	SPI
V.3 Direction Control Serial Interface (DCSIO)	DCSIO
V.4 Card Interface (CARD)	CARD
V.5 I ² S Interface (I ² S)	I ² S
VI PERIPHERAL MODULES 4 (PORTS)	VI
VI.1 General-Purpose I/O Ports (GPIO)	GPIO
VI.2 Extended General-Purpose I/O Ports (EGPIO)	EGPIO
VII PERIPHERAL MODULES 5 (ANALOG)	VII
VII.1 A/D Converter (ADC)	ADC
VIII PERIPHERAL MODULES 6 (LCD)	VIII
VIII.1 LCD Controller (LCDC)	LCDC
VIII.2 IVRAM and IVRAM Arbiter	IVRAM
IX PERIPHERAL MODULES 7 (USB)	IX
IX.1 USB Function Controller (USB)	USB
X PERIPHERAL MODULES 8 (MP3)	X
X.1 MP3 Decoder (MP3)	MP3
APPENDIX	AP
A I/O Map	I/Omap
B Differences Between C33 PE Core and Other C33 Core	C33PE
C Development Tools	DEV
D Boot	Boot
E Summary of Precautions	Notes
F Supplementary Description for Clock Control	Clock

Application

This manual describes the hardware functions and control registers of the S1C33E08 or Seiko Epson's RISC-type 32-bit microcomputer, and precautions to observe when designing the application system for the microcomputer. Since this manual is written for those who design applications and circuits, knowledge of embedded-type microcomputers and the functionality and control of general peripheral circuits is required to understand the contents of this manual.

Organization of the Manual

I. S1C33E08 Specifications

This chapter outlines the S1C33E08 and describes the pin functions, and electrical characteristics. Also noise protection and other precautions to be taken when mounting the chip on the circuit board are included.

II. Bus Module

This chapter describes the modules to control DMA and the bus.

III–X Peripheral Modules

These chapters describe each peripheral module embedded in the S1C33E08.

Appendix

Provides a list of control registers built into the S1C33E08 and other additional information.

Notational Conventions for Control Bits and Addresses

This manual describes some control bits as follows:

Example: 16-bit timer RUN/STOP control bits

PRUN x (D0/0x300786 + 8 x)

' x ' in this example represents a timer number (0 to 5). Timer 0 to timer 5 have control bits for each timer that have the same functions as other timers. This manual uses ' x ' to describe two or more control bits (or addresses) in a bit name (or an expression). Therefore, ' x ' should be substituted with 0 to 5 in this example to obtain the actual bit names and addresses.

Timer 0: PRUN0 (D0/0x300786)	* 0x300786 + 8 × 0 = 0x300786
Timer 1: PRUN1 (D0/0x30078E)	* 0x300786 + 8 × 1 = 0x30078E
Timer 2: PRUN2 (D0/0x300796)	* 0x300786 + 8 × 2 = 0x300796
Timer 3: PRUN3 (D0/0x30079E)	* 0x300786 + 8 × 3 = 0x30079E
Timer 4: PRUN4 (D0/0x3007A6)	* 0x300786 + 8 × 4 = 0x3007A6
Timer 5: PRUN5 (D0/0x3007AE)	* 0x300786 + 8 × 5 = 0x3007AE

' x ' is used for not only timer numbers, but also memory block numbers, A/D converter channel numbers and others.

CONTENTS

I S1C33E08 SPECIFICATIONS

I.1 Overview	I-1-1
I.2 Block Diagram	I-2-1
I.3 Pin Description	I-3-1
I.3.1 Pin Arrangement	I-3-1
I.3.1.1 QFP Package Pin Arrangement (S1C33E08F00A)	I-3-1
I.3.1.2 PFBGA Package Pin Arrangement (S1C33E08B00A)	I-3-2
I.3.2 Pin Functions	I-3-3
I.3.3 Switching Over the Multiplexed Pin Functions	I-3-9
I.3.3.1 Pin Function Select Bits.....	I-3-9
I.3.3.2 List of Port Function Select Registers.....	I-3-12
0x3003A0: P00–P03 Port Function Select Register (pP0_03_CFP).....	I-3-13
0x3003A1: P04–P07 Port Function Select Register (pP0_47_CFP).....	I-3-14
0x3003A2: P10–P13 Port Function Select Register (pP1_03_CFP).....	I-3-15
0x3003A3: P14–P17 Port Function Select Register (pP1_47_CFP).....	I-3-16
0x3003A4: P20–P23 Port Function Select Register (pP2_03_CFP).....	I-3-17
0x3003A5: P24–P27 Port Function Select Register (pP2_47_CFP).....	I-3-18
0x3003A6: P30–P33 Port Function Select Register (pP3_03_CFP).....	I-3-19
0x3003A7: P34–P36 Port Function Select Register (pP3_46_CFP).....	I-3-20
0x3003A8: P40–P43 Port Function Select Register (pP4_03_CFP).....	I-3-21
0x3003A9: P44–P47 Port Function Select Register (pP4_47_CFP).....	I-3-22
0x3003AA: P50–P53 Port Function Select Register (pP5_03_CFP).....	I-3-23
0x3003AB: P54–P57 Port Function Select Register (pP5_47_CFP).....	I-3-24
0x3003AC: P60–P63 Port Function Select Register (pP6_03_CFP).....	I-3-25
0x3003AD: P64–P67 Port Function Select Register (pP6_47_CFP).....	I-3-26
0x3003AE: P70–P73 Port Function Select Register (pP7_03_CFP).....	I-3-27
0x3003AF: P74 Port Function Select Register (pP7_4_CFP).....	I-3-28
0x3003B0: P80–P83 Port Function Select Register (pP8_03_CFP).....	I-3-29
0x3003B1: P84–P85 Port Function Select Register (pP8_45_CFP).....	I-3-30
0x3003B2: P90–P93 Port Function Select Register (pP9_03_CFP).....	I-3-31
0x3003B3: P94–P97 Port Function Select Register (pP9_47_CFP).....	I-3-32
0x300C20: PA0–PA3 Port Function Select Register (pPA_03_CFP).....	I-3-33
0x300C21: PA4 Port Function Select Register (pPA_4_CFP).....	I-3-34
0x300C22: PB0–PB3 Port Function Select Register (pPB_03_CFP).....	I-3-35
0x300C24: PC0–PC3 Port Function Select Register (pPC_03_CFP).....	I-3-36
0x300C25: PC4–PC7 Port Function Select Register (pPC_47_CFP).....	I-3-37
I.3.4 Input/Output Cells and Input/Output Characteristics.....	I-3-38
I.3.5 Package	I-3-41
I.3.5.1 QFP24-144pin Package.....	I-3-41
I.3.5.2 PFBGA-180pin Package.....	I-3-43
I.3.5.3 Thermal Resistance of the Package	I-3-45
I.3.6 Pad Layout	I-3-46
I.4 Power Supply	I-4-1
I.4.1 Power Supply Pins	I-4-1
I.4.2 Operating Voltage (V_{DD} , V_{SS}).....	I-4-2
I.4.3 Power Supply for PLL (PLV $_{DD}$, PLV $_{SS}$).....	I-4-2
I.4.4 Power Supply for I/O Interface (V_{DDH})	I-4-2
I.4.5 Power Supply for Analog Circuits (AV_{DD})	I-4-2
I.4.6 Precautions on Power Supply	I-4-3

I.5 CPU Core and Bus Architecture	I-5-1
I.5.1 Features of the C33 PE Core	I-5-1
I.5.2 CPU Registers	I-5-2
I.5.3 Instruction Set	I-5-3
I.5.4 Trap Table	I-5-7
I.5.5 Power-Down Mode	I-5-9
I.5.6 Debug Mode	I-5-10
I.5.7 Bus Architecture	I-5-11
I.5.7.1 32-bit High-Speed Bus	I-5-11
I.5.7.2 SAPB Bus	I-5-12
I.5.7.3 External Bus	I-5-12
I.5.8 Chip ID	I-5-13
I.6 Memory Map	I-6-1
I.6.1 Boot Address and Gate ROM	I-6-3
I.6.2 Area 0 (A0RAM)	I-6-3
I.6.3 Area 1 (Specific ROM for Firmware)	I-6-4
I.6.4 Area 2 (Debug Area)	I-6-4
I.6.5 Area 3 (IVRAM)	I-6-4
I.6.6 Area 3 (DST RAM)	I-6-4
I.6.7 Area 6 (I/O Area)	I-6-5
I.6.8 External Memory Areas	I-6-5
I.7 Electrical Characteristics	I-7-1
I.7.1 Absolute Maximum Rating	I-7-1
I.7.2 Recommended Operating Conditions	I-7-1
I.7.3 DC Characteristics	I-7-2
I.7.4 Current Consumption	I-7-3
I.7.5 A/D Converter Characteristics	I-7-5
I.7.6 Oscillation Characteristics	I-7-6
I.7.7 PLL Characteristics	I-7-7
I.7.8 AC Characteristics	I-7-8
I.7.8.1 Symbol Description	I-7-8
I.7.8.2 AC Characteristics Measurement Condition	I-7-8
I.7.8.3 SRAMC AC Characteristic Tables	I-7-9
I.7.8.4 SRAMC AC Characteristic Timing Charts	I-7-10
I.7.8.5 SDRAM Interface AC Characteristics	I-7-11
I.7.8.6 LCDC AC Characteristics	I-7-14
I.7.9 USB DC and AC Characteristics	I-7-27
I.8 Basic External Wiring Diagram	I-8-1
I.9 Precautions on Mounting	I-9-1

II BUS MODULES

II.1 High-Speed DMA (HSDMA)	II-1-1
II.1.1 Functional Outline of HSDMA	II-1-1
II.1.2 I/O Pins of HSDMA	II-1-5
II.1.3 Programming Control Information	II-1-6
II.1.3.1 Standard Mode and Advanced Mode	II-1-6
II.1.3.2 Sequential Access Time for IDMA and HSDMA	II-1-7
II.1.3.3 Setting the Registers in Dual-Address Mode	II-1-8
II.1.3.4 Setting the Registers in Single-Address Mode	II-1-11
II.1.4 Enabling/Disabling DMA Transfer	II-1-14
II.1.5 Trigger Source	II-1-15
II.1.6 Operation of HSDMA	II-1-16
II.1.6.1 Operation in Dual-Address Mode	II-1-16
II.1.6.2 Operation in Single-Address Mode	II-1-20
II.1.7 Interrupt Function of HSDMA	II-1-24
II.1.8 HSDMA Operating Clock.....	II-1-25
II.1.9 Details of Control Registers	II-1-26
0x301120–0x301150: HSDMA Ch.x Transfer Counter Registers (pHSx_CNT)	II-1-28
0x301122–0x301152: HSDMA Ch.x Control Registers.....	II-1-29
0x301124–0x301154: HSDMA Ch.x Low-Order Source Address Setup Registers (pHSx_SADR)	II-1-30
0x301126–0x301156: HSDMA Ch.x High-Order Source Address Setup Registers.....	II-1-31
0x301128–0x301158: HSDMA Ch.x Low-Order Destination Address Setup Registers (pHSx_DADR)	II-1-33
0x30112A–0x30115A: HSDMA Ch.x High-Order Destination Address Setup Registers	II-1-34
0x30112C–0x30115C: HSDMA Ch.x Enable Registers (pHSx_EN)	II-1-36
0x30112E–0x30115E: HSDMA Ch.x Trigger Flag Registers (pHSx_TF)	II-1-37
0x301162–0x301192: HSDMA Ch.x Control Registers (pHSx_ADVMODE) for ADV mode....	II-1-38
0x301164–0x301196: HSDMA Ch.x Source Address Setup Registers (pHSx_AD_SADR) for ADV mode	II-1-40
0x301168–0x30119A: HSDMA Ch.x Destination Address Setup Registers (pHSx_ADV_DADR) for ADV mode	II-1-42
0x30119C: HSDMA STD/ADV Mode Select Register (pHS_CNTLMODE)	II-1-44
0x30119E: DMA Sequential Access Time Register (pHS_ACCTIME)	II-1-45
II.1.10 Precautions	II-1-46
II.2 Intelligent DMA (IDMA)	II-2-1
II.2.1 Functional Outline of IDMA	II-2-1
II.2.2 Programming Control Information	II-2-3
II.2.2.1 Setting the Base Address	II-2-3
II.2.2.2 Control Information	II-2-3
II.2.3 IDMA Invocation	II-2-8
II.2.4 Operation of IDMA	II-2-11
II.2.4.1 Single Transfer Mode	II-2-11
II.2.4.2 Successive Transfer Mode	II-2-12
II.2.4.3 Block Transfer Mode	II-2-13
II.2.4.4 Cause-of-Interrupt Processing by Trigger Type	II-2-14
II.2.5 Linking.....	II-2-15
II.2.6 Interrupt Function of Intelligent DMA.....	II-2-16
II.2.7 Details of Control Registers	II-2-17
0x301100: IDMA Base Address Register 0 (pIDMABASE)	II-2-18
0x301102: IDMA Base Address Register 1	II-2-18

0x301104: IDMA Start Register (pIDMA_START)	II-2-19
0x301105: IDMA Enable Register (pIDMA_EN)	II-2-20
II.2.8 Precautions	II-2-21
II.3 SRAM Controller (SRAMC)	II-3-1
II.3.1 Overview of the SRAMC	II-3-1
II.3.2 SRAMC Pins	II-3-2
II.3.3 External Memory Area (Areas 4, 5, 7 to 22)	II-3-3
II.3.3.1 Chip Enable Signals.....	II-3-4
II.3.3.2 Area Condition Settings	II-3-4
II.3.4 Connection of External Devices and Bus Operation	II-3-7
II.3.4.1 Connecting External Devices	II-3-7
II.3.4.2 Data Configuration in Memory	II-3-7
II.3.4.3 External Bus Operation.....	II-3-8
II.3.5 SRAMC Operating Clock and Bus Clock	II-3-9
II.3.5.1 Operating Clock of the SRAMC	II-3-9
II.3.5.2 Generation of the Bus Clock	II-3-10
II.3.5.3 External Output of the Bus Clock.....	II-3-10
II.3.6 Bus Access Timing Chart.....	II-3-11
II.3.6.1 SRAM Read/Write Timings with No External #WAIT	II-3-11
II.3.6.2 SRAM Read/Write Timings with External #WAIT.....	II-3-13
II.3.6.3 SRAM Read/Write Timings with #CE4/#CE11 Setup Time	II-3-14
II.3.6.4 SRAM Read Timings with #CE9 Output Disable Time.....	II-3-15
II.3.7 Control Register Details	II-3-16
0x301500: BCLK and Setup Time Control Register (pSRAMC_BCLK_SETUP)	II-3-17
0x301504: Wait Control Register (pSRAMC_SWAIT)	II-3-18
0x301508: Device Size Setup Register (pSRAMC_SLV_SIZE)	II-3-20
0x30150C: Device Type Setup Register (pSRAMC_A0_BSL).....	II-3-21
0x301510: Area Location Setup Register (pSRAMC_ALS).....	II-3-22
II.3.8 Precautions	II-3-23
II.4 SDRAM Controller (SDRAMC)	II-4-1
II.4.1 SDRAM Interface	II-4-1
II.4.1.1 Overview of the SDRAM Interface	II-4-1
II.4.1.2 SDRAMC Pins	II-4-2
II.4.1.3 Configuration of SDRAM	II-4-3
II.4.1.4 SDRAMC Operating Clock and SDRAM Clock.....	II-4-8
II.4.1.5 Control and Operation of SDRAM Interface.....	II-4-10
II.4.2 Instruction/Data Queue Buffers.....	II-4-18
II.4.2.1 Overview	II-4-18
II.4.2.2 IQB (Instruction Queue Buffer)	II-4-18
II.4.2.3 DQB (Data Queue Buffer).....	II-4-19
II.4.2.4 Operations using IQB/DQB.....	II-4-20
II.4.3 Bus Arbiter	II-4-21
II.4.3.1 Overview	II-4-21
II.4.3.2 Controlling the Bus Arbiter	II-4-21
II.4.4 Control Register Details	II-4-22
0x301600: SDRAM Initial Register (pSDRAMC_INI)	II-4-23
0x301604: SDRAM Configuration Register (pSDRAMC_CTL)	II-4-25
0x301608: SDRAM Refresh Register (pSDRAMC_REF).....	II-4-27
0x301610: SDRAM Application Configuration Register (pSDRAMC_APP)	II-4-29
II.4.5 Precautions	II-4-31

III PERIPHERAL MODULES 1 (SYSTEM)

III.1 Clock Management Unit (CMU)	III-1-1
III.1.1 Overview of the CMU	III-1-1
III.1.2 Reset Input and Initial Reset	III-1-2
III.1.2.1 Initial Reset Pin	III-1-2
III.1.2.2 Initial Reset Status	III-1-2
III.1.2.3 Power-on Reset	III-1-3
III.1.2.4 Precautions to be Taken during Initial Reset	III-1-4
III.1.3 NMI Input	III-1-5
III.1.4 Selecting the System Clock Source	III-1-6
III.1.5 Controlling the Oscillator Circuit	III-1-7
III.1.5.1 OSC3 Oscillator Circuit	III-1-7
III.1.5.2 Setting the OSC3 Divider	III-1-8
III.1.5.3 OSC1 Oscillator Circuit	III-1-8
III.1.6 Controlling the PLL	III-1-10
III.1.6.1 On/Off Control of the PLL	III-1-10
III.1.6.2 Selecting the PLL Input Clock	III-1-10
III.1.6.3 Setting the Frequency Multiplication Rate	III-1-11
III.1.6.4 Other PLL Settings	III-1-12
III.1.6.5 Power Supply for PLL	III-1-13
III.1.7 Control of the SSCG	III-1-14
III.1.7.1 Turning the SSCG On/Off	III-1-14
III.1.7.2 Setting SS Modulation Parameters	III-1-15
III.1.8 Setting the Main System Clock (MCLK)	III-1-16
III.1.9 Controlling Clock Supply	III-1-17
III.1.9.1 MCLK Clock Supply to Each Module	III-1-17
III.1.9.2 Automatic Clock Control in HALT Mode	III-1-17
III.1.9.3 Clock Supply to the LCDC	III-1-18
III.1.9.4 Clock Supply to the SDRAMC	III-1-19
III.1.9.5 Clock Supply to the SRAMC	III-1-19
III.1.9.6 Clock Supply to the GPIO	III-1-20
III.1.9.7 Clock Supply to the EFSIO	III-1-20
III.1.9.8 Clock Supply to the USB	III-1-20
III.1.9.9 Clock Supply to the RTC	III-1-21
III.1.10 Setting the External Clock Output (CMU_CLK)	III-1-22
III.1.11 Standby Modes	III-1-23
III.1.11.1 HALT Mode	III-1-23
III.1.11.2 SLEEP Mode	III-1-23
III.1.11.3 Precautions	III-1-25
III.1.12 Clock Setup Procedure	III-1-26
III.1.12.1 Changing the Clock Source from OSC3 to PLL	III-1-26
III.1.12.2 Changing the Clock Source from PLL to OSC3, then Turning Off the PLL	III-1-27
III.1.12.3 Changing the Clock Source from OSC3 or PLL to OSC1, then Turning Off OSC3 and PLL	III-1-28
III.1.12.4 Changing the Clock Source from OSC1 to OSC3	III-1-29
III.1.12.5 Changing the Clock Source from OSC1 to PLL	III-1-30
III.1.12.6 Turning Off OSC3 during SLEEP	III-1-31
III.1.12.7 SLEEP Keeping Oscillation On (without Clock Change)	III-1-32
III.1.13 Power-Down Control	III-1-33

III.1.14	Details of Control Registers	III-1-34
	0x301B00: Gated Clock Control Register 0 (pCMU_GATEDCLK0)	III-1-35
	0x301B04: Gated Clock Control Register 1 (pCMU_GATEDCLK1)	III-1-37
	0x301B08: System Clock Control Register (pCMU_CLKCNTL).....	III-1-40
	0x301B0C: PLL Control Register (pCMU_PLL)	III-1-44
	0x301B10: SSCG Macro Control Register (pCMU_SSCG)	III-1-46
	0x301B14: Clock Option Register (pCMU_OPT).....	III-1-47
	0x301B24: Clock Control Protect Register (pCMU_PROTECT).....	III-1-49
III.1.15	Precautions	III-1-50
III.2	Interrupt Controller (ITC)	III-2-1
III.2.1	Outline of Interrupt Functions.....	III-2-1
III.2.1.1	Maskable Interrupts	III-2-1
III.2.1.2	Causes of Interrupt and Intelligent DMA.....	III-2-4
III.2.1.3	Nonmaskable Interrupt (NMI)	III-2-4
III.2.1.4	Interrupt Processing by the CPU	III-2-4
III.2.1.5	Clearing Standby Mode by Interrupts	III-2-4
III.2.2	Trap Table	III-2-6
III.2.3	ITC Operating Clock.....	III-2-7
III.2.4	Control of Maskable Interrupts	III-2-8
III.2.4.1	Structure of the Interrupt Controller	III-2-8
III.2.4.2	Processor Status Register (PSR)	III-2-8
III.2.4.3	Cause-of-Interrupt Flag and Interrupt Enable Register.....	III-2-9
III.2.4.4	Interrupt Priority Register and Interrupt Levels	III-2-10
III.2.5	IDMA Invocation	III-2-11
III.2.6	HSDMA Invocation	III-2-13
III.2.7	Details of Control Registers	III-2-14
	0x300260: Port Input 0–1 Interrupt Priority Register (pINT_PP01L)	III-2-16
	0x300261: Port Input 2–3 Interrupt Priority Register (pINT_PP23L)	III-2-17
	0x300262: Key Input Interrupt Priority Register (pINT_PK01L).....	III-2-18
	0x300263: HSDMA Ch.0–1 Interrupt Priority Register (pINT_PHSD01L).....	III-2-19
	0x300264: HSDMA Ch.2–3 Interrupt Priority Register (pINT_PHSD23L).....	III-2-20
	0x300265: IDMA Interrupt Priority Register (pINT_PDM).....	III-2-21
	0x300266: 16-bit Timer 0–1 Interrupt Priority Register (pINT_P16T01)	III-2-22
	0x300267: 16-bit Timer 2–3 Interrupt Priority Register (pINT_P16T23).....	III-2-23
	0x300268: 16-bit Timer 4–5 Interrupt Priority Register (pINT_P16T45).....	III-2-24
	0x300269: LCDC, Serial I/F Ch.0 Interrupt Priority Register (pINT_PLCDC_PSI00).....	III-2-25
	0x30026A: Serial I/F Ch.1, A/D Interrupt Priority Register (pINT_PSI01_PAD)	III-2-26
	0x30026B: RTC Interrupt Priority Register (pINT_PRTC)	III-2-27
	0x30026C: Port Input 4–5 Interrupt Priority Register (pINT_PP45L).....	III-2-28
	0x30026D: Port Input 6–7 Interrupt Priority Register (pINT_PP67L).....	III-2-29
	0x30026E: Serial I/F Ch.2, SPI Interrupt Priority Register (pINT_PSI02_PSPI)	III-2-30
	0x300270: Key Input, Port Input 0–3 Interrupt Enable Register (pINT_EK01_EP03).....	III-2-31
	0x300271: DMA Interrupt Enable Register (pINT_EDMA)	III-2-32
	0x300272: 16-bit Timer 0–1 Interrupt Enable Register (pINT_E16T01)	III-2-33
	0x300273: 16-bit Timer 2–3 Interrupt Enable Register (pINT_E16T23).....	III-2-34
	0x300274: 16-bit Timer 4–5 Interrupt Enable Register (pINT_E16T45).....	III-2-35
	0x300276: Serial I/F Ch.0–1 Interrupt Enable Register (pINT_ESIF01).....	III-2-36
	0x300277: Port Input 4–7, RTC, A/D Interrupt Enable Register (pINT_EP47_ERTC_EAD) ..	III-2-37
	0x300278: LCDC Interrupt Enable Register (pINT_ELCDC).....	III-2-38
	0x300279: Serial I/F Ch.2, SPI Interrupt Enable Register (pINT_ESIF2_ESPI).....	III-2-39
	0x300280: Key Input, Port Input 0–3 Interrupt Cause Flag Register (pINT_FK01_FP03).....	III-2-40
	0x300281: DMA Interrupt Cause Flag Register (pINT_FDMA)	III-2-42
	0x300282: 16-bit Timer 0–1 Interrupt Cause Flag Register (pINT_F16T01).....	III-2-43
	0x300283: 16-bit Timer 2–3 Interrupt Cause Flag Register (pINT_F16T23).....	III-2-44

0x300284: 16-bit Timer 4–5 Interrupt Cause Flag Register (pINT_F16T45)	III-2-45
0x300286: Serial I/F Ch.0–1 Interrupt Cause Flag Register (pINT_FSIF01).....	III-2-46
0x300287: Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (pINT_FP47_FRTC_FAD).....	III-2-47
0x300288: LCDC Interrupt Cause Flag Register (pINT_FLCDC).....	III-2-48
0x300289: Serial I/F Ch.2, SPI Interrupt Cause Flag Register (pINT_FSIF2_FSPI).....	III-2-49
0x300290: Port Input 0–3, HSDMA Ch.0–1, 16-bit Timer 0 IDMA Request Register (pIDMAREQ_RP03_RHS_R16T0)	III-2-50
0x300291: 16-bit Timer 1–4 IDMA Request Register (pIDMAREQ_R16T14)	III-2-51
0x300292: 16-bit Timer 5, Serial I/F Ch.0 IDMA Request Register (pIDMAREQ_R16T5_RSIF0).....	III-2-52
0x300293: Serial I/F Ch.1, A/D, Port Input 4–7 IDMA Request Register (pIDMAREQ_RSIF1_RAD_RP47)	III-2-53
0x300294: Port Input 0–3, HSDMA Ch.0–1, 16-bit Timer 0 IDMA Enable Register (pIDMAEN_DEP03_DEHS_DE16T0)	III-2-54
0x300295: 16-bit Timer 1–4 IDMA Enable Register (pIDMAEN_DE16T14).....	III-2-55
0x300296: 16-bit Timer 5, Serial I/F Ch.0 IDMA Enable Register (pIDMAEN_DE16T5_DESIF0).....	III-2-56
0x300297: Serial I/F Ch.1, A/D, Port Input 4–7 IDMA Enable Register (pIDMAEN_DESIF1_DEAD_DEP47).....	III-2-57
0x300298: HSDMA Ch.0–1 Trigger Set-up Register (pHSDMA_HTGR1).....	III-2-58
0x300299: HSDMA Ch.2–3 Trigger Set-up Register (pHSDMA_HTGR2).....	III-2-58
0x30029A: HSDMA Software Trigger Register (pHSDMA_HSOFTTGR).....	III-2-60
0x30029B: LCDC, Serial I/F Ch.2, SPI IDMA Request Register (pIDMAREQ_RLCDC_RSIF2_RSPI).....	III-2-61
0x30029C: LCDC, Serial I/F Ch.2, SPI IDMA Enable Register (pIDMAEN_DELCDC_DESIF2_DESPI)	III-2-62
0x30029F: Flag Set/Reset Method Select Register (pRST_RESET).....	III-2-63
0x3002A0: Port Input 8–9 Interrupt Priority Register (pINT_PP89L).....	III-2-64
0x3002A1: Port Input 10–11 Interrupt Priority Register (pINT_PP1011L).....	III-2-65
0x3002A2: Port Input 12–13 Interrupt Priority Register (pINT_PP1213L).....	III-2-66
0x3002A3: Port Input 14–15 Interrupt Priority Register (pINT_PP1415L).....	III-2-67
0x3002A4: I ² S Interrupt Priority Register (pINT_PI2S).....	III-2-68
0x3002A6: Port Input 8–15 Interrupt Enable Register (pINT_EP815)	III-2-69
0x3002A7: I ² S Interrupt Enable Register (pINT_EI2S).....	III-2-70
0x3002A9: Port Input 8–15 Interrupt Cause Flag Register (pINT_FP815).....	III-2-71
0x3002AA: I ² S Interrupt Cause Flag Register (pINT_FI2S)	III-2-72
0x3002AC: Port Input 8–15 IDMA Request Register (pIDMAREQ_RP815)	III-2-73
0x3002AD: I ² S IDMA Request Register (pIDMAREQ_RI2S)	III-2-74
0x3002AE: Port Input 8–15 IDMA Enable Register (pIDMAEN_DEP815)	III-2-75
0x3002AF: I ² S IDMA Enable Register (pIDMAEN_DEI2S)	III-2-76
0x3003C4: Port Input Interrupt Select Register 3 (pPINTSEL_SPT811)	III-2-77
III.2.8 Precautions	III-2-78
III.3 Real-Time Clock (RTC).....	III-3-1
III.3.1 Overview of the RTC	III-3-1
III.3.2 RTC Counters	III-3-2
III.3.3 Control of the RTC	III-3-5
III.3.3.1 Controlling the Operating Clock.....	III-3-5
III.3.3.2 Initial Sequence of the RTC	III-3-6
III.3.3.3 Selecting 12/24-hour Mode and Setting the Counters.....	III-3-7
III.3.3.4 Starting, Stopping, and Resetting Counters	III-3-7
III.3.3.5 Counter Hold and Busy Flag.....	III-3-8
III.3.3.6 Reading from and Writing to Counters in Operation	III-3-9
III.3.3.7 30-second Correction	III-3-9
III.3.4 RTC Interrupts.....	III-3-10

III.3.5	OSC1 Oscillator Circuit	III-3-11
III.3.5.1	Input/Output Pins of the OSC1 Oscillator Circuit	III-3-11
III.3.5.2	Structure of the OSC1 Oscillator Circuit	III-3-11
III.3.5.3	Oscillation Control	III-3-12
III.3.6	Details of Control Registers	III-3-13
	0x301900: RTC Interrupt Status Register (pRTCINTSTAT)	III-3-14
	0x301904: RTC Interrupt Mode Register (pRTCINTMODE)	III-3-15
	0x301908: RTC Control Register (pRTC_CNTL0)	III-3-16
	0x30190C: RTC Access Control Register (pRTC_CNTL1)	III-3-18
	0x301910: RTC Second Register (pRTCSEC)	III-3-19
	0x301914: RTC Minute Register (pRTCMIN)	III-3-20
	0x301918: RTC Hour Register (pRTC HOUR)	III-3-21
	0x30191C: RTC Day Register (pRTC DAY)	III-3-22
	0x301920: RTC Month Register (pRTC MONTH)	III-3-23
	0x301924: RTC Year Register (pRTC YEAR)	III-3-24
	0x301928: RTC Days of Week Register (pRTC DAY WEEK)	III-3-25
III.3.7	Precautions	III-3-26
III.4	Misc Registers	III-4-1
III.4.1	RTC and USB Wait Control Registers	III-4-1
III.4.1.1	Setting Wait Cycles for Accessing the RTC	III-4-1
III.4.1.2	Settings for the USB	III-4-1
III.4.2	Debug Port MUX Register	III-4-2
III.4.3	Boot Register	III-4-3
III.4.4	Pin Control Registers	III-4-4
III.4.4.1	Pull-up Control	III-4-4
III.4.4.2	Driving Bus Signals Low	III-4-4
III.4.5	Misc Register Operating Clock	III-4-5
III.4.6	Details of Control Registers	III-4-6
	0x300010: RTC Wait Control Register (pMISC_RTCWT)	III-4-7
	0x300012: USB Wait Control Register (pMISC_USBWT)	III-4-8
	0x300014: Debug Port MUX Register (pMISC_PMUX)	III-4-9
	0x300016: Performance Analyzer Control Register (pMISC_PAC)	III-4-10
	0x300018: Boot Register (pMISC_BOOT)	III-4-11
	0x30001A: COROM Switch Register (pMISC_COROM)	III-4-12
	0x300020: Misc Protect Register (pMISC_PROT)	III-4-13
	0x300C41: Bus Signal Low Drive Control Register (pMISC_BUSLOW)	III-4-14
	0x300C42: P0 Pull-up Control Register (pMISC_PUP0)	III-4-15
	0x300C43: P1 Pull-up Control Register (pMISC_PUP1)	III-4-16
	0x300C44: P2 Pull-up Control Register (pMISC_PUP2)	III-4-17
	0x300C45: P3 Pull-up Control Register (pMISC_PUP3)	III-4-18
	0x300C46: P4 Pull-up Control Register (pMISC_PUP4)	III-4-19
	0x300C47: P5 Pull-up Control Register (pMISC_PUP5)	III-4-20
	0x300C48: P6 Pull-up Control Register (pMISC_PUP6)	III-4-21
	0x300C49: P7 Pull-up Control Register (pMISC_PUP7)	III-4-22
	0x300C4A: P8 Pull-up Control Register (pMISC_PUP8)	III-4-23
	0x300C4B: P9 Pull-up Control Register (pMISC_PUP9)	III-4-24
	0x300C4C: PA Pull-up Control Register (pMISC_PUPA)	III-4-25
	0x300C4D: PB Pull-up Control Register (pMISC_PUPB)	III-4-26
III.4.7	Precautions	III-4-27

IV PERIPHERAL MODULES 2 (TIMERS)

IV.1 16-Bit Timers (T16)	IV-1-1
IV.1.1 Configuration of 16-bit Timer.....	IV-1-1
IV.1.2 I/O Pins of 16-bit Timers.....	IV-1-3
IV.1.3 Uses of 16-bit Timers	IV-1-4
IV.1.4 16-bit Timer Operating Clock.....	IV-1-5
IV.1.5 Control and Operation of 16-bit Timer.....	IV-1-6
IV.1.6 Controlling Clock Output	IV-1-10
IV.1.7 16-bit Timer Interrupts and DMA.....	IV-1-13
IV.1.8 Details of Control Registers	IV-1-16
0x300780–0x3007A8: 16-bit Timer x Comparison Data A Setup Registers (pT16_CRxA)....	IV-1-17
0x300782–0x3007AA: 16-bit Timer x Comparison Data B Setup Registers (pT16_CRxB) ...	IV-1-18
0x300784–0x3007AC: 16-bit Timer x Counter Data Registers (pT16_TCx).....	IV-1-19
0x300786–0x3007AE: 16-bit Timer x Control Registers (pT16_CTLx)	IV-1-20
0x3007D0–0x3007D4: DA16 Ch.x Registers (pDA16_CRxA).....	IV-1-22
0x3007DC: Count Pause Register (pT16_CNT_PAUSE)	IV-1-23
0x3007DE: 16-bit Timer STD/ADV Mode Select Register (pT16_ADVMODE).....	IV-1-24
0x3007E0–0x3007EA: 16-bit Timer x Clock Control Registers (pT16_CLKCTL_x)	IV-1-25
IV.1.9 Precautions	IV-1-26
IV.2 Watchdog Timer (WDT)	IV-2-1
IV.2.1 Configuration of the Watchdog Timer	IV-2-1
IV.2.2 Input/Output Pins of the Watchdog Timer	IV-2-2
IV.2.3 Operating Clock of the Watchdog Timer	IV-2-3
IV.2.4 Control of the Watchdog Timer.....	IV-2-4
IV.2.4.1 Setting Up the Watchdog Timer	IV-2-4
IV.2.4.2 Starting/Stopping the Watchdog Timer	IV-2-5
IV.2.4.3 Resetting the Watchdog Timer	IV-2-5
IV.2.4.4 Operation in Standby Mode	IV-2-5
IV.2.4.5 Clock Output of the Watchdog Timer	IV-2-6
IV.2.4.6 External NMI Output	IV-2-6
IV.2.5 Details of Control Registers	IV-2-7
0x300660: Watchdog Timer Write-Protect Register (pWD_WP)	IV-2-8
0x300662: Watchdog Timer Enable Register (pWD_EN)	IV-2-9
0x300664: Watchdog Timer Comparison Data Setup Register 0 (pWD_COMP_LOW).....	IV-2-11
0x300666: Watchdog Timer Comparison Data Setup Register 1 (pWD_COMP_HIGH).....	IV-2-11
0x300668: Watchdog Timer Count Register 0 (pWD_CNT_LOW)	IV-2-12
0x30066A: Watchdog Timer Count Register 1 (pWD_CNT_HIGH).....	IV-2-12
0x30066C: Watchdog Timer Control Register (pWD_CNTL).....	IV-2-13
IV.2.6 Precautions	IV-2-14

V PERIPHERAL MODULES 3 (INTERFACE)

V.1 General-Purpose Serial Interface (EFSIO)	V-1-1
V.1.1 Configuration of Serial Interfaces	V-1-1
V.1.1.1 Features of Serial Interfaces	V-1-1
V.1.1.2 I/O Pins of Serial Interface	V-1-2
V.1.1.3 Setting Interface Mode and Transfer Mode	V-1-3
V.1.1.4 Serial Interface Operating Clock	V-1-4
V.1.1.5 Standard Mode and Advanced Mode	V-1-5
V.1.2 Baud-Rate Timer (Setting Baud Rate)	V-1-6
V.1.3 Clock-Synchronized Interface	V-1-8
V.1.3.1 Outline of Clock-Synchronized Interface	V-1-8
V.1.3.2 Setting Clock-Synchronized Interface	V-1-9
V.1.3.3 Control and Operation of Clock-Synchronized Transfer	V-1-10
V.1.4 Asynchronous Interface.....	V-1-17
V.1.4.1 Outline of Asynchronous Interface	V-1-17
V.1.4.2 Setting Asynchronous Interface	V-1-18
V.1.4.3 Control and Operation of Asynchronous Transfer	V-1-20
V.1.5 IrDA Interface	V-1-25
V.1.5.1 Outline of IrDA Interface.....	V-1-25
V.1.5.2 Setting IrDA Interface.....	V-1-25
V.1.5.3 Control and Operation of IrDA Interface.....	V-1-27
V.1.6 ISO7816 Interface (Ch.1)	V-1-28
V.1.6.1 Outline of ISO7816 Interface	V-1-28
V.1.6.2 Setting ISO7816 Interface.....	V-1-29
V.1.6.3 Control and Operation of ISO7816 Mode	V-1-32
V.1.7 Serial Interface Interrupts and DMA.....	V-1-37
V.1.8 Details of Control Registers	V-1-40
0x300B00–0x300B20: Serial I/F Ch.x Transmit Data Registers (pEFSIFx_TXD).....	V-1-41
0x300B01–0x300B21: Serial I/F Ch.x Receive Data Registers (pEFSIFx_RXD).....	V-1-42
0x300B02–0x300B22: Serial I/F Ch.x Status Registers (pEFSIFx_STATUS)	V-1-43
0x300B03–0x300B23: Serial I/F Ch.x Control Registers (pEFSIFx_CTL)	V-1-45
0x300B04–0x300B24: Serial I/F Ch.x IrDA Registers (pEFSIFx_IRDA)	V-1-47
0x300B05–0x300B25: Serial I/F Ch.x Baud-rate Timer Control Registers (pEFSIFx_BRTRUN)	V-1-49
0x300B06–0x300B26: Serial I/F Ch.x Baud-rate Timer Reload Data Registers (LSB) (pEFSIFx_BRTRDL).....	V-1-50
0x300B07–0x300B27: Serial I/F Ch.x Baud-rate Timer Reload Data Registers (MSB) (pEFSIFx_BRTRDM).....	V-1-50
0x300B08–0x300B28: Serial I/F Ch.x Baud-rate Timer Count Data Registers (LSB) (pEFSIFx_BRTCDL).....	V-1-51
0x300B09–0x300B29: Serial I/F Ch.x Baud-rate Timer Count Data Registers (MSB) (pEFSIFx_BRTCDM).....	V-1-51
0x300B1A: Serial I/F Ch.1 ISO7816 Mode Control Register (pEFSIF1_7816CTL)	V-1-52
0x300B1B: Serial I/F Ch.1 ISO7816 Mode Status Register (pEFSIF1_7816STA)	V-1-54
0x300B1C: Serial I/F Ch.1 ISO7816 Mode FI/DI Ratio Register (LSB) (pEFSIF1_FIDIL)	V-1-55
0x300B1D: Serial I/F Ch.1 ISO7816 Mode FI/DI Ratio Register (MSB) (pEFSIF1_FIDIM)	V-1-55
0x300B1E: Serial I/F Ch.1 Transmit Time Guard Register (pEFSIF1_TTGR)	V-1-57
0x300B1F: Serial I/F Ch.1 ISO7816 Mode Output Clock Setup Register (pEFSIF1_CLKNUM).....	V-1-58
0x300B4F: Serial I/F STD/ADV Mode Select Register (pEFSIF_ADV)	V-1-59
V.1.9 Precautions	V-1-60

V.2 Serial Peripheral Interface (SPI)	V-2-1
V.2.1 Outline of SPI Module	V-2-1
V.2.2 I/O Pins of SPI Module	V-2-2
V.2.3 SPI Operating Clock.....	V-2-3
V.2.4 Setting SPI Module	V-2-4
V.2.5 Control of Data Transfer	V-2-6
V.2.6 SPI Interrupts and DMA	V-2-10
V.2.7 Details of Control Registers	V-2-13
0x301700: SPI Receive Data Register (pSPI_RXD)	V-2-14
0x301704: SPI Transmit Data Register (pSPI_TXD)	V-2-15
0x301708: SPI Control Register 1 (pSPI_CTL1)	V-2-16
0x30170C: SPI Control Register 2 (pSPI_CTL2)	V-2-18
0x301710: SPI Wait Register (pSPI_WAIT).....	V-2-19
0x301714: SPI Status Register (pSPI_STAT).....	V-2-20
0x301718: SPI Interrupt Control Register (pSPI_INT).....	V-2-21
0x30171C: SPI Receive Data Mask Register (pSPI_RXMK)	V-2-23
V.2.8 Precautions	V-2-24
V.3 Direction Control Serial Interface (DCSIO)	V-3-1
V.3.1 Outline of DCSIO	V-3-1
V.3.2 I/O Pins of DCSIO	V-3-2
V.3.3 DCSIO Operating Clock	V-3-3
V.3.4 Setting DCSIO Module	V-3-4
V.3.5 Control of Data Transfer	V-3-7
V.3.6 DCSIO Interrupts and DMA.....	V-3-11
V.3.7 Details of Control Registers	V-3-13
0x301800: DCSIO Control Register (pDCSIO_CTL).....	V-3-14
0x301804: DCSIO Data Load Register (pDCSIO_LOAD)	V-3-15
0x301808: DCSIO Receive Data Register (pDCSIO_RCV)	V-3-16
0x301814: DCSIO Interrupt Control Register (pDCSIO_INT)	V-3-17
0x301818: DCSIO Status Register (pDCSIO_STAT).....	V-3-18
0x30181C: DCSIO Port Direction Control Register (pDCSIO_DIR)	V-3-19
V.3.8 Precautions	V-3-20
V.4 Card Interface (CARD)	V-4-1
V.4.1 Outline of the Card Interface	V-4-1
V.4.2 Card Interface Pins.....	V-4-2
V.4.3 Card Area	V-4-3
V.4.3.1 Selecting the Area	V-4-3
V.4.3.2 Setting Area Access Conditions.....	V-4-3
V.4.4 Card Interface Control Signals	V-4-4
V.4.4.1 SmartMedia Interface	V-4-4
V.4.4.2 CompactFlash Interface.....	V-4-5
V.4.4.3 PC Card Interface	V-4-5
V.4.5 Card Interface Operating Clock.....	V-4-7
V.4.6 ECC Generator	V-4-8
V.4.7 Details of Control Registers	V-4-11
0x300300: Card I/F Area Configuration Register (pCARDSETUP).....	V-4-12
0x300302: Card I/F Output Port Configuration Register (pCARDFUNCSEL05)	V-4-13
0x300310: ECC Trigger Area Select Register (pECCTRIGSEL)	V-4-14
0x300311: ECC Reset/Ready Register (pECCRSTRDY).....	V-4-15
0x300312: ECC Enable Register (pECCENA)	V-4-16
0x300313: ECC Mode Register (pECCMD)	V-4-17

0x300314: Area 0 ECC Column Parity Data Register (pECC0CP)	V-4-18
0x300316: Area 0 ECC Line Parity Register 0 (pECC0LPL)	V-4-19
0x300317: Area 0 ECC Line Parity Register 1 (pECC0LPH)	V-4-19
0x300318: Area 1 ECC Column Parity Data Register (pECC1CP)	V-4-20
0x30031A: Area 1 ECC Line Parity Register 0 (pECC1LPL).....	V-4-21
0x30031B: Area 1 ECC Line Parity Register 1 (pECC1LPH)	V-4-21
V.4.8 Precautions	V-4-22
V.5 I²S Interface (I²S).....	V-5-1
V.5.1 Outline of I ² S Module	V-5-1
V.5.2 Output Pins of I ² S Module.....	V-5-2
V.5.3 I ² S Module Operating Clock	V-5-3
V.5.4 Setting I ² S Module	V-5-4
V.5.5 Data Output Control	V-5-9
V.5.6 I ² S Interrupt.....	V-5-11
V.5.7 Details of Control Registers	V-5-12
0x301C00: I ² S Control Register (pl2S_CONTROL)	V-5-13
0x301C04: I ² S MCLK Divide Ratio Register (pl2S_DV_MCLK)	V-5-16
0x301C08: I ² S Audio Clock Divide Ratio Register (pl2S_DV_LRCLK).....	V-5-17
0x301C0C: I ² S Start Register (pl2S_START).....	V-5-18
0x301C10: I ² S HSDMA Mode Select Register (pl2S_HSDMAMD).....	V-5-19
0x301C14: I ² S FIFO Status Register (pl2S_FIFO_EMPTY)	V-5-20
0x301C20: I ² S FIFO Register (pl2S_FIFO).....	V-5-21

VI PERIPHERAL MODULES 4 (PORTS)

VI.1 General-Purpose I/O Ports (GPIO)	VI-1-1
VI.1.1 Structure of I/O Port.....	VI-1-1
VI.1.2 Selecting the I/O Pin Functions	VI-1-1
VI.1.3 I/O Control Register and I/O Modes	VI-1-2
VI.1.4 Input Interrupt	VI-1-3
VI.1.4.1 Port Input Interrupt	VI-1-3
VI.1.4.2 Key Input Interrupt.....	VI-1-5
VI.1.4.3 Control Registers of the Interrupt Controller.....	VI-1-7
VI.1.5 I/O Port Operating Clock.....	VI-1-9
VI.1.6 Details of Control Registers.....	VI-1-10
0x300380–0x300392: Px Port Data Registers (pPx_PxD)	VI-1-11
0x30038E: P7 Port Data Register (pP7_P7D).....	VI-1-12
0x300381–0x300393: Px I/O Control Registers (pPx_IOCx)	VI-1-13
0x3003A0–0x3003B3: Pxx Port Function Select Registers (pPx_xx_CFP)	VI-1-14
0x3003C0: Port Input Interrupt Select Register 1 (pPINTSEL_SPT03)	VI-1-15
0x3003C1: Port Input Interrupt Select Register 2 (pPINTSEL_SPT47)	VI-1-15
0x3003C4: Port Input Interrupt Select Register 3 (pPINTSEL_SPT811)	VI-1-15
0x3003C5: Port Input Interrupt Select Register 4 (pPINTSEL_SPT1215)	VI-1-15
0x3003C2: Port Input Interrupt Polarity Select Register 1 (pPINTPOL_SPP07).....	VI-1-17
0x3003C6: Port Input Interrupt Polarity Select Register 2 (pPINTPOL_SPP815).....	VI-1-17
0x3003C3: Port Input Interrupt Edge/Level Select Register 1 (pPINTEL_SEPT07).....	VI-1-18
0x3003C7: Port Input Interrupt Edge/Level Select Register 2 (pPINTEL_SEPT815).....	VI-1-18
0x3003D0: Key Input Interrupt Select Register (pKINTSEL_SPPK01)	VI-1-19
0x3003D2: Key Input Interrupt (FPK0) Input Comparison Register (pKINTCOMP_SCPK0)...	VI-1-20
0x3003D3: Key Input Interrupt (FPK1) Input Comparison Register (pKINTCOMP_SCPK1)...	VI-1-20
0x3003D4: Key Input Interrupt (FPK0) Input Mask Register (pKINTCOMP_SMPK0).....	VI-1-21
0x3003D5: Key Input Interrupt (FPK1) Input Mask Register (pKINTCOMP_SMPK1).....	VI-1-21
VI.1.7 Precautions.....	VI-1-22
VI.2 Extended General-Purpose I/O Ports (EGPIO)	VI-2-1
VI.2.1 Structure of EGPIO Port	VI-2-1
VI.2.2 Selecting the I/O Pin Functions	VI-2-1
VI.2.3 I/O Control Register and I/O Modes	VI-2-2
VI.2.4 EGPIO Operating Clock.....	VI-2-3
VI.2.5 Details of Control Registers	VI-2-4
0x300C00–0x300C04: Px I/O Control Registers (pPx_IOC)	VI-2-5
0x300C01–0x300C05: Px Port Data Registers (pPx_DATA)	VI-2-6
0x300C20–0x300C25: Pxx Port Function Select Registers (pPx_CFP0/1).....	VI-2-7

VII PERIPHERAL MODULES 5 (ANALOG)

VII.1 A/D Converter (ADC)	VII-1-1
VII.1.1 Features and Structure of A/D Converter	VII-1-1
VII.1.2 Input Pins of A/D Converter	VII-1-2
VII.1.3 A/D Converter Operating Clock	VII-1-3
VII.1.4 Setting A/D Converter.....	VII-1-4
VII.1.5 Control and Operation of A/D Conversion	VII-1-9
VII.1.6 A/D Converter Interrupt and DMA.....	VII-1-13
VII.1.7 Details of Control Registers	VII-1-15
0x300520: A/D Clock Control Register (pAD_CLKCTL).....	VII-1-16
0x300540: A/D Conversion Result Register (pAD_ADD)	VII-1-17
0x300542: A/D Trigger/Channel Select Register (pAD_TRIG_CHNL)	VII-1-18
0x300544: A/D Control/Status Register (pAD_EN_SMPL_STAT)	VII-1-20
0x300546: A/D Channel Status Flag Register (pAD_END).....	VII-1-23
0x300548–0x300550: A/D Ch.x Conversion Result Buffer Registers (pAD_CHx_BUF)	VII-1-24
0x300558: A/D Upper Limit Value Register (pAD_UPPER).....	VII-1-25
0x30055A: A/D Lower Limit Value Register (pAD_LOWER).....	VII-1-26
0x30055C: A/D Conversion Complete Interrupt Mask Register (pAD_CH04_INTMASK)	VII-1-27
0x30055E: A/D Converter Mode Select/Internal Status Register (pAD_ADVMODE).....	VII-1-28
VII.1.8 Precautions.....	VII-1-29

VIII PERIPHERAL MODULES 6 (LCD)

VIII.1 LCD Controller (LCDC)	VIII-1-1
VIII.1.1 Overview	VIII-1-1
VIII.1.2 Block Diagram	VIII-1-3
VIII.1.3 Output Pins of the LCD Controller	VIII-1-4
VIII.1.4 System Settings	VIII-1-6
VIII.1.4.1 Configuration of Display Data Memory (VRAM)	VIII-1-6
VIII.1.4.2 Setting the LCDC Clock	VIII-1-7
VIII.1.5 Setting the LCD Panel	VIII-1-9
VIII.1.5.1 Types of Panels	VIII-1-9
VIII.1.5.2 STN Panel Timing Parameters	VIII-1-10
VIII.1.5.3 HR-TFT Panel Timing Parameters	VIII-1-12
VIII.1.5.4 Display Modes	VIII-1-17
VIII.1.5.5 Look-up Tables	VIII-1-20
VIII.1.5.6 Frame Rates	VIII-1-26
VIII.1.5.7 Other Settings	VIII-1-26
VIII.1.6 Display Control	VIII-1-27
VIII.1.6.1 Controlling LCD Power Up/Down	VIII-1-27
VIII.1.6.2 Setting the Display Start Address and Line Address Offset	VIII-1-28
VIII.1.6.3 Writing Display Data	VIII-1-29
VIII.1.6.4 Inverting and Blanking the Display	VIII-1-29
VIII.1.6.5 Picture-in-Picture Plus	VIII-1-29
VIII.1.7 LCDC Interrupt and DMA	VIII-1-32
VIII.1.8 Power Save	VIII-1-34
VIII.1.9 Details of Control Registers	VIII-1-35
0x301A00: Frame Interrupt Register (pLCDC_INT)	VIII-1-36
0x301A04: Status and Power Save Configuration Register (pLCDC_PS)	VIII-1-37
0x301A10: Horizontal Display Register (pLCDC_HD)	VIII-1-38
0x301A14: Vertical Display Register (pLCDC_VD)	VIII-1-39
0x301A18: MOD Rate Register (pLCDC_MR)	VIII-1-40
0x301A20: Horizontal Display Start Position Register (pLCDC_HDPS)	VIII-1-41
0x301A24: Vertical Display Start Position Register (pLCDC_VDPS)	VIII-1-42
0x301A28: FPLINE Pulse Setup Register (pLCDC_L)	VIII-1-43
0x301A2C: FPFRAME Pulse Setup Register (pLCDC_F)	VIII-1-44
0x301A30: FPFRAME Pulse Offset Register (pLCDC_FO)	VIII-1-45
0x301A40: HR-TFT Special Output Register (pLCDC_TSO)	VIII-1-46
0x301A44: TFT_CTL1 Pulse Register (pLCDC_TC1)	VIII-1-47
0x301A48: TFT_CTL0 Pulse Register (pLCDC_TC0)	VIII-1-48
0x301A4C: TFT_CTL2 Register (pLCDC_TC2)	VIII-1-49
0x301A60: LCDC Display Mode Register (pLCDC_DMD)	VIII-1-50
0x301A64: IRAM Select Register (pLCDC_IRAM)	VIII-1-53
0x301A70: Main Window Display Start Address Register (pLCDC_MADD)	VIII-1-54
0x301A74: Main Window Line Address Offset Register (pLCDC_MLADD)	VIII-1-55
0x301A80: Sub-window Display Start Address Register (pLCDC_SADD)	VIII-1-56
0x301A88: Sub-window Start Position Register (pLCDC_SSP)	VIII-1-57
0x301A8C: Sub-window End Position Register (pLCDC_SEP)	VIII-1-58
0x301AA0: Look-up Table Data Register 0 (pLCDC_LUT_03)	VIII-1-59
0x301AA4: Look-up Table Data Register 1 (pLCDC_LUT_47)	VIII-1-60
0x301AA8: Look-up Table Data Register 2 (pLCDC_LUT_8B)	VIII-1-61
0x301AAC: Look-up Table Data Register 3 (pLCDC_LUT_CF)	VIII-1-62
VIII.1.10 Precautions	VIII-1-63

VIII.2 IVRAM and IVRAM Arbiter	VIII-2-1
VIII.2.1 IVRAM	VIII-2-1
VIII.2.2 IVRAM Arbiter	VIII-2-2
VIII.2.3 IVRAM Arbiter Operating Clock	VIII-2-3
VIII.2.4 Details of the Control Register	VIII-2-4
0x301A64: IRAM Select Register (pLCDC_IRAM)	VIII-2-4

IX PERIPHERAL MODULES 7 (USB)

IX.1 USB Function Controller (USB)	IX-1-1
IX.1.1 Outline of the USB Function Controller.....	IX-1-1
IX.1.2 Pins for the USB Interface	IX-1-2
IX.1.3 USB Operating Clocks and DMA Registers.....	IX-1-3
IX.1.3.1 Controlling the USB Clocks	IX-1-3
IX.1.3.2 Setting the Misc Register	IX-1-3
IX.1.3.3 Setting the ITC and HSDMA Controllers	IX-1-4
IX.1.4 Functional Description	IX-1-7
IX.1.4.1 USB Control	IX-1-8
IX.1.4.2 FIFO Management	IX-1-25
IX.1.4.3 Port Interface.....	IX-1-28
IX.1.4.4 Snooze	IX-1-33
IX.1.5 Registers.....	IX-1-34
IX.1.5.1 List of Registers.....	IX-1-34
IX.1.5.2 Detailed Description of Registers	IX-1-37
0x300900: MainIntStat (Main Interrupt Status).....	IX-1-37
0x300901: SIE_IntStat (SIE Interrupt Status).....	IX-1-38
0x300902: EPrIntStat (EPr Interrupt Status)	IX-1-39
0x300903: DMA_IntStat (DMA Interrupt Status)	IX-1-40
0x300904: FIFO_IntStat (FIFO Interrupt Status).....	IX-1-41
0x300907: EP0IntStat (EP0 Interrupt Status).....	IX-1-42
0x300908: EPaIntStat (EPa Interrupt Status).....	IX-1-43
0x300909: EPbIntStat (EPb Interrupt Status).....	IX-1-44
0x30090A: EPcIntStat (EPc Interrupt Status).....	IX-1-45
0x30090B: EPdIntStat (EPd Interrupt Status)	IX-1-46
0x300910: MainIntEnb (Main Interrupt Enable).....	IX-1-47
0x300911: SIE_IntEnb (SIE Interrupt Enable).....	IX-1-48
0x300912: EPrIntEnb (EPr Interrupt Enable)	IX-1-49
0x300913: DMA_IntEnb (DMA Interrupt Enable)	IX-1-50
0x300914: FIFO_IntEnb (FIFO Interrupt Enable).....	IX-1-51
0x300917: EP0IntEnb (EP0 Interrupt Enable).....	IX-1-52
0x300918: EPaIntEnb (EPa Interrupt Enable).....	IX-1-53
0x300919: EPbIntEnb (EPb Interrupt Enable).....	IX-1-54
0x30091A: EPcIntEnb (EPc Interrupt Enable).....	IX-1-55
0x30091B: EPdIntEnb (EPd Interrupt Enable)	IX-1-56
0x300920: RevisionNum (Revision Number).....	IX-1-57
0x300921: USB_Control (USB Control)	IX-1-58
0x300922: USB_Status (USB Status)	IX-1-59
0x300923: XcvrControl (Xcvr Control).....	IX-1-60
0x300924: USB_Test (USB Test).....	IX-1-61
0x300925: EPnControl (Endpoint Control)	IX-1-62
0x300926: EPrFIFO_Clr (EPr FIFO Clear).....	IX-1-63
0x30092E: FrameNumber_H (Frame Number HIGH)	IX-1-64
0x30092F: FrameNumber_L (Frame Number LOW)	IX-1-65
0x300930–0x300937: EP0Setup_0 (EP0 Setup 0)–EP0Setup_7 (EP0 Setup 7)	IX-1-66
0x300938: USB_Address (USB Address)	IX-1-67
0x300939: EP0Control (EP0 Control).....	IX-1-68
0x30093A: EP0ControlIN (EP0 Control IN).....	IX-1-69
0x30093B: EP0ControlOUT (EP0 Control OUT).....	IX-1-70
0x30093F: EP0MaxSize (EP0 Max Packet Size)	IX-1-71
0x300940: EPaControl (EPa Control).....	IX-1-72
0x300941: EPbControl (EPb Control).....	IX-1-73
0x300942: EPcControl (EPc Control)	IX-1-74
0x300943: EPdControl (EPd Control).....	IX-1-75
0x300950: EPaMaxSize_H (EPa Max Packet Size HIGH)	IX-1-76

0x300951: EPaMaxSize_L (EPa Max Packet Size LOW)	IX-1-76
0x300952: EPaConfig_0 (EPa Configuration 0)	IX-1-77
0x300953: EPaConfig_1 (EPa Configuration 1)	IX-1-78
0x300954: EPbMaxSize_H (EPb Max Packet Size HIGH)	IX-1-79
0x300955: EPbMaxSize_L (EPb Max Packet Size LOW)	IX-1-79
0x300956: EPbConfig_0 (EPb Configuration 0)	IX-1-80
0x300957: EPbConfig_1 (EPb Configuration 1)	IX-1-81
0x300958: EPcMaxSize_H (EPc Max Packet Size HIGH)	IX-1-82
0x300959: EPcMaxSize_L (EPc Max Packet Size LOW)	IX-1-82
0x30095A: EPcConfig_0 (EPc Configuration 0)	IX-1-83
0x30095B: EPcConfig_1 (EPc Configuration 1)	IX-1-84
0x30095C: EPdMaxSize_H (EPd Max Packet Size HIGH)	IX-1-85
0x30095D: EPdMaxSize_L (EPd Max Packet Size LOW)	IX-1-85
0x30095E: EPdConfig_0 (EPd Configuration 0)	IX-1-86
0x30095F: EPdConfig_1 (EPd Configuration 1)	IX-1-87
0x300970: EPaStartAdrs_H (EPa FIFO Start Address HIGH)	IX-1-88
0x300971: EPaStartAdrs_L (EPa FIFO Start Address LOW)	IX-1-88
0x300972: EPbStartAdrs_H (EPb FIFO Start Address HIGH)	IX-1-89
0x300973: EPbStartAdrs_L (EPb FIFO Start Address LOW)	IX-1-89
0x300974: EPcStartAdrs_H (EPc FIFO Start Address HIGH)	IX-1-90
0x300975: EPcStartAdrs_L (EPc FIFO Start Address LOW)	IX-1-90
0x300976: EPdStartAdrs_H (EPd FIFO Start Address HIGH)	IX-1-91
0x300977: EPdStartAdrs_L (EPd FIFO Start Address LOW)	IX-1-91
0x300980: CPU_JoinRd (CPU Join FIFO Read)	IX-1-92
0x300981: CPU_JoinWr (CPU Join FIFO Write)	IX-1-93
0x300982: EnEPnFIFO_Access (Enable EPn FIFO Access)	IX-1-94
0x300983: EPnFIFOforCPU (EPn FIFO for CPU)	IX-1-95
0x300984: EPnRdRemain_H (EPn FIFO Read Remain HIGH)	IX-1-96
0x300985: EPnRdRemain_L (EPn FIFO Read Remain LOW)	IX-1-96
0x300986: EPnWrRemain_H (EPn FIFO Write Remain HIGH)	IX-1-97
0x300987: EPnWrRemain_L (EPn FIFO Write Remain LOW)	IX-1-97
0x300988: DescAdrs_H (Descriptor Address HIGH)	IX-1-98
0x300989: DescAdrs_L (Descriptor Address LOW)	IX-1-98
0x30098A: DescSize_H (Descriptor Size HIGH)	IX-1-99
0x30098B: DescSize_L (Descriptor Size LOW)	IX-1-99
0x30098F: DescDoor (Descriptor Door)	IX-1-100
0x300990: DMA_FIFO_Control (DMA FIFO Control)	IX-1-101
0x300991: DMA_Join (DMA Join FIFO)	IX-1-102
0x300992: DMA_Control (DMA Control)	IX-1-103
0x300994: DMA_Config_0 (DMA Configuration 0)	IX-1-104
0x300995: DMA_Config_1 (DMA Configuration 1)	IX-1-105
0x300997: DMA_Latency (DMA Latency)	IX-1-106
0x300998: DMA_Remain_H (DMA FIFO Remain HIGH)	IX-1-107
0x300999: DMA_Remain_L (DMA FIFO Remain LOW)	IX-1-107
0x30099C: DMA_Count_HH (DMA Transfer Byte Counter HIGH/HIGH)	IX-1-108
0x30099D: DMA_Count_HL (DMA Transfer Byte Counter HIGH/LOW)	IX-1-108
0x30099E: DMA_Count_LH (DMA Transfer Byte Counter LOW/HIGH)	IX-1-108
0x30099F: DMA_Count_LL (DMA Transfer Byte Counter LOW/LOW)	IX-1-108

X PERIPHERAL MODULES 8 (MP3)

X.1 MP3 Decoder (MP3)	X-1-1
X.1.1 Overview.....	X-1-1
X.1.2 Composition of MP3 Decoder System.....	X-1-3
X.1.2.1 MP3 Decoder BIOS.....	X-1-4
X.1.2.2 Hardware Resource Requirements.....	X-1-5
X.1.3 Usage of MP3 API Functions.....	X-1-6
X.1.3.1 High-Level MP3 API Functions.....	X-1-6
X.1.3.2 Low-Level MP3 API Functions.....	X-1-11
X.1.4 MP3 Format.....	X-1-12
X.1.4.1 MP3 File Format.....	X-1-12
X.1.4.2 MP3 Bit Stream Format.....	X-1-13
X.1.5 Details of MP3 Decoder BIOS API.....	X-1-14
X.1.5.1 Return Values and Event Values.....	X-1-14
X.1.5.2 c33MP3DecOpen — Open MP3 Decoder.....	X-1-15
X.1.5.3 c33MP3DecClose — Close MP3 Decoder.....	X-1-18
X.1.5.4 c33MP3DecStart — Start Playback.....	X-1-19
X.1.5.5 c33MP3DecStop — Stop Playback.....	X-1-21
X.1.5.6 c33MP3DecPause — Pause Playback.....	X-1-22
X.1.5.7 c33MP3DecResume — Resume Playback.....	X-1-23
X.1.5.8 c33MP3DecResync — Resynchronization.....	X-1-24
X.1.5.9 TransferData (Callback Function) — Set MP3 Data.....	X-1-26
X.1.5.10 Event (Callback Function) — Event Handler.....	X-1-28
X.1.5.11 c33MP3CalcStereoProcess — Stereo Process.....	X-1-29
X.1.5.12 c33MP3CalcStereoProcessLSF — Stereo Process for LSF.....	X-1-31
X.1.5.13 c33MP3CalcAntiAlias — Anti Alias Process.....	X-1-33
X.1.5.14 c33MP3CalcIdctLong — Inverse MDCT Calculation for Long Block.....	X-1-34
X.1.5.15 c33MP3CalcIdctWindow — Window Calculation.....	X-1-35
X.1.5.16 c33MP3CalcIdctShort — Inverse MDCT Calculation for Short Block.....	X-1-36
X.1.5.17 c33MP3CalcSubbandSynthesisFdct — Fast-DCT Calculation.....	X-1-37
X.1.5.18 c33MP3CalcSubbandSynthesisWindow — Window Calculation for Sub-band Synthesis.....	X-1-38
X.1.5.19 c33MP3CalcSubbandSynthesisWindow2/4/8 — Window Calculation for Sub-band Synthesis 2/4/8.....	X-1-39
X.1.6 Performance.....	X-1-40
X.1.6.1 Support Formats.....	X-1-40
X.1.6.2 CPU Occupancy Ratio.....	X-1-41

Appendix

Appendix A I/O Map.....	AP-A-1
0x300010–0x300020	Misc Register (1) AP-A-2
0x300260–0x3002AF	Interrupt Controller AP-A-3
0x300300–0x30031B	Card Interface..... AP-A-11
0x300380–0x3003D5	I/O Ports AP-A-13
0x300520–0x30055E	A/D Converter..... AP-A-24
0x300660–0x30066C	Watchdog Timer AP-A-27
0x300780–0x3007EA	16-bit Timer AP-A-29
0x300900–0x30099F	USB Function Controller..... AP-A-37
0x300B00–0x300B4F	Serial Interface AP-A-47
0x300C00–0x300C25	Extended Ports..... AP-A-52
0x300C40–0x300C4D	Misc Register (2) AP-A-54
0x301100–0x301105	Intelligent DMA AP-A-56
0x301120–0x30119E	High-Speed DMA AP-A-57
0x301500–0x301510	SRAM Controller AP-A-70
0x301600–0x301610	SDRAM Controller..... AP-A-71
0x301700–0x30171C	SPI..... AP-A-72
0x301800–0x30181C	DCSIO AP-A-73
0x301900–0x301928	Real Time Clock AP-A-74
0x301A00–0x301AAC	LCD Controller..... AP-A-76
0x301B00–0x301B24	Clock Management Unit AP-A-83
0x301C00–0x301C20	I ² S Interface..... AP-A-87
Appendix B Differences Between C33 PE Core and Other C33 Core	AP-B-1
B.1 Instructions	AP-B-1
B.2 Registers.....	AP-B-2
B.3 Address Space and Other	AP-B-2
Appendix C Development Tools	AP-C-1
C.1 Major Development Tools.....	AP-C-1
C.2 Precautions on Use of S5U1C33001C	AP-C-2
Appendix D Boot	AP-D-1
D.1 Boot Mode	AP-D-1
D.2 NAND Flash Boot	AP-D-2
D.2.1 Configuration of NAND Flash Boot System	AP-D-2
D.2.2 NAND Flash Boot Sequence	AP-D-3
D.2.3 NAND Flash Data.....	AP-D-5
D.3 NOR Flash/External ROM Boot.....	AP-D-7
D.3.1 Configuration of NOR Flash/External ROM Boot System	AP-D-7
D.3.2 NOR Flash/External ROM Boot Sequence	AP-D-7
D.3.3 Reset Vector for NOR Flash/External ROM Boot	AP-D-8
D.4 SPI-EEPROM Boot.....	AP-D-9
D.4.1 Configuration of SPI-EEPROM Boot System	AP-D-9
D.4.2 SPI-EEPROM Boot Sequence	AP-D-10
D.4.3 EEPROM Data	AP-D-11
D.5 PC RS232C Boot.....	AP-D-12
D.5.1 Configuration of PC RS232C Boot System	AP-D-12
D.5.2 PC RS232C Boot Sequence	AP-D-13
D.5.3 Transfer Data	AP-D-14
D.6 Precautions.....	AP-D-15
Appendix E Summary of Precautions	AP-E-1
Appendix F Supplementary Description for Clock Control	AP-F-1

S1C33E08 Technical Manual

I S1C33E08 SPECIFICATIONS

I.1 Overview

The S1C33E08 is a high cost performance 32-bit RISC controller for specific applications that require an MP3 player, several general-purpose I/O ports, a powerful PWM Timer/Counter function, several serial interfaces including USB-FS device controller, an ADC and a LCD display system, such as middle range electronic dictionaries and educational products with voice/music playback function.

The S1C33E08 consists of a 32-bit RISC CPU-Core, an MP3 decoder supporting Layer III specifications, a generic DMA controller, a USB-FS device controller, a PWM control Timer/Counter, several interfaces (SIO including IrDA1.0 and ISO7816-3 protocol, SPI, I²S and DCSIO), an ADC, RAM/Shared IVRAM and RTC implemented by EPSON SoC design technology using 0.18 μ m Mixed Analog Low CMOS Process.

Table I.1.1 Product Line

Model	Package
S1C33E08F00A***	QFP24-144pin
S1C33E08B00A***	PFBGA-180pin
S1C33E08D00A***	Die form

Note: The EPSON S1C33E08 incorporates MP3 technology of which Thomson SA in France holds the patent. Manufacturers using the EPSON S1C33E08 to develop MP3 products must pay royalties to Thomson SA in order to procure the license for the MP3 technology.

The main functions and features of the S1C33E08 are outlined below.

Technology

- 0.18 μ m AL-4-Layers mixed analog low power CMOS process technology

CPU

- EPSON original C33 PE 32-bit RISC CPU-Core with AMBA bus optimized for SoC
- Max. 60 MHz operation
- Internal 2-stage pipeline and 4 instruction queues
- Instruction set: 128 instructions (16-bit fixed length)
- Basic instructions are compatible with the S1C33 32-bit RISC Cores.
- Dual AMBA bus system for CPU and LCDC

Internal Memories

- 8K-byte RAM (1K bytes are available when the MP3 decoder BIOS is used)
- 12K-byte IVRAM (used as general-purpose RAM, VRAM, or MP3 work area)
- 2K-byte DST RAM (used as general-purpose RAM or IDMA descriptor table RAM)

Oscillator Circuit / PLL

OSC3 Oscillator Circuit

- Crystal oscillation: 5 MHz min. to 48 MHz max.
- Ceramic oscillation: 48 MHz, fixed
- External clock input: 5 MHz min. to 48 MHz max.
- A 48 MHz clock source with 0.25% of accuracy should be connected for using the USB function.

PLL

- PLL input frequency: 5 MHz min. to 50 MHz max. (OSC3 \times 1, \times 1/2, \times 1/3, ... \times 1/9, \times 1/10)
- PLL output frequency: 20 MHz min. to 90 MHz max.
- Multiplication rate: \times 1, \times 2, \times 4, \times 8, \times 10, \times 15

OSC1 Oscillator Circuit

- Crystal oscillation: 32.768 kHz typ.
- External clock input: 32.768 kHz typ.

High Speed Bus (HB) Modules

SRAMC (SRAM Controller)

- 25-bit address lines and 8/16-bit selectable data bus
- UP to a 512M-byte (A[24:0]) address space is provided for each chip enable signal.
- Max. 8 chip enable signals are available to connect external devices.
- Programmable bus wait cycle (0 to 7 cycles)
- Supports external wait signals.
- 4GB physical address space is available.
 - The physical address space is divided into 23 areas: Area 0 to Area 22.
 - Areas 0 to 4 and Area 6 are system reserved.
- Supports only Little-Endian access to each area.
- Memory mapped I/O
- Supports both A0 and BS (Bus Strobe) access type external devices.
- SRAM, ROM, and Flash ROM direct access interfaces are built in.

SDRAMC (SDRAM Controller with SDRAM APP and AHB Local Bus Arbiter)

- Supports SDRAM direct interface.
- Supports only SDRAM devices with 16-bit data bus.
 - Minimum configuration: 16M bits (2MB), 16-bit SDRAM × 1
 - Maximum configuration: 512M bits (64MB), 16-bit SDRAM × 1
- CAS latency: 1, 2 or 3 programmable
- Supports burst and single read/write.
- Supports DQM (byte write) function.
- Supports max. 4 SDRAM banks and bank active mode.
- Incorporates a 12-bit auto-refresh counter.
- Intelligent self-refresh function for low power operation
- 2-stage × 32-bit data buffer and 8-stage × 16-bit × 2-slot instruction buffer built-in
- Supports up to 90 MHz SDRAM clock.
 - When the CPU clock is 48 MHz, the SDRAM clock can be set to 48 MHz.
 - When the CPU clock is 45 MHz, the SDRAM clock can be set to 90 MHz using the PLL.
- Arbitrates ownership of the external bus between the CPU, DMAC, LCDC and SRAMC.

DMAC (Direct Memory Access Controller)

- 4-ch. high speed hardware DMA
- 128-ch. intelligent DMA (variable data transfer controller) with specific control table

IVRAMARB (Internal Video RAM Arbiter)

- Contains a 12KB SRAM (3,072 words × 16 bits × 2).
- Arbitrates accesses from the LCDC and CPU.
- Allows the CPU and LCDC to access IVRAM in minimum 2 cycles by 32-bit access.
- Supports UMA (Unified Memory Access) for display.
- IVRAM is configurable as a 12KB general-purpose RAM in Area 0 using a control register if it is not used as a video RAM.

Peripheral Bus (SAPB) Modules

TCU (Timer/Counter Unit with PWM Outputs)

- 6-ch. 16-bit timer/counter
- Supports PWM outputs with DA16 (Digital D/A) mode.
- Contains a prescaler, which can divide the peripheral clock by 1 to 4,096, to generate the operating clock for each channel.
- Possible to invoke DMA transfer.

WDT (Watchdog Timer)

- 30-bit watchdog timer to generate an NMI interrupt
- The watchdog timer overflow cycle (NMI interrupt cycle) is programmable.
- The watchdog timer overflow signal can be output outside the IC.

ADC (A/D Converter)

- 5-ch. 10-bit A/D converter
- Upper/lower limit interrupt is available.
- Each ADC channel includes a data buffer.
- Contains a prescaler, which can divide the peripheral clock by 2 to 256, to generate the operating clock for ADC.

ITC (Interrupt Controller)

- Possible to invoke DMA transfer
- DMA controller interrupt: 5 types
- Input interrupt: 18 types
- TCU interrupt: 12 types
- EFSIO interrupt: 9 types
- ADC interrupt: 2 types
- RTC interrupt: 1 type
- SPI interrupt: 3 types
- DCSIO interrupt: 1 type
- USB interrupt: 2 types
- I²S interrupt: 1 type
- LCDC interrupt: 1 type

GPIO (General-Purpose I/O Ports)

- Max. 82 ports in the QFP24-144pin model.
- * The S1C33E08 GPIO ports are shared with other peripheral function pins (EFSIO, PWM etc.). Therefore, the number of GPIO ports depends on the peripheral functions used.

USB (Universal Serial Bus 2.0 compliant Full-Speed Device Controller)

- Supports USB2.0 full speed (12M bps) mode.
- Supports auto negotiation function.
- Supports control, bulk, isochronous and interrupt transfers.
- Supports 4 general-purpose end points and end point 0 (control).
- Embedded 1K-byte programmable FIFO
- Supports 8-bit local bus DMA port.
- Possible to invoke DMA transfer.
- Supports Async. DMA transfer.
- Supports DMA slave mode.
- Fixed 48 MHz clock for USB-FS.
- Supports snooze mode.

RTC (Real Time Clock)

- Contains time counters (seconds, minutes, and hours) and calendar counters (days, days of the week, months, and year).
- BCD data can be read from and written to both counters.
- Capable of controlling the starting and stopping of time clocks.
- 24-hour or 12-hour mode can be selected.
- A 30-second correction function can be implemented in software.
- Periodic interrupts are possible.

CARD (Serial Input/Output with Direction Control)

- Provides SmartMedia I/F signals (#SMRE, #SMWE).
- Provides 8- or 16-bit NAND Flash I/F signals.
- ECC function is available when reading/writing from/to NAND Flash type devices.
- Supports NAND Flash booting function.
- Supports EPSON middleware FS33.

EFSIO (Extended Serial Interface with FIFO Buffer)

- 3-ch. clock sync./async. serial interface
- Contains FIFO data buffers (4 receive data buffer and 2 transmit data buffer are available for each channel).
- Supports IrDA1.0 interface.
- Contains a baud-rate generator (12-bit programmable timer).
- Supports ISO7816 mode (Ch.1 only).
 - Alternative MSB or LSB
 - Memory card interface compatible with ISO7816-3 T=0 & T=1 protocol
 - Programmable baud-rate and guard-time generation
 - ISO7816 acknowledge and automatically repeat transmission
- Possible to invoke DMA transfer.

SPI (Serial Peripheral Interface)

- 1 ch. SPI that operates in either master or slave mode
- Supports 1- to 32-bit data transfer.
- Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
- A 1 to 65,536 clocks of delay can be inserted between transfers.
- Generates transmit data register empty and receive data register full interrupts.
- Supports both MMC and SD-card capabilities.
- Possible to invoke DMA transfer.

DCSIO (Direction Control Serial Input/Output Port)

- 2-ch. input/output ports with a serial shifter
- Input/output level detection to drive a state machine
- 1-wire or 2-wire communication protocol is simulated with software.

EGPIO (Extended GPIO)

- Max. 17 configurable GPIO ports are available in addition to the standard GPIO ports. In die form, Max. 91 ports are available.
 - * The EGPIO ports are shared with other peripheral function pins. Therefore, the number of EGPIO ports depends on the peripheral functions used.
- Most ports have a pull-up resistor that can be enabled/disabled with the control register.
- Possible to drive the ports low.

CMU (Extended Clock Management Unit)

- Controls clock supply to each peripheral module (static).
- Manages reset and NMI inputs.
- Switches the system clock source (MCLK, SDRAM_CLK, or RTC_CLK).
- Controls the MCLK and RTC_CLK oscillator circuits.
- Turns on/off and controls frequency multiplication rate of the PLL.
- Controls clocks according to the standby mode (SLEEP and HALT).
- Controls divide ratios of the LCDC clock.
- Manages the external bus clock.

MISC (Misc. Setting Register)

- USB/RTC wait configuration registers
- Debug port function select register
- Boot mode configuration register

I²S (Inter-IC Sound Bus Interface)

- Supports universal audio I²S Bus Interface.
- Operates as the master to generate the bit clock, word-select signal, data and master clock.
- Generates the I²S interrupt signal.
- Possible to invoke DMA transfer.

LCDC (STN/TFT LCD Controller with AMBA Bus)**VRAM:**

- Built-in a 12KB RAM usable as a display buffer or general-purpose RAM (register selectable)
- Supports the UMA method allowing LCDC to access SDRAM (external VRAM) or IVRAM (internal VRAM).
- The external VRAM map (SDRAM) is configurable.
- The sub-window area can be located in IVRAM or external VRAM regardless of whether it contains the main window area or not.

Display Support:

- 4- or 8-bit monochrome LCD interface
 - 4- or 8-bit color LCD interface
 - Single-panel, single-drive passive displays
 - 12-bit Generic HR-TFT interface
 - 320 × 240-dot Sharp HR-TFT panel, SII liquid TFT panel, or some other TFT panels
 - Typical resolutions
 - 320 × 240 (8-bpp mode, external VRAM is required) bpp = bits per pixel
 - 320 × 240 (1-bpp mode)
- * Note that the panel width must be a multiple of 16 ÷ bits per pixel.

Display Modes:

- Due to frame rate modulation, grayscale display is possible in up to 16 shades of gray when a monochrome passive LCD panel is used.
 - Two-shade display in 1-bpp mode
 - Four-shade display in 2-bpp mode
 - 16-shade display in 4-bpp mode
- A maximum of 64K colors can be simultaneously displayed on a color passive LCD panel.
 - 256-color display in 8-bpp mode
 - 4K-color display in 12-bpp mode
 - 64K-color display in 16-bpp mode
- A maximum of 4096 colors can be simultaneously displayed on a TFT panel.
 - Two-color display in 1-bpp mode
 - Four-color display in 2-bpp mode
 - 16-color display in 4-bpp mode
 - 256-color display in 8-bpp mode
 - 4K-color display in 12-bpp mode
- A look-up table, which consists of 6 bits × 16 entries × 3 colors, is provided.
 - In monochrome 1/2/4-bpp or color 8/12-bpp mode, the look-up table can be used or bypassed.
 - In color 1/2/4/16-bpp mode, the look-up table cannot be used (must be bypassed).

Display Features:

- Picture-in-Picture Plus (PIP+)

Picture-in-Picture Plus enables a secondary window (or sub-window) within the main display window. The sub-window may be positioned anywhere within the main window and is controlled through registers. The sub-window retains the same color depth as the main window.

The speed of generating a sub-window by hardware is faster than software. By using this PIP+ function, it can greatly speed the GUI performance and CPU can have more performance to assign other processing. (e.g. Voice etc.)
- 12-bit Generic HR-TFT interface

The 12-bit Generic HR-TFT interface can support 320 × 240 Sharp HR-TFT panel, SII TFT panel or some other TFT panels. Because the timing of FPFRRAM, FPLINE, and TFT_CTL0–3 are not fixed for TFT panels, they can be controlled by register setting. By different register settings, you can get your specified TFT I/F signal timing.
- Clock source

The LCDC clock can be internally divided 48 MHz by 1 to 16. The clock division register is located in CMU part.

MP3 Decoder

- MP3 decoder BIOS provides MP3 decode and playback API functions.
 - Main APIs for playback: Start playback, stop playback, pause playback, resume playback
 - Main APIs for decoding: Stereo processing, anti alias processing, IDCT calculation, sub-band analysis
 - Sampling frequency: 32, 44.1, and 48 kHz (MPEG1 Audio Layer-III)
 16, 22.05, and 24 kHz (MPEG2 Audio Layer-III LSF)
 - * The higher the sampling frequency, the more the CPU occupancy rate during playback increases. The CPU clock frequency should be set as high as possible to perform parallel processing using the RTOS, etc.
 - Bit-rate: 32 to 320 kbps (MPEG1 Audio Layer-III, CBR/VBR/ABR)
 8 to 160 kbps (MPEG2 Audio Layer-III LSF, CBR/VBR/ABR)
 - Channel mode: Stereo (joint stereo and dual channel) and monaural
 - Sound quality mode: High/middle/low
 - * The sound quality mode affects the CPU occupancy rate.
 - CPU occupancy rate: The CPU occupancy rate depends on the CPU clock frequency, sound quality mode, MP3 data sampling frequency and bit-rate, however, the S1C33E08 is capable of multitask operations under a real-time OS such as the RTOS even in MP3 playback.

Example) When CPU clock = 48 MHz, sampling frequency = 44.1 kHz and sound quality mode = middle, the MP3 decode/playback processing occupies the CPU for about 18 ms in about 26 ms cycles (CPU occupancy rate: 70% or less).
 - A built-in hardware MP3 accelerator
- * When using the MP3 decoder BIOS (MP3 decoder module), a 7KB internal RAM area, 12KB IVRAM, a 20KB external RAM area (min.), and a 40-byte stack area are required. When using the MP3 calculation module only, a 40-byte stack area is required.

Operating Voltage

- V_{DD} (Core): 1.70 to 1.90 V (typ. 1.8 V) when a ceramic resonator is used for the USB clock
- V_{DD} (Core): 1.65 to 1.95 V (typ. 1.8 V) when a crystal is used or an external clock is input
- PLV_{DD}: 1.65 to 1.95 V (typ. 1.8 V)
- V_{DDH} (I/O): 2.70 to 3.60 V when the USB is not used (5-V tolerant I/O not supported)
- V_{DDH} (I/O): 3.00 to 3.60 V (typ. 3.3 V) when the USB is used (5-V tolerant I/O not supported)

Operating Frequency

- CPU: 60 MHz max.
- USB: 48 MHz fixed.
- SDRAMC: 90 MHz max.
- LCDC: 60 MHz max.
- Other peripheral circuits: 60 MHz max.

Operating Temperatures

- -40 to 85°C
 (0 to 70°C when a ceramic resonator is used for the USB clock)

Current Consumption

- During SLEEP: 1.0 μ A typ. (operation clock = 48 MHz)
 - During HALT: 3.0 mA typ. (operation clock = 48 MHz)
 - During execution:
 - Core 19.0 mA typ. (operation clock = 48 MHz)
 - SRAMC 3.4 mA typ. (operation clock = 48 MHz, idle state with the clock supplied)
 - SDRAMC 5.4 mA typ. (operation clock = 48 MHz, idle state with the clock supplied)
 - DMA 3.9 mA typ. (operation clock = 48 MHz, idle state with the clock supplied)
 - LCDC 5.3 mA typ. (operation clock = 48 MHz, idle state with the clock supplied)
 - USB 10.0 mA typ. (operation clock = 48 MHz, idle state with the clock supplied)
 - ADC 260.0 μ A typ. (idle state when ADC is enabled)
 - MP3 decoder 8.0 mA typ. (when MP3 decoder is on playback)
- * By controlling the CPU clock through the Clock-Gear (CMU), current consumption can be reduced.

Shipping Form

- Package: QFP24-144pin (16 mm \times 16 mm \times 1.0 mm and 0.4 mm pin pitch)
PFPGA-180pin (12 mm \times 12 mm \times 1.2 mm and 0.8 mm ball pitch)
- Die form: 168 pads with pad pitch 90 μ m

THIS PAGE IS BLANK.

I.2 Block Diagram

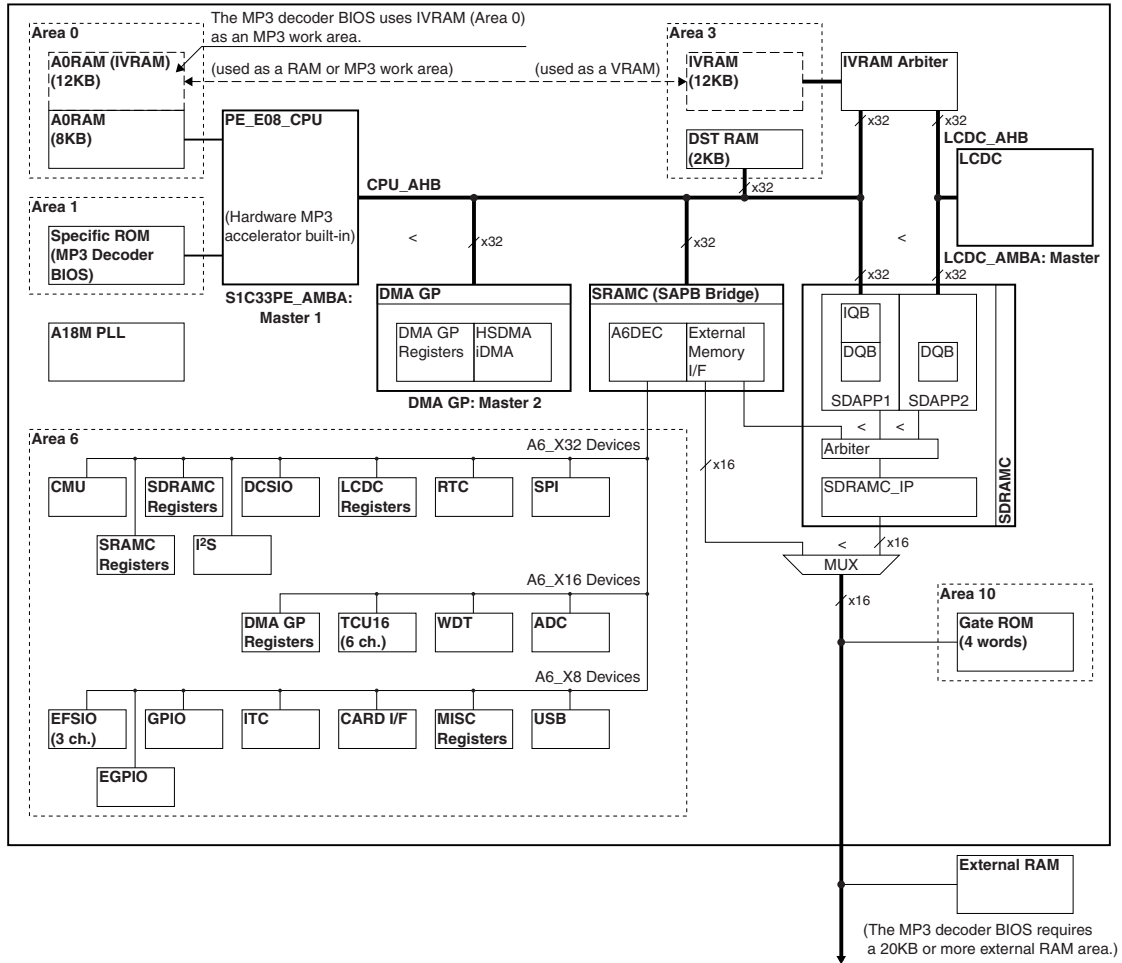


Figure I.2.1 S1C33E08 Block Diagram

THIS PAGE IS BLANK.

I.3 Pin Description

I.3.1 Pin Arrangement

The S1C33E08 comes in a QFP24-144pin or PFBGA-180pin plastic package.

I.3.1.1 QFP Package Pin Arrangement (S1C33E08F00A)

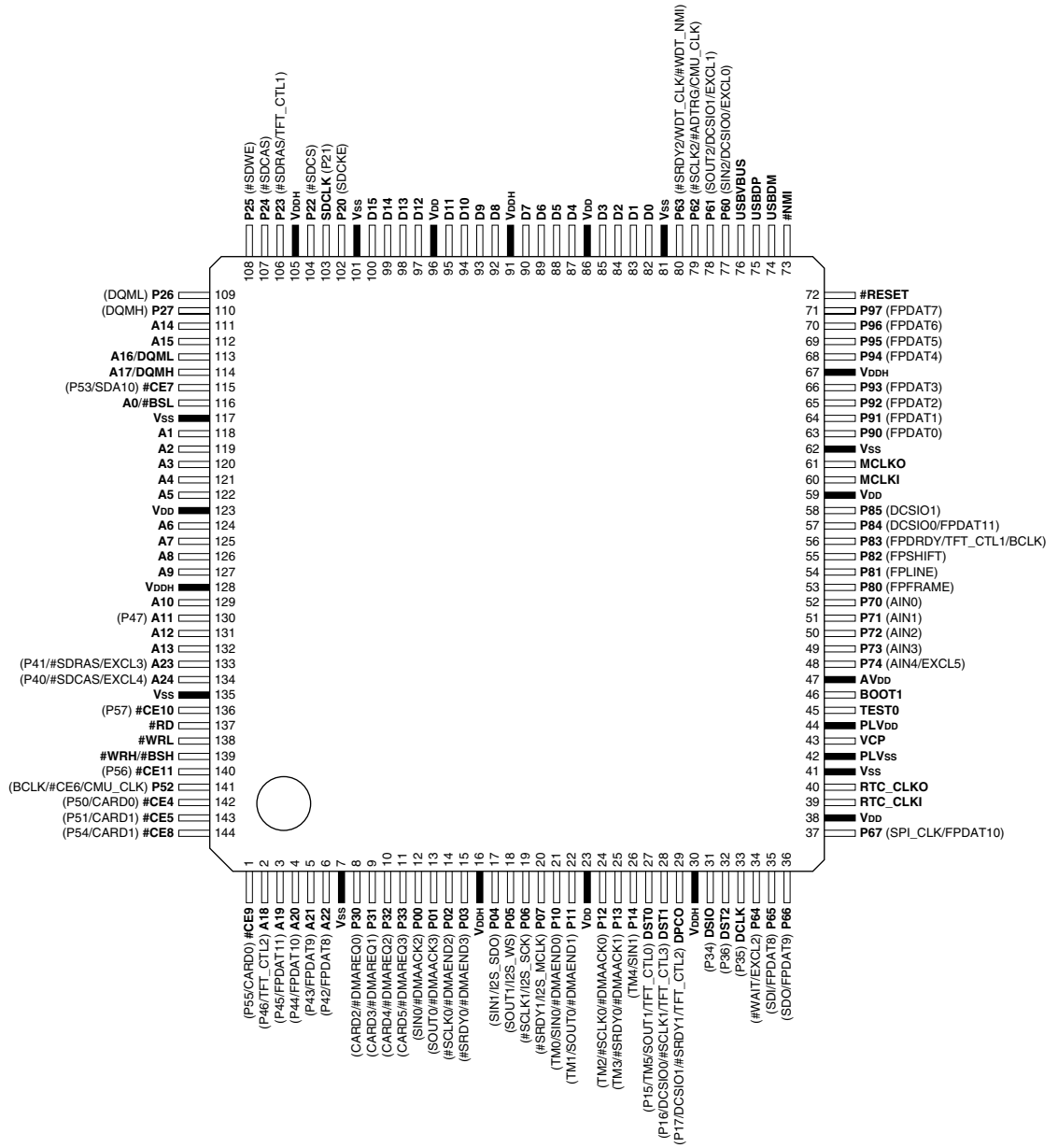


Figure I.3.1.1.1 Pin Arrangement (QFP24-144pin)

I.3.1.2 PFBGA Package Pin Arrangement (S1C33E08B00A)

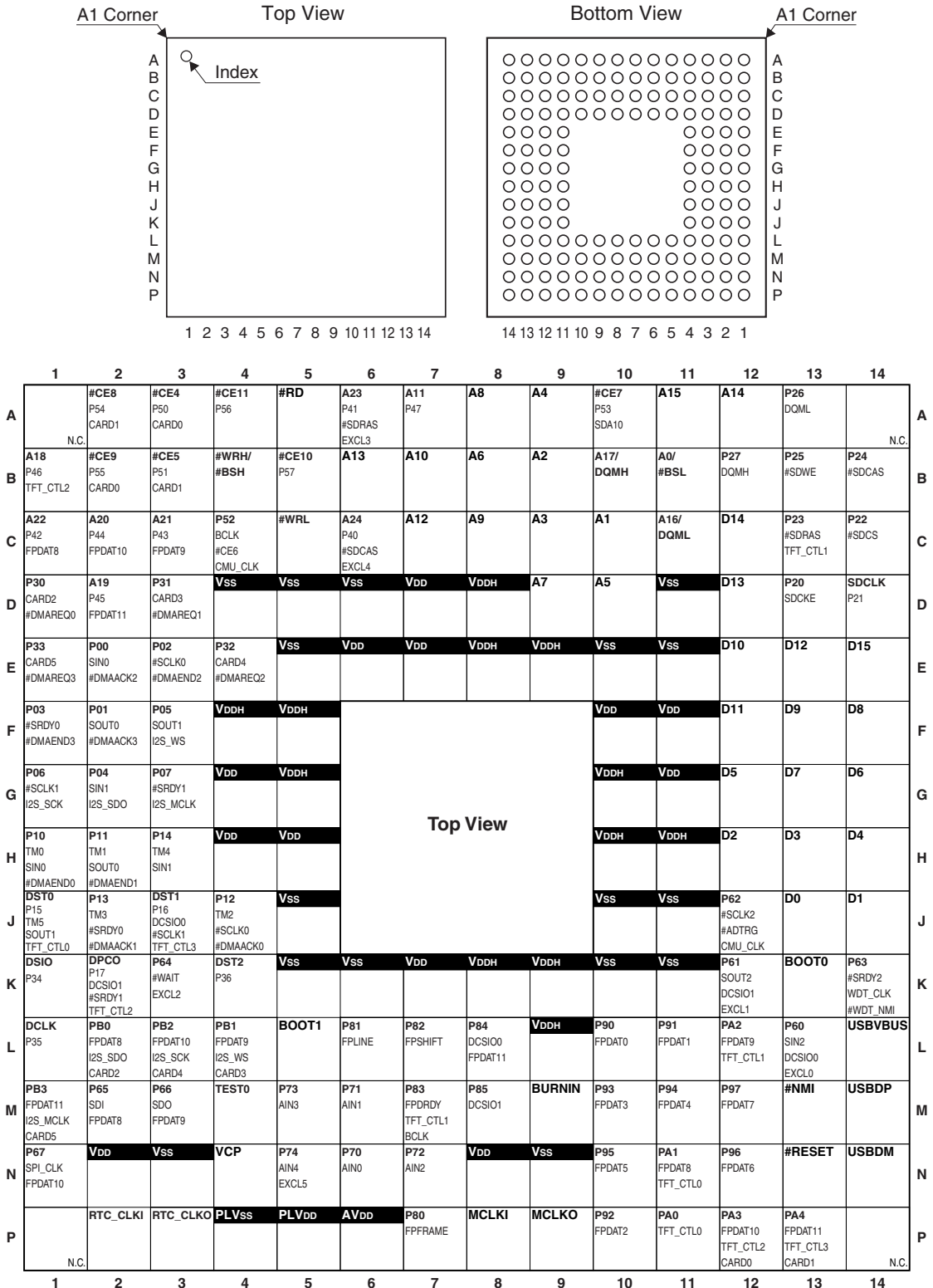


Figure I.3.1.2.1 Pin Arrangement (PFBGA-180pin)

I.3.2 Pin Functions

Tables I.3.2.1 to I.3.2.6 list the function of each pin on the S1C33E08.

Table I.3.2.1 Power Supply Pin List

Pin name	Pin No.		Function
	QFP	PFBGA	
V _{DD}	23,38,59,86,96,123	D7,E6,E7,F10,F11,G4, G11,H4,H5,K7,N2,N8	Power supply (+) for core (1.8 V)
V _{DDH}	16,30,67,91,105,128	D8,E8,E9,F4,F5,G5,G10, H10,H11,K8,K9,L9	Power supply (+) for I/O (3.3 V)
V _{SS}	7,41,62,81,101,117,135	D4,D5,D6,D11,E5,E10, E11,J5,J10,J11,K5,K6, K10,K11,N3,N9	Power supply (-); GND
PLV _{DD}	44	P5	Power supply (+) for PLL (PLV _{DD} = V _{DD})
PLV _{SS}	42	P4	Power supply (-) for PLL (PLV _{SS} = V _{SS})
AV _{DD}	47	P6	Power supply (+) for analog system and AIN0–AIN4 (3.3 V, AV _{DD} = V _{DDH})

Table I.3.2.2 Clock Pin List

Pin name	Pin No.		I/O	Pull-up/down	Function
	QFP	PFBGA			
MCLKI	60	P8	I	–	High speed (OSC3) oscillation input (crystal/ceramic oscillator or external clock input with V _{DD} level)
MCLKO	61	P9	O	–	High speed (OSC3) oscillation output
RTC_CLKI	39	P2	I	–	Real Time Clock (OSC1) oscillation input (crystal/ceramic oscillator or external clock input with V _{DD} level)
RTC_CLKO	40	P3	O	–	Real Time Clock (OSC1) oscillation output
VCP	43	N4	O	–	PLL analog monitor (used for current monitor)

Table I.3.2.3 External Bus Pin List

Pin name	Pin No.		I/O	Pull-up/down	Function
	QFP	PFBGA			
D[7:0]	90–87, 85–82	G13,G14, G12,H14, H13,H12, J14,J13	I/O	*2	Data bus (D7–D0)
D[15:8] PC[7:0]	100–97, 95–92	E14,C12, D12,E13, F12,E12, F13,F14	I/O	*2	D[15:8]: Data bus (D15–D8) PC[7:0]: Extended general-purpose I/O ports
A0 / #BSL	116	B11	i/O (L)	100k PU *1	Address bus (A0) or bus strobe (low byte) signal
A[10:1]	129, 127–124, 122–118	B7,C8, A8,D9, B8,D10, A9,C9, B9,C10	i/O (L)	100k PU *1	Address bus (A10–A1)
A11 P47	130	A7	i/O (L)	100k PU *1	A11: Address bus (A11) (default) P47: General-purpose I/O port
A[15:12]	112,111, 132,131	A11,A12, B6,C7	i/O (L)	100k PU *1	Address bus (A15–A12)
A16 / DQML	113	C11	i/O (L)	100k PU *1	Address bus (A16) or SDRAM data (low byte) input/output mask signal output
A17 / DQMH	114	B10	i/O (L)	100k PU *1	Address bus (A17) or SDRAM data (high byte) input/output mask signal output
A18 P46 TFT_CTL2	2	B1	i/O (L)	100k PU *1	A18: Address bus (A18) (default) P46: General-purpose I/O port TFT_CTL2: LCDC TFT I/F control signal 2 output
A19 P45 FPDAT11	3	D2	i/O (L)	100k PU *1	A19: Address bus (A19) (default) P45: General-purpose I/O port FPDAT11: LCD data
A20 P44 FPDAT10	4	C2	i/O (L)	100k PU *1	A20: Address bus (A20) (default) P44: General-purpose I/O port FPDAT10: LCD data
A21 P43 FPDAT9	5	C3	i/O (L)	100k PU *1	A21: Address bus (A21) (default) P43: General-purpose I/O port FPDAT9: LCD data

I S1C33E08 SPECIFICATIONS: PIN DESCRIPTION

Pin name	Pin No.		I/O	Pull-up/down	Function
	QFP	PFBGA			
A22 P42 FPDAT8	6	C1	i/O (H)	100k PU *1	A22: Address bus (A22) (default) P42: General-purpose I/O port FPDAT8: LCD data
A23 P41 #SDRAS EXCL3	133	A6	i/O (H)	100k PU *1	A23: Address bus (A23) (default) P41: General-purpose I/O port #SDRAS: SDRAM row address strobe signal output EXCL3: 16-bit timer 3 event counter input
A24 P40 #SDCAS EXCL4	134	C6	i/O (L)	100k PU *1	A24: Address bus (A24) (default) P40: General-purpose I/O port #SDCAS: SDRAM column address strobe signal output EXCL4: 16-bit timer 4 event counter input
#RD	137	A5	O	100k PU *1	Read signal
#WRL	138	C5	O	100k PU *1	Write (low byte) signal
#WRH / #BSH	139	B4	O	100k PU *1	Write (high byte) signal or bus strobe (high byte) signal
#CE10 P57	136	B5	*3	100k PU *1	#CE10: Chip enable signal for areas 10, 13 and 20 (default) P57: General-purpose I/O port
#CE4 P50 CARD0	142	A3	i/O (H)	100k PU *1	#CE4: Chip enable signal for areas 4 and 14 (default) P50: General-purpose I/O port CARD0: Card I/F signal 0 output
#CE5 P51 CARD1	143	B3	i/O (H)	100k PU *1	#CE5: Chip enable signal for areas 5, 15 and 16 (default) P51: General-purpose I/O port CARD1: Card I/F signal 1 output
P52 BCLK #CE6 CMU_CLK	141	C4	I/o (PU)	100k PU *1	P52: General-purpose I/O port (default) BCLK: Bus clock output #CE6: Chip enable signal for areas 6, 17 and 18 CMU_CLK: CMU external clock output
#CE7 P53 SDA10	115	A10	i/O (H)	100k PU *1	#CE7: Chip enable signal for areas 7 and 19 (default) P53: General-purpose I/O port SDA10: SDRAM address bit 10
#CE8 P54 CARD1	144	A2	i/O (H)	100k PU *1	#CE8: Chip enable signal for areas 8 and 21 (default) P54: General-purpose I/O port CARD1: Card I/F signal 1 output
#CE9 P55 CARD0	1	B2	i/O (H)	100k PU *1	#CE9: Chip enable signal for areas 9 and 22 (default) P55: General-purpose I/O port CARD0: Card I/F signal 0 output
#CE11 P56	140	A4	i/O (H)	100k PU *1	#CE11: Chip enable signal for areas 11 and 12 (default) P56: General-purpose I/O port
P20 SDCKE	102	D13	I/o (PU)	100k PU *1	P20: General-purpose I/O port (default) SDCKE: SDRAM clock enable signal output
SDCLK P21	103	D14	I/o (PU)	100k PU *1	SDCLK: SDRAM clock output (default) P21: General-purpose I/O port
P22 #SDCS	104	C14	I/o (PU)	100k PU *1	P22: General-purpose I/O port (default) #SDCS: SDRAM chip enable signal output
P23 #SDRAS TFT_CTL1	106	C13	I/o (PU)	100k PU *1	P23: General-purpose I/O port (default) #SDRAS: SDRAM row address strobe signal output TFT_CTL1: LCDC TFT I/F control signal 1 output
P24 #SDCAS	107	B14	I/o (PU)	100k PU *1	P24: General-purpose I/O port (default) #SDCAS: SDRAM column address strobe signal output
P25 #SDWE	108	B13	I/o (PU)	100k PU *1	P25: General-purpose I/O port (default) #SDWE: SDRAM write signal output
P26 DQML	109	A13	I/o (PU)	100k PU *1	P26: General-purpose I/O port (default) DQML: SDRAM data (low byte) input/output mask signal output
P27 DQMH	110	B12	I/o (PU)	100k PU *1	P27: General-purpose I/O port (default) DQMH: SDRAM data (high byte) input/output mask signal output

Table I.3.2.4 Input/Output Port and Peripheral Circuit Pin List

Pin name	Pin No.		I/O	Pull-up/down	Function
	QFP	PFBGA			
P00 SIN0 #DMAACK2	12	E2	I/o (Hi-Z)	100k PU *1	P00: General-purpose I/O port (default) SIN0: Serial I/F Ch.0 data input #DMAACK2: HSDMA Ch.2 acknowledge signal output
P01 SOUT0 #DMAACK3	13	F2	I/o (Hi-Z)	100k PU *1	P01: General-purpose I/O port (default) SOUT0: Serial I/F Ch.0 data output #DMAACK3: HSDMA Ch.3 acknowledge signal output
P02 #SCLK0 #DMAEND2	14	E3	I/o (Hi-Z)	100k PU *1	P02: General-purpose I/O port (default) #SCLK0: Serial I/F Ch.0 clock input/output #DMAEND2: HSDMA Ch.2 end-of-transfer signal output
P03 #SRDY0 #DMAEND3	15	F1	I/o (Hi-Z)	100k PU *1	P03: General-purpose I/O port (default) #SRDY0: Serial I/F Ch.0 ready input/output #DMAEND3: HSDMA Ch.3 end-of-transfer signal output
P04 SIN1 I2S_SDO	17	G2	I/o (Hi-Z)	100k PU *1	P04: General-purpose I/O port (default) SIN1: Serial I/F Ch.1 data input I2S_SDO: I ² S send data signal
P05 SOUT1 I2S_WS	18	F3	I/o (Hi-Z)	100k PU *1	P05: General-purpose I/O port (default) SOUT1: Serial I/F Ch.1 data output I2S_WS: I ² S word select signal
P06 #SCLK1 I2S_SCK	19	G1	I/o (Hi-Z)	100k PU *1	P06: General-purpose I/O port (default) #SCLK1: Serial I/F Ch.1 clock input/output I2S_SCK: I ² S serial clock signal
P07 #SRDY1 I2S_MCLK	20	G3	I/o (Hi-Z)	100k PU *1	P07: General-purpose I/O port (default) #SRDY1: Serial I/F Ch.1 ready input/output I2S_MCLK: I ² S master clock signal
P10 TM0 SIN0 #DMAEND0	21	H1	I/o (Hi-Z)	100k PU *1	P10: General-purpose I/O port (default) TM0: 16-bit timer 0 output SIN0: Serial I/F Ch.0 data input #DMAEND0: HSDMA Ch.0 end-of-transfer signal output
P11 TM1 SOUT0 #DMAEND1	22	H2	I/o (Hi-Z)	100k PU *1	P11: General-purpose I/O port (default) TM1: 16-bit timer 1 output SOUT0: Serial I/F Ch.0 data output #DMAEND1: HSDMA Ch.1 end-of-transfer signal output
P12 TM2 #SCLK0 #DMAACK0	24	J4	I/o (Hi-Z)	100k PU *1	P12: General-purpose I/O port (default) TM2: 16-bit timer 2 output #SCLK0: Serial I/F Ch.0 clock input/output #DMAACK0: HSDMA Ch.0 acknowledge signal output
P13 TM3 #SRDY0 #DMAACK1	25	J2	I/o (Hi-Z)	100k PU *1	P13: General-purpose I/O port (default) TM3: 16-bit timer 3 output #SRDY0: Serial I/F Ch.0 ready input/output #DMAACK1: HSDMA Ch.1 acknowledge signal output
P14 TM4 SIN1	26	H3	I/o (Hi-Z)	100k PU *1	P14: General-purpose I/O port (default) TM4: 16-bit timer 4 output SIN1: Serial I/F Ch.1 data input
DST0 P15 TM5 SOUT1 TFT_CTL0	27	J1	i/O (H)	100k PU *1	DST0: DST0 signal output for debugging (default) P15: General-purpose I/O port TM5: 16-bit timer 5 output SOUT1: Serial I/F Ch.1 data output TFT_CTL0: LCD C TFT I/F control signal 0 output
DST1 P16 DCSIO0 #SCLK1 TFT_CTL3	28	J3	i/O (H)	100k PU *1	DST1: DST1 signal output for debugging (default) P16: General-purpose I/O port DCSIO0: DCSIO port #SCLK1: Serial I/F Ch.1 clock input/output TFT_CTL3: LCD C TFT I/F control signal 3 output
DPCO P17 DCSIO1 #SRDY1 TFT_CTL2	29	K2	i/O (H)	100k PU *1	DPCO: DPCO signal output for debugging (default) P17: General-purpose I/O port DCSIO1: DCSIO port #SRDY1: Serial I/F Ch.1 ready input/output TFT_CTL2: LCD C TFT I/F control signal 2 output
P30 CARD2 #DMAREQ0	8	D1	I/o (Hi-Z)	100k PU *1	P30: General-purpose I/O port (default) CARD2: Card I/F signal 2 output #DMAREQ0: HSDMA Ch.0 request input
P31 CARD3 #DMAREQ1	9	D3	I/o (Hi-Z)	100k PU *1	P31: General-purpose I/O port (default) CARD3: Card I/F signal 3 output #DMAREQ1: HSDMA Ch.1 request input
P32 CARD4 #DMAREQ2	10	E4	I/o (Hi-Z)	100k PU *1	P32: General-purpose I/O port (default) CARD4: Card I/F signal 4 output #DMAREQ2: HSDMA Ch.2 request input

I S1C33E08 SPECIFICATIONS: PIN DESCRIPTION

Pin name	Pin No.		I/O	Pull-up/down	Function
	QFP	PFBGA			
P33 CARD5 #DMAREQ3	11	E1	I/o (Hi-Z)	100k PU *1	P33: General-purpose I/O port (default) CARD5: Card I/F signal 5 output #DMAREQ3: HSDMA Ch.3 request input
P60 SIN2 DCSIO0 EXCL0	77	L13	I/o (Hi-Z)	100k PU *1	P60: General-purpose I/O port (default) SIN2: Serial I/F Ch.2 data input DCSIO0: DCSIO port EXCL0: 16-bit timer 0 event counter input
P61 SOUT2 DCSIO1 EXCL1	78	K12	I/o (Hi-Z)	100k PU *1	P61: General-purpose I/O port (default) SOUT2: Serial I/F Ch.2 data output DCSIO1: DCSIO port EXCL1: 16-bit timer 1 event counter input
P62 #SCLK2 #ADTRG CMU_CLK	79	J12	I/o (Hi-Z)	100k PU *1	P62: General-purpose I/O port (default) #SCLK2: Serial I/F Ch.2 clock input/output #ADTRG: A/D converter trigger input CMU_CLK: CMU external clock output
P63 #SRDY2 WDT_CLK #WDT_NMI	80	K14	I/o (Hi-Z)	100k PU *1	P63: General-purpose I/O port (default) #SRDY2: Serial I/F Ch.2 ready input/output WDT_CLK: Watchdog timer clock output #WDT_NMI: Watchdog timer NMI signal output
P64 #WAIT EXCL2	34	K3	I/o (Hi-Z)	100k PU *1	P64: General-purpose I/O port (default) #WAIT: Wait cycle request input EXCL2: 16-bit timer 2 event counter input
P65 SDI FPDAT8	35	M2	I/o (Hi-Z)	100k PU *1	P65: General-purpose I/O port (default) SDI: SPI data input FPDAT8: LCD data
P66 SDO FPDAT9	36	M3	I/o (Hi-Z)	100k PU *1	P66: General-purpose I/O port (default) SDO: SPI data output FPDAT9: LCD data
P67 SPL_CLK FPDAT10	37	N1	I/o (Hi-Z)	100k PU *1	P67: General-purpose I/O port (default) SPL_CLK: SPI clock FPDAT10: LCD data
P70 AIN0	52	N6	I (Hi-Z)	100k PU *1	P70: General-purpose I/O port (default) AIN0: A/D converter Ch.0 input
P71 AIN1	51	M6	I (Hi-Z)	100k PU *1	P71: General-purpose I/O port (default) AIN1: A/D converter Ch.1 input
P72 AIN2	50	N7	I (Hi-Z)	100k PU *1	P72: General-purpose I/O port (default) AIN2: A/D converter Ch.2 input
P73 AIN3	49	M5	I (Hi-Z)	100k PU *1	P73: General-purpose I/O port (default) AIN3: A/D converter Ch.3 input
P74 AIN4 EXCL5	48	N5	I (Hi-Z)	100k PU *1	P74: General-purpose I/O port (default) AIN4: A/D converter Ch.4 input EXCL5: 16-bit timer 5 event counter input
P80 FPFRAME	53	P7	I/o (Hi-Z)	100k PU *1	P80: General-purpose I/O port (default) FPFRAME: LCD frame clock output
P81 FPLINE	54	L6	I/o (Hi-Z)	100k PU *1	P81: General-purpose I/O port (default) FPLINE: LCD line clock output
P82 FPSHIFT	55	L7	I/o (Hi-Z)	100k PU *1	P82: General-purpose I/O port (default) FPSHIFT: LCD shift clock output
P83 FPDRDY TFT_CTL1 BCLK	56	M7	I/o (Hi-Z)	100k PU *1	P83: General-purpose I/O port (default) FPDRDY: LCD DRDY/MOD signal output TFT_CTL1: LCDC TFT I/F control signal 1 output BCLK: Bus clock output
P84 DCSIO0 FPDAT11	57	L8	I/o (Hi-Z)	100k PU *1	P84: General-purpose I/O port (default) DCSIO0: DCSIO port FPDAT11: LCD data
P85 DCSIO1	58	M8	I/o (Hi-Z)	100k PU *1	P85: General-purpose I/O port (default) DCSIO1: DCSIO port
P90 FPDAT0	63	L10	I/o (Hi-Z)	100k PU *1	P90: General-purpose I/O port (default) FPDAT0: LCD data
P91 FPDAT1	64	L11	I/o (Hi-Z)	100k PU *1	P91: General-purpose I/O port (default) FPDAT1: LCD data
P92 FPDAT2	65	P10	I/o (Hi-Z)	100k PU *1	P92: General-purpose I/O port (default) FPDAT2: LCD data
P93 FPDAT3	66	M10	I/o (Hi-Z)	100k PU *1	P93: General-purpose I/O port (default) FPDAT3: LCD data
P94 FPDAT4	68	M11	I/o (Hi-Z)	100k PU *1	P94: General-purpose I/O port (default) FPDAT4: LCD data

Pin name	Pin No.		I/O	Pull-up/down	Function
	QFP	PFBGA			
P95 FPDAT5	69	N10	I/o (Hi-Z)	100k PU *1	P95: General-purpose I/O port (default) FPDAT5: LCD data
P96 FPDAT6	70	N12	I/o (Hi-Z)	100k PU *1	P96: General-purpose I/O port (default) FPDAT6: LCD data
P97 FPDAT7	71	M12	I/o (Hi-Z)	100k PU *1	P97: General-purpose I/O port (default) FPDAT7: LCD data
PA0 TFT_CTL0	–	P11	I/o (PU)	100k PU *1	PA0: Extended general-purpose I/O port (default) TFT_CTL0: LCDDC TFT I/F control signal 0 output
PA1 FPDAT8 TFT_CTL0	–	N11	I/o (PU)	100k PU *1	PA1: Extended general-purpose I/O port (default) FPDAT8: LCD data TFT_CTL0: LCDDC TFT I/F control signal 0 output
PA2 FPDAT9 TFT_CTL1	–	L12	I/o (PU)	100k PU *1	PA2: Extended general-purpose I/O port (default) FPDAT9: LCD data TFT_CTL1: LCDDC TFT I/F control signal 1 output
PA3 FPDAT10 TFT_CTL2 CARD0	–	P12	I/o (PU)	100k PU *1	PA3: Extended general-purpose I/O port (default) FPDAT10: LCD data TFT_CTL2: LCDDC TFT I/F control signal 2 output CARD0: Card I/F signal 0 output
PA4 FPDAT11 TFT_CTL3 CARD1	–	P13	I/o (PU)	100k PU *1	PA4: Extended general-purpose I/O port (default) FPDAT11: LCD data TFT_CTL3: LCDDC TFT I/F control signal 3 output CARD1: Card I/F signal 1 output
PB0 FPDAT8 I2S_SDO CARD2	–	L2	I/o (PU)	100k PU *1	PB0: Extended general-purpose I/O port (default) FPDAT8: LCD data I2S_SDO: I ² S send data signal CARD2: Card I/F signal 2 output
PB1 FPDAT9 I2S_WS CARD3	–	L4	I/o (PU)	100k PU *1	PB1: Extended general-purpose I/O port (default) FPDAT9: LCD data I2S_WS: I ² S word select signal CARD3: Card I/F signal 3 output
PB2 FPDAT10 I2S_SCK CARD4	–	L3	I/o (PU)	100k PU *1	PB2: Extended general-purpose I/O port (default) FPDAT10: LCD data I2S_SCK: I ² S serial clock signal CARD4: Card I/F signal 4 output
PB3 FPDAT11 I2S_MCLK CARD5	–	M1	I/o (PU)	100k PU *1	PB3: Extended general-purpose I/O port (default) FPDAT11: LCD data I2S_MCLK: I ² S master clock signal CARD5: Card I/F signal 5 output

Table I.3.2.5 USB Interface Pin List

Pin name	Pin No.		I/O	Pull-up/down	Function
	QFP	PFBGA			
USBDP	75	M14	I/o	–	USB D+ pin
USBDM	74	N14	I/o	–	USB D- pin
USBVBUS	76	L14	I	–	USB VBUS pin. Allows input of 5 V

Table I.3.2.6 Other Pin List

Pin name	Pin No.		I/O	Pull-up/down	Function
	QFP	PFBGA			
#RESET	72	N13	I	50k PU	Initial reset input pin (with noise reduction circuit)
#NMI	73	M13	I	50k PU	NMI request input pin (with noise reduction circuit)
DSIO P34	31	K1	I/o	50k PU	DSIO: Serial input/output for debugging (with noise reduction circuit) (default) P34: General-purpose I/O port
DCLK P35	33	L1	i/O (H)	50k PU	DCLK: DCLK signal output for debugging (default) P35: General-purpose I/O port
DST2 P36	32	K4	i/O (L)	100k PU	DST2: DST2 signal output for debugging (default) P36: General-purpose I/O port
BOOT1	46	L5	I	–	Boot mode select signal 1 input
BOOT0	–	K13	I	–	Boot mode select signal 0 input
BURNIN	–	M9	I	50k PD	Wafer level burn-in test enable input
TEST0	45	M4	I	60k PD	TEST-0 input

I S1C33E08 SPECIFICATIONS: PIN DESCRIPTION

*1: These pins can have pull-ups enabled or disabled by setting the pin control registers.

*2: These pins come with a bus hold latch.

*3: The input/output direction of the #CE10 pin at initial reset depends on the configuration of the BOOT[1:0] pins. Refer to “Appendix D Boot” for details.

- Notes:**
- The # prefixed to pin names indicates that input/output signals of the pin are active low.
 - The pin names listed in boldface denote the default pin (signal) name.
 - The I/O listed in boldface and uppercase denote the default input/output direction.
() for I/O indicates the following pin states:
(H), (L): Default output level. This is only indicated for signals whose level is fixed high or low when the chip is initially reset.
(PU): The pin is pulled up at initial reset (register control pull-up is enabled).
(Hi-Z): The pin is placed in high impedance state at initial reset (register control pull-up is disabled).
 - The input level must be V_{DD} only for the MCLKI and RTC_CLKI pins. Input levels for other pins should be V_{DDH} (or AV_{DD}) level.
 - The PA* and PB* port pins are not available in the QFP package. Do not set these ports to a condition (input mode and pull-up off) that may place the port into floating status.
 - The BOOT0 pin is not available in the QFP package. The BOOT0 signal is pulled down to low inside the package.

I.3.3 Switching Over the Multiplexed Pin Functions

I

I.3.3.1 Pin Function Select Bits

Pin

Each pin is assigned one to four functions, as listed in Table I.3.3.1.1.

When the chip is powered on or cold-reset, each pin defaults to function 0 except for the P21/SDCLK pin and debug pins. If any pin must be used for other than this default function, select the desired function by writing data to the corresponding pin function select bits.

Table I.3.3.1.1 List of Pin Function Select Bits

Pin function 0	Pin function 1	Pin function 2	Pin function 3	Debug function	Function select bit
MCLKI					
MCLKO					
RTC_CLKI					
RTC_CLKO					
VCP					
#RESET					
#NMI					
BOOT1					
BOOT0					
BURNIN					
DSIO	P34				CFP34[1:0] (D[1:0]/0x3003A7)
DCLK	P35				CFP35[1:0] (D[3:2]/0x3003A7)
DST2	P36				CFP36[1:0] (D[5:4]/0x3003A7)
D0					
D1					
D2					
D3					
D4					
D5					
D6					
D7					
D8	PC0				CFPC0[1:0] (D[1:0]/0x300C24)
D9	PC1				CFPC1[1:0] (D[3:2]/0x300C24)
D10	PC2				CFPC2[1:0] (D[5:4]/0x300C24)
D11	PC3				CFPC3[1:0] (D[7:6]/0x300C24)
D12	PC4				CFPC4[1:0] (D[1:0]/0x300C25)
D13	PC5				CFPC5[1:0] (D[3:2]/0x300C25)
D14	PC6				CFPC6[1:0] (D[5:4]/0x300C25)
D15	PC7				CFPC7[1:0] (D[7:6]/0x300C25)
A0/#BSL					
A1					
A2					
A3					
A4					
A5					
A6					
A7					
A8					
A9					
A10					
A11	P47				CFP47[1:0] (D[7:6]/0x3003A9)
A12					
A13					
A14					
A15					
A16/DQML					
A17/DQMH					
A18	P46	TFT_CTL2			CFP46[1:0] (D[5:4]/0x3003A9)
A19	P45	FPDAT11			CFP45[1:0] (D[3:2]/0x3003A9)
A20	P44	FPDAT10			CFP44[1:0] (D[1:0]/0x3003A9)

I S1C33E08 SPECIFICATIONS: PIN DESCRIPTION

Pin function 0	Pin function 1	Pin function 2	Pin function 3	Debug function	Function select bit
A21	P43	FPDAT9			CFP43[1:0] (D[7:6]/0x3003A8)
A22	P42	FPDAT8			CFP42[1:0] (D[5:4]/0x3003A8)
A23	P41	#SDRAS	EXCL3		CFP41[1:0] (D[3:2]/0x3003A8)
A24	P40	#SDCAS	EXCL4		CFP40[1:0] (D[1:0]/0x3003A8)
#RD					
#WRL					
#WRH/#BSH					
#CE10	P57				CFP57[1:0] (D[7:6]/0x3003AB)
#CE4	P50	CARD0			CFP50[1:0] (D[1:0]/0x3003AA)
#CE5	P51	CARD1			CFP51[1:0] (D[3:2]/0x3003AA)
P52	BCLK	#CE6	CMU_CLK		CFP52[1:0] (D[5:4]/0x3003AA)
#CE7	P53	SDA10			CFP53[1:0] (D[7:6]/0x3003AA)
#CE8	P54	CARD1			CFP54[1:0] (D[1:0]/0x3003AB)
#CE9	P55	CARD0			CFP55[1:0] (D[3:2]/0x3003AB)
#CE11	P56				CFP56[1:0] (D[5:4]/0x3003AB)
P00	SIN0	#DMAACK2			CFP00[1:0] (D[1:0]/0x3003A0)
P01	SOUT0	#DMAACK3			CFP01[1:0] (D[3:2]/0x3003A0)
P02	#SCLK0	#DMAEND2			CFP02[1:0] (D[5:4]/0x3003A0)
P03	#SRDY0	#DMAEND3			CFP03[1:0] (D[7:6]/0x3003A0)
P04	SIN1	I2S_SDO			CFP04[1:0] (D[1:0]/0x3003A1)
P05	SOUT1	I2S_WS			CFP05[1:0] (D[3:2]/0x3003A1)
P06	#SCLK1	I2S_SCK			CFP06[1:0] (D[5:4]/0x3003A1)
P07	#SRDY1	I2S_MCLK			CFP07[1:0] (D[7:6]/0x3003A1)
P10	TM0	SIN0	#DMAEND0		CFP10[1:0] (D[1:0]/0x3003A2)
P11	TM1	SOUT0	#DMAEND1		CFP11[1:0] (D[3:2]/0x3003A2)
P12	TM2	#SCLK0	#DMAACK0		CFP12[1:0] (D[5:4]/0x3003A2)
P13	TM3	#SRDY0	#DMAACK1		CFP13[1:0] (D[7:6]/0x3003A2)
P14	TM4	SIN1			CFP14[1:0] (D[1:0]/0x3003A3)
P15	TM5	SOUT1	TFT_CTL0	DST0	CFP15[1:0] (D[3:2]/0x3003A3)
P16	DCSIO0	#SCLK1	TFT_CTL3	DST1	CFP16[1:0] (D[5:4]/0x3003A3)
P17	DCSIO1	#SRDY1	TFT_CTL2	DPCO	CFP17[1:0] (D[7:6]/0x3003A3)
P20	SDCKE				CFP20[1:0] (D[1:0]/0x3003A4)
P21	SDCLK				CFP21[1:0] (D[3:2]/0x3003A4)
P22	#SDCS				CFP22[1:0] (D[5:4]/0x3003A4)
P23	#SDRAS	TFT_CTL1			CFP23[1:0] (D[7:6]/0x3003A4)
P24	#SDCAS				CFP24[1:0] (D[1:0]/0x3003A5)
P25	#SDWE				CFP25[1:0] (D[3:2]/0x3003A5)
P26	DQML				CFP26[1:0] (D[5:4]/0x3003A5)
P27	DQMH				CFP27[1:0] (D[7:6]/0x3003A5)
P30	CARD2	#DMAREQ0			CFP30[1:0] (D[1:0]/0x3003A6)
P31	CARD3	#DMAREQ1			CFP31[1:0] (D[3:2]/0x3003A6)
P32	CARD4	#DMAREQ2			CFP32[1:0] (D[5:4]/0x3003A6)
P33	CARD5	#DMAREQ3			CFP33[1:0] (D[7:6]/0x3003A6)
P60	SIN2	DCSIO0	EXCL0		CFP60[1:0] (D[1:0]/0x3003AC)
P61	SOUT2	DCSIO1	EXCL1		CFP61[1:0] (D[3:2]/0x3003AC)
P62	#SCLK2	#ADTRG	CMU_CLK		CFP62[1:0] (D[5:4]/0x3003AC)
P63	#SRDY2	WDT_CLK	#WDT_NMI		CFP63[1:0] (D[7:6]/0x3003AC)
P64	#WAIT	EXCL2			CFP64[1:0] (D[1:0]/0x3003AD)
P65	SDI	FPDAT8			CFP65[1:0] (D[3:2]/0x3003AD)
P66	SDO	FPDAT9			CFP66[1:0] (D[5:4]/0x3003AD)
P67	SPI_CLK	FPDAT10			CFP67[1:0] (D[7:6]/0x3003AD)
P70	AIN0				CFP70[1:0] (D[1:0]/0x3003AE)
P71	AIN1				CFP71[1:0] (D[3:2]/0x3003AE)
P72	AIN2				CFP72[1:0] (D[5:4]/0x3003AE)
P73	AIN3				CFP73[1:0] (D[7:6]/0x3003AE)
P74	AIN4	EXCL5			CFP74[1:0] (D[1:0]/0x3003AF)
P80	FPFRAME				CFP80[1:0] (D[1:0]/0x3003B0)
P81	FPLINE				CFP81[1:0] (D[3:2]/0x3003B0)
P82	FPSHIFT				CFP82[1:0] (D[5:4]/0x3003B0)
P83	FPDRDY	TFT_CTL1	BCLK		CFP83[1:0] (D[7:6]/0x3003B0)
P84	DCSIO0	FPDAT11			CFP84[1:0] (D[1:0]/0x3003B1)
P85	DCSIO1				CFP85[1:0] (D[3:2]/0x3003B1)

Pin function 0	Pin function 1	Pin function 2	Pin function 3	Debug function	Function select bit
P90	FPDAT0				CFP90[1:0] (D[1:6]/0x3003B2)
P91	FPDAT1				CFP91[1:0] (D[3:2]/0x3003B2)
P92	FPDAT2				CFP92[1:0] (D[5:4]/0x3003B2)
P93	FPDAT3				CFP93[1:0] (D[7:6]/0x3003B2)
P94	FPDAT4				CFP94[1:0] (D[1:0]/0x3003B3)
P95	FPDAT5				CFP95[1:0] (D[3:2]/0x3003B3)
P96	FPDAT6				CFP96[1:0] (D[5:4]/0x3003B3)
P97	FPDAT7				CFP97[1:0] (D[7:6]/0x3003B3)
PA0	TFT_CTL0				CFPA0[1:0] (D[1:0]/0x300C20)
PA1	FPDAT8	TFT_CTL0			CFPA1[1:0] (D[3:2]/0x300C20)
PA2	FPDAT9	TFT_CTL1			CFPA2[1:0] (D[5:4]/0x300C20)
PA3	FPDAT10	TFT_CTL2	CARD0		CFPA3[1:0] (D[7:6]/0x300C20)
PA4	FPDAT11	TFT_CTL3	CARD1		CFPA4[1:0] (D[1:0]/0x300C21)
PB0	FPDAT8	I2S_SDO	CARD2		CFPB0[1:0] (D[1:0]/0x300C22)
PB1	FPDAT9	I2S_WS	CARD3		CFPB1[1:0] (D[3:2]/0x300C22)
PB2	FPDAT10	I2S_SCK	CARD4		CFPB2[1:0] (D[5:4]/0x300C22)
PB3	FPDAT11	I2S_MCLK	CARD5		CFPB3[1:0] (D[7:6]/0x300C22)
USBDP					
USBDM					
USBVBUS					
TEST0					

- The set values 0 to 3 of the pin function select bits correspond to functions 0 to 3, respectively.
- When TRCMUX (D0/0x300014) = 1 (default), the P15–P17 and P34–P36 pins are configured as debug-only pins and the function select bits are ineffective. To use the pin function other than debugging, set TRCMUX (D0/0x300014) to 0 before setting the function select bit.

* **TRCMUX**: P15–17, P34–36 Debug Function Select Bit in the Debug Port MUX Register (D0/0x300014)

- CARD0 to CARD5 are the output pins for card interfaces. The functions of respective pins can be selected according to the card interface used, as listed in Table I.3.3.1.2. Use the Card I/F Output Port Configuration Register (0x300302) to select the functions of these pins. For details of the card interfaces and output signals, see Section V.4, “Card Interface (CARD).”

Table I.3.3.1.2 Relationship between Ports and Card Interface Signals

Pin name	Function 0 (default)	Function 1	Function select bit
CARD0	#SMRD	#CFCE1	CARDIO0 (D0/0x300302)
CARD1	#SMWR	#CFCE2	CARDIO1 (D1/0x300302)
CARD2	#IORD	#SMRD	CARDIO2 (D2/0x300302)
CARD3	#IOWR	#SMWR	CARDIO3 (D3/0x300302)
CARD4	#OE	#CFCE1	CARDIO4 (D4/0x300302)
CARD5	#WE	#CFCE2	CARDIO5 (D5/0x300302)

#SMRD, #SMWR: Output pins for SmartMedia (NAND flash)

#CFCE1, #CFCE2: Output pins for CompactFlash

#IORD, #IOWR, #OE, #WE: Output pins for PC Card

* **CARDIOx**: CARDx Port Function Select Bit in the Card I/F Output Port Configuration Register (Dx/0x300302)

I.3.3.2 List of Port Function Select Registers

Table I.3.3.2.1 List of Port Function Select Registers

Address	Register name	Size	Function
0x003003A0	P00–P03 Port Function Select Register (pP0_03_CFP)	8	Selects P00–P03 port pin functions.
0x003003A1	P04–P07 Port Function Select Register (pP0_47_CFP)	8	Selects P04–P07 port pin functions.
0x003003A2	P10–P13 Port Function Select Register (pP1_03_CFP)	8	Selects P10–P13 port pin functions.
0x003003A3	P14–P17 Port Function Select Register (pP1_47_CFP)	8	Selects P14–P17 port pin functions.
0x003003A4	P20–P23 Port Function Select Register (pP2_03_CFP)	8	Selects P20–P23 port pin functions.
0x003003A5	P24–P27 Port Function Select Register (pP2_47_CFP)	8	Selects P24–P27 port pin functions.
0x003003A6	P30–P33 Port Function Select Register (pP3_03_CFP)	8	Selects P30–P33 port pin functions.
0x003003A7	P34–P36 Port Function Select Register (pP3_46_CFP)	8	Selects P34–P36 port pin functions.
0x003003A8	P40–P43 Port Function Select Register (pP4_03_CFP)	8	Selects P40–P43 port pin functions.
0x003003A9	P44–P47 Port Function Select Register (pP4_47_CFP)	8	Selects P44–P47 port pin functions.
0x003003AA	P50–P53 Port Function Select Register (pP5_03_CFP)	8	Selects P50–P53 port pin functions.
0x003003AB	P54–P57 Port Function Select Register (pP5_47_CFP)	8	Selects P54–P57 port pin functions.
0x003003AC	P60–P63 Port Function Select Register (pP6_03_CFP)	8	Selects P60–P63 port pin functions.
0x003003AD	P64–P67 Port Function Select Register (pP6_47_CFP)	8	Selects P64–P67 port pin functions.
0x003003AE	P70–P73 Port Function Select Register (pP7_03_CFP)	8	Selects P70–P73 port pin functions.
0x003003AF	P74 Port Function Select Register (pP7_4_CFP)	8	Selects P74 port pin function.
0x003003B0	P80–P83 Port Function Select Register (pP8_03_CFP)	8	Selects P80–P83 port pin functions.
0x003003B1	P84–P85 Port Function Select Register (pP8_45_CFP)	8	Selects P84–P85 port pin functions.
0x003003B2	P90–P93 Port Function Select Register (pP9_03_CFP)	8	Selects P90–P93 port pin functions.
0x003003B3	P94–P97 Port Function Select Register (pP9_47_CFP)	8	Selects P94–P97 port pin functions.
0x00300C20	PA0–PA3 Port Function Select Register (pPA_03_CFP)	8	Selects PA0–PA3 port pin functions. *
0x00300C21	PA4 Port Function Select Register (pPA_4_CFP)	8	Selects PA4 port pin function. *
0x00300C22	PB0–PB3 Port Function Select Register (pPB_03_CFP)	8	Selects PB0–PB3 port pin functions. *
0x00300C24	PC0–PC3 Port Function Select Register (pPC_03_CFP)	8	Selects PC0–PC3 port pin functions.
0x00300C25	PC4–PC7 Port Function Select Register (pPC_47_CFP)	8	Selects PC4–PC7 port pin functions.

* These ports are not available in the QFP24-144pin model.

The following describes each port function select register.

The port function select registers are mapped to addresses 0x3003A0 to 0x3003B3 and 0x300C20 to 0x300C25, and can be accessed in units of bytes.

0x3003A0: P00–P03 Port Function Select Register (pP0_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P00–P03 port function select register (pP0_03_CFP)	003003A0 (B)	D7	CFP031	P03 port extended function	CFP03[1:0]	Function	0	R/W
		D6	CFP030		11	reserved	0	
					10	#DMAEND3		
					01	#SRDY0		
					00	P03		
		D5	CFP021	P02 port extended function	CFP02[1:0]	Function	0	R/W
		D4	CFP020		11	reserved	0	
					10	#DMAEND2		
					01	#SCLK0		
					00	P02		
		D3	CFP011	P01 port extended function	CFP01[1:0]	Function	0	R/W
		D2	CFP010		11	reserved	0	
					10	#DMAACK3		
					01	SOUT0		
					00	P01		
		D1	CFP001	P00 port extended function	CFP00[1:0]	Function	0	R/W
D0	CFP000	11	reserved		0			
		10	#DMAACK2					
		01	SIN0					
		00	P00					

This register selects the functions of ports P00 to P03.

D[7:6] CFP03[1:0]: P03 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): #DMAEND3
- 01 (R/W): #SRDY0
- 00 (R/W): P03 (default)

D[5:4] CFP02[1:0]: P02 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): #DMAEND2
- 01 (R/W): #SCLK0
- 00 (R/W): P02 (default)

D[3:2] CFP01[1:0]: P01 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): #DMAACK3
- 01 (R/W): SOUT0
- 00 (R/W): P01 (default)

D[1:0] CFP00[1:0]: P00 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): #DMAACK2
- 01 (R/W): SIN0
- 00 (R/W): P00 (default)

0x3003A1: P04–P07 Port Function Select Register (pP0_47_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
P04–P07 port function select register (pP0_47_CFP)	003003A1 (B)	D7 D6	CFP071	P07 port extended function	CFP07[1:0]	Function	0	R/W		
			CFP070		11	reserved	0			
							10	I2S_MCLK		
							01	#SRDY1		
							00	P07		
				D5 D4	CFP061	P06 port extended function	CFP06[1:0]	Function	0	R/W
		CFP060	11		reserved		0			
							10	I2S_SCK		
							01	#SCLK1		
							00	P06		
				D3 D2	CFP051	P05 port extended function	CFP05[1:0]	Function	0	R/W
		CFP050	11		reserved		0			
							10	I2S_WS		
							01	SOUT1		
							00	P05		
				D1 D0	CFP041	P04 port extended function	CFP04[1:0]	Function	0	R/W
CFP040	11	reserved	0							
					10	I2S_SDO				
					01	SIN1				
					00	P04				

This register selects the functions of ports P04 to P07.

D[7:6] CFP07[1:0]: P07 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): I2S_MCLK
- 01 (R/W): #SRDY1
- 00 (R/W): P07 (default)

D[5:4] CFP06[1:0]: P06 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): I2S_SCK
- 01 (R/W): #SCLK1
- 00 (R/W): P06 (default)

D[3:2] CFP05[1:0]: P05 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): I2S_WS
- 01 (R/W): SOUT1
- 00 (R/W): P05 (default)

D[1:0] CFP04[1:0]: P04 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): I2S_SDO
- 01 (R/W): SIN1
- 00 (R/W): P04 (default)

0x3003A2: P10–P13 Port Function Select Register (pP1_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P10–P13 port function select register (pP1_03_CFP)	003003A2 (B)	D7	CFP131	P13 port extended function	CFP13[1:0]	Function	0	R/W
		D6	CFP130		11	#DMAACK1	0	
					10	#SRDY0		
					01	TM3		
					00	P13		
		D5	CFP121	P12 port extended function	CFP12[1:0]	Function	0	R/W
		D4	CFP120		11	#DMAACK0	0	
					10	#SCLK0		
					01	TM2		
					00	P12		
		D3	CFP111	P11 port extended function	CFP11[1:0]	Function	0	R/W
		D2	CFP110		11	#DMAEND1	0	
					10	SOUT0		
					01	TM1		
					00	P11		
		D1	CFP101	P10 port extended function	CFP10[1:0]	Function	0	R/W
D0	CFP100	11	#DMAEND0		0			
		10	SIN0					
		01	TM0					
		00	P10					

This register selects the functions of ports P10 to P13.

D[7:6] CFP13[1:0]: P13 Port Extended Function Select Bits

- 11 (R/W): #DMAACK1
- 10 (R/W): #SRDY0
- 01 (R/W): TM3
- 00 (R/W): P13 (default)

D[5:4] CFP12[1:0]: P12 Port Extended Function Select Bits

- 11 (R/W): #DMAACK0
- 10 (R/W): #SCLK0
- 01 (R/W): TM2
- 00 (R/W): P12 (default)

D[3:2] CFP11[1:0]: P11 Port Extended Function Select Bits

- 11 (R/W): #DMAEND1
- 10 (R/W): SOUT0
- 01 (R/W): TM1
- 00 (R/W): P11 (default)

D[1:0] CFP10[1:0]: P10 Port Extended Function Select Bits

- 11 (R/W): #DMAEND0
- 10 (R/W): SIN0
- 01 (R/W): TM0
- 00 (R/W): P10 (default)

0x3003A3: P14–P17 Port Function Select Register (pP1_47_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P14–P17 port function select register (pP1_47_CFP)	003003A3 (B)	D7 D6	CFP171 CFP170	P17 port extended function	CFP17[1:0]	Function	0	R/W	When TRCMUX (D0/0x300014) is set to 1 (default), this register becomes ineffective and the port is configured for debugging.
					11	TFT_CTL2	0		
					10	#SRDY1			
					01	DCSIO1			
		D5 D4	CFP161 CFP160	P16 port extended function	CFP16[1:0]	Function	0	R/W	
					11	TFT_CTL3	0		
					10	#SCLK1			
					01	DCSIO0			
		D3 D2	CFP151 CFP150	P15 port extended function	CFP15[1:0]	Function	0	R/W	
					11	TFT_CTL0	0		
					10	SOUT1			
					01	TM5			
		D1 D0	CFP141 CFP140	P14 port extended function	CFP14[1:0]	Function	0	R/W	
					11	reserved			
					10	SIN1			
					01	TM4			
					00				

This register selects the functions of ports P14 to P17.

D[7:6] CFP17[1:0]: P17 Port Extended Function Select Bits

11 (R/W): TFT_CTL2

10 (R/W): #SRDY1

01 (R/W): DCSIO1

00 (R/W): P17 (default)

When TRCMUX (D0/0x300014) = 1 (default), these control bits are ineffective and the P17 is configured as the DPCO pin for debugging.

* **TRCMUX**: P15–17, P34–36 Debug Function Select Bit in the Debug Port MUX Register (D0/0x300014)

D[5:4] CFP16[1:0]: P16 Port Extended Function Select Bits

11 (R/W): TFT_CTL3

10 (R/W): #SCLK1

01 (R/W): DCSIO0

00 (R/W): P16 (default)

When TRCMUX (D0/0x300014) = 1 (default), these control bits are ineffective and the P16 is configured as the DST1 pin for debugging.

D[3:2] CFP15[1:0]: P15 Port Extended Function Select Bits

11 (R/W): TFT_CTL0

10 (R/W): SOUT1

01 (R/W): TM5

00 (R/W): P15 (default)

When TRCMUX (D0/0x300014) = 1 (default), these control bits are ineffective and the P15 is configured as the DST0 pin for debugging.

D[1:0] CFP14[1:0]: P14 Port Extended Function Select Bits

11 (R/W): Reserved

10 (R/W): SIN1

01 (R/W): TM4

00 (R/W): P14 (default)

0x3003A4: P20–P23 Port Function Select Register (pP2_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P20–P23 port function select register (pP2_03_CFP)	003003A4 (B)	D7	CFP231	P23 port extended function	CFP23[1:0]	Function	0	R/W	
		D6	CFP230		11	reserved	0		
					10	TFT_CTL1			
					01	#SDRAS			
				00	P23				
		D5	CFP221	P22 port extended function	CFP22[1:0]	Function	0	R/W	
		D4	CFP220		1*	reserved	0		
					01	#SDCS			
					00	P22			
		D3	CFP211	P21 port extended function	CFP21[1:0]	Function	0	R/W	
		D2	CFP210		1*	reserved	1		
					01	SDCLK			
		00	P21						
D1	CFP201	P20 port extended function	CFP20[1:0]	Function	0	R/W			
D0	CFP200		1*	reserved	0				
			01	SDCKE					
			00	P20					

This register selects the functions of ports P20 to P23.

D[7:6] CFP23[1:0]: P23 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): TFT_CTL1
- 01 (R/W): #SDRAS
- 00 (R/W): P23 (default)

D[5:4] CFP22[1:0]: P22 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): #SDCS
- 00 (R/W): P22 (default)

D[3:2] CFP21[1:0]: P21 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): SDCLK (default)
- 00 (R/W): P21

D[1:0] CFP20[1:0]: P20 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): SDCKE
- 00 (R/W): P20 (default)

0x3003A5: P24–P27 Port Function Select Register (pP2_47_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P24–P27 port function select register (pP2_47_CFP)	003003A5 (B)	D7	CFP271	P27 port extended function	CFP27[1:0]	Function	0	R/W
		D6	CFP270		1*	reserved	0	
					01	DQMH		
					00	P27		
		D5	CFP261	P26 port extended function	CFP26[1:0]	Function	0	R/W
		D4	CFP260		1*	reserved	0	
					01	DQML		
					00	P26		
		D3	CFP251	P25 port extended function	CFP25[1:0]	Function	0	R/W
		D2	CFP250		1*	reserved	0	
					01	#SDWE		
					00	P25		
D1	CFP241	P24 port extended function	CFP24[1:0]	Function	0	R/W		
D0	CFP240		1*	reserved	0			
			01	#SDCAS				
			00	P24				

This register selects the functions of ports P24 to P27.

D[7:6] CFP27[1:0]: P27 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): DQMH
- 00 (R/W): P27 (default)

D[5:4] CFP26[1:0]: P26 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): DQML
- 00 (R/W): P26 (default)

D[3:2] CFP25[1:0]: P25 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): #SDWE
- 00 (R/W): P25 (default)

D[1:0] CFP24[1:0]: P24 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): #SDCAS
- 00 (R/W): P24 (default)

0x3003A6: P30–P33 Port Function Select Register (pP3_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P30–P33 port function select register (pP3_03_CFP)	003003A6 (B)	D7	CFP331	P33 port extended function	CFP33[1:0]	Function	0	R/W
		D6	CFP330		11	reserved	0	
					10	#DMAREQ3		
					01	CARD5		
					00	P33		
		D5	CFP321	P32 port extended function	CFP32[1:0]	Function	0	R/W
		D4	CFP320		11	reserved	0	
					10	#DMAREQ2		
					01	CARD4		
					00	P32		
		D3	CFP311	P31 port extended function	CFP31[1:0]	Function	0	R/W
		D2	CFP310		11	reserved	0	
					10	#DMAREQ1		
					01	CARD3		
					00	P31		
		D1	CFP301	P30 port extended function	CFP30[1:0]	Function	0	R/W
D0	CFP300	11	reserved		0			
		10	#DMAREQ0					
		01	CARD2					
		00	P30					

This register selects the functions of ports P30 to P33.

D[7:6] CFP33[1:0]: P33 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): #DMAREQ3
- 01 (R/W): CARD5
- 00 (R/W): P33 (default)

D[5:4] CFP32[1:0]: P32 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): #DMAREQ2
- 01 (R/W): CARD4
- 00 (R/W): P32 (default)

D[3:2] CFP31[1:0]: P31 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): #DMAREQ1
- 01 (R/W): CARD3
- 00 (R/W): P31 (default)

D[1:0] CFP30[1:0]: P30 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): #DMAREQ0
- 01 (R/W): CARD2
- 00 (R/W): P30 (default)

0x3003A7: P34–P36 Port Function Select Register (pP3_46_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
P34–P36 port function select register (pP3_46_CFP)	003003A7 (B)	D7–6	–	–	reserved	–	–	–	0 when being read.	
		D5 D4	CFP361 CFP360	P36 port extended function	CFP36[1:0]	Function	0	R/W	When TRCMUX (D0/0x300014) is set to 1 (default), this register becomes ineffective and the port is configured for debugging.	
					1*	reserved	0			
					01	P36				
		D3 D2	CFP351 CFP350	P35 port extended function	CFP35[1:0]	Function	0	R/W		
					1*	reserved	0			
					01	P35				
		D1 D0	CFP341 CFP340	P34 port extended function	CFP34[1:0]	Function	0	R/W		
					1*	reserved	0			
					01	P34				
							00	DSIO		

This register selects the functions of ports P34 to P36.

D[7:6] Reserved**D[5:4] CFP36[1:0]: P36 Port Extended Function Select Bits**

11 (R/W): Reserved

10 (R/W): Reserved

01 (R/W): P36

00 (R/W): DST2 (default)

To use P36 as a general-purpose I/O port, TRCMUX (D0/0x300014) must be set to 0 and CFP36[1:0] set to 01.

* **TRCMUX**: P15–17, P34–36 Debug Function Select Bit in the Debug Port MUX Register (D0/0x300014)

D[3:2] CFP35[1:0]: P35 Port Extended Function Select Bits

11 (R/W): Reserved

10 (R/W): Reserved

01 (R/W): P35

00 (R/W): DCLK (default)

To use P35 as a general-purpose I/O port, TRCMUX (D0/0x300014) must be set to 0 and CFP35[1:0] set to 01.

D[1:0] CFP34[1:0]: P34 Port Extended Function Select Bits

11 (R/W): Reserved

10 (R/W): Reserved

01 (R/W): P34

00 (R/W): DSIO (default)

To use P34 as a general-purpose I/O port, TRCMUX (D0/0x300014) must be set to 0 and CFP34[1:0] set to 01.

0x3003A8: P40–P43 Port Function Select Register (pP4_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P40–P43 port function select register (pP4_03_CFP)	003003A8 (B)	D7	CFP431	P43 port extended function	CFP43[1:0]	Function	0	R/W
		D6	CFP430		11	reserved	0	
					10	FPDAT9		
					01	P43		
				00	A21			
		D5	CFP421	P42 port extended function	CFP42[1:0]	Function	0	R/W
		D4	CFP420		11	reserved	0	
					10	FPDAT8		
					01	P42		
				00	A22			
		D3	CFP411	P41 port extended function	CFP41[1:0]	Function	0	R/W
		D2	CFP410		11	EXCL3	0	
					10	#SDRAS		
					01	P41		
				00	A23			
		D1	CFP401	P40 port extended function	CFP40[1:0]	Function	0	R/W
D0	CFP400	11	EXCL4		0			
		10	#SDCAS					
		01	P40					
		00	A24					

This register selects the functions of ports P40 to P43.

D[7:6] CFP43[1:0]: P43 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): FPDAT9
- 01 (R/W): P43
- 00 (R/W): A21 (default)

D[5:4] CFP42[1:0]: P42 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): FPDAT8
- 01 (R/W): P42
- 00 (R/W): A22 (default)

D[3:2] CFP41[1:0]: P41 Port Extended Function Select Bits

- 11 (R/W): EXCL3
- 10 (R/W): #SDRAS
- 01 (R/W): P41
- 00 (R/W): A23 (default)

D[1:0] CFP40[1:0]: P40 Port Extended Function Select Bits

- 11 (R/W): EXCL4
- 10 (R/W): #SDCAS
- 01 (R/W): P40
- 00 (R/W): A24 (default)

0x3003A9: P44–P47 Port Function Select Register (pP4_47_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P44–P47 port function select register (pP4_47_CFP)	003003A9 (B)	D7	CFP471	P47 port extended function	CFP47[1:0]	Function	0	R/W
		D6	CFP470		1*	reserved	0	
					01	P47		
					00	A11		
	D5	CFP461	P46 port extended function	CFP46[1:0]	Function	0	R/W	
	D4	CFP460		11	reserved	0		
				10	TFT_CTL2			
				01	P46			
	00	A18						
	D3	CFP451	P45 port extended function	CFP45[1:0]	Function	0	R/W	
	D2	CFP450		11	reserved	0		
				10	FPDAT11			
				01	P45			
	00	A19						
	D1	CFP441	P44 port extended function	CFP44[1:0]	Function	0	R/W	
	D0	CFP440		11	reserved	0		
10				FPDAT10				
01				P44				
00	A20							

This register selects the functions of ports P44 to P47.

D[7:6] CFP47[1:0]: P47 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): P47
- 00 (R/W): A11 (default)

D[5:4] CFP46[1:0]: P46 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): TFT_CTL2
- 01 (R/W): P46
- 00 (R/W): A18 (default)

D[3:2] CFP45[1:0]: P45 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): FPDAT11
- 01 (R/W): P45
- 00 (R/W): A19 (default)

D[1:0] CFP44[1:0]: P44 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): FPDAT10
- 01 (R/W): P44
- 00 (R/W): A20 (default)

0x3003AA: P50–P53 Port Function Select Register (pP5_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P50–P53 port function select register (pP5_03_CFP)	003003AA (B)	D7	CFP531	P53 port extended function	CFP53[1:0]	Function	0	R/W
		D6	CFP530		11	reserved	0	
					10	SDA10		
					01	P53		
					00	#CE7		
		D5	CFP521	P52 port extended function	CFP52[1:0]	Function	0	R/W
		D4	CFP520		11	CMU_CLK	0	
					10	#CE6		
					01	BCLK		
					00	P52		
		D3	CFP511	P51 port extended function	CFP51[1:0]	Function	0	R/W
		D2	CFP510		11	reserved	0	
					10	CARD1		
					01	P51		
					00	#CE5		
		D1	CFP501	P50 port extended function	CFP50[1:0]	Function	0	R/W
D0	CFP500	11	reserved		0			
		10	CARD0					
		01	P50					
		00	#CE4					

This register selects the functions of ports P50 to P53.

D[7:6] CFP53[1:0]: P53 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): SDA10
- 01 (R/W): P53
- 00 (R/W): #CE7 (default)

D[5:4] CFP52[1:0]: P52 Port Extended Function Select Bits

- 11 (R/W): CMU_CLK
- 10 (R/W): #CE6
- 01 (R/W): BCLK
- 00 (R/W): P52 (default)

D[3:2] CFP51[1:0]: P51 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): CARD1
- 01 (R/W): P51
- 00 (R/W): #CE5 (default)

D[1:0] CFP50[1:0]: P50 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): CARD0
- 01 (R/W): P50
- 00 (R/W): #CE4 (default)

0x3003AB: P54–P57 Port Function Select Register (pP5_47_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P54–P57 port function select register (pP5_47_CFP)	003003AB (B)	D7	CFP571	P57 port extended function	CFP57[1:0]	Function	0	R/W
		D6	CFP570		1*	reserved	0	
					01	P57		
					00	#CE10		
		D5	CFP561	P56 port extended function	CFP56[1:0]	Function	0	R/W
		D4	CFP560		1*	reserved	0	
					01	P56		
					00	#CE11		
		D3	CFP551	P55 port extended function	CFP55[1:0]	Function	0	R/W
		D2	CFP550		11	reserved	0	
					10	CARD0		
					01	P55		
	D1	CFP541	P54 port extended function	CFP54[1:0]	Function	0	R/W	
	D0	CFP540		11	reserved	0		
				10	CARD1			
				01	P54			
				00	#CE8			

This register selects the functions of ports P54 to P57.

D[7:6] CFP57[1:0]: P57 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): P57
- 00 (R/W): #CE10 (default)

D[5:4] CFP56[1:0]: P56 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): P56
- 00 (R/W): #CE11 (default)

D[3:2] CFP55[1:0]: P55 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): CARD0
- 01 (R/W): P55
- 00 (R/W): #CE9 (default)

D[1:0] CFP54[1:0]: P54 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): CARD1
- 01 (R/W): P54
- 00 (R/W): #CE8 (default)

0x3003AC: P60–P63 Port Function Select Register (pP6_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P60–P63 port function select register (pP6_03_CFP)	003003AC (B)	D7	CFP631	P63 port extended function	CFP63[1:0]	Function	0	R/W
		D6	CFP630		11	#WDT_NMI	0	
					10	WDT_CLK		
					01	#SRDY2		
				00	P63			
		D5	CFP621	P62 port extended function	CFP62[1:0]	Function	0	R/W
		D4	CFP620		11	CMU_CLK	0	
					10	#ADTRG		
					01	#SCLK2		
				00	P62			
		D3	CFP611	P61 port extended function	CFP61[1:0]	Function	0	R/W
		D2	CFP610		11	EXCL1	0	
					10	DCSIO1		
					01	SOUT2		
				00	P61			
		D1	CFP601	P60 port extended function	CFP60[1:0]	Function	0	R/W
D0	CFP600	11	EXCL0		0			
		10	DCSIO0					
		01	SIN2					
		00	P60					

This register selects the functions of ports P60 to P63.

D[7:6] CFP63[1:0]: P63 Port Extended Function Select Bits

- 11 (R/W): #WDT_NMI
- 10 (R/W): WDT_CLK
- 01 (R/W): #SRDY2
- 00 (R/W): P63 (default)

D[5:4] CFP62[1:0]: P62 Port Extended Function Select Bits

- 11 (R/W): CMU_CLK
- 10 (R/W): #ADTRG
- 01 (R/W): #SCLK2
- 00 (R/W): P62 (default)

D[3:2] CFP61[1:0]: P61 Port Extended Function Select Bits

- 11 (R/W): EXCL1
- 10 (R/W): DCSIO1
- 01 (R/W): SOUT2
- 00 (R/W): P61 (default)

D[1:0] CFP60[1:0]: P60 Port Extended Function Select Bits

- 11 (R/W): EXCL0
- 10 (R/W): DCSIO0
- 01 (R/W): SIN2
- 00 (R/W): P60 (default)

0x3003AD: P64–P67 Port Function Select Register (pP6_47_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P64–P67 port function select register (pP6_47_CFP)	003003AD (B)	D7 D6	CFP671	P67 port extended function	CFP67[1:0]	Function	0	R/W	
			CFP670		11	reserved	0		
					10	FPDAT10			
					01	SPI_CLK			
		00		P67					
		D5 D4	CFP661	P66 port extended function	CFP66[1:0]	Function	0	R/W	
			CFP660		11	reserved	0		
					10	FPDAT9			
					01	SDO			
		00		P66					
		D3 D2	CFP651	P65 port extended function	CFP65[1:0]	Function	0	R/W	
			CFP650		11	reserved	0		
					10	FPDAT8			
					01	SDI			
		00		P65					
		D1 D0	CFP641	P64 port extended function	CFP64[1:0]	Function	0	R/W	
CFP640	11		reserved		0				
	10		EXCL2						
	01		#WAIT						
	00	P64							

This register selects the functions of ports P64 to P67.

D[7:6] CFP67[1:0]: P67 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): FPDAT10
- 01 (R/W): SPI_CLK
- 00 (R/W): P67 (default)

D[5:4] CFP66[1:0]: P66 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): FPDAT9
- 01 (R/W): SDO
- 00 (R/W): P66 (default)

D[3:2] CFP65[1:0]: P65 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): FPDAT8
- 01 (R/W): SDI
- 00 (R/W): P65 (default)

D[1:0] CFP64[1:0]: P64 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): EXCL2
- 01 (R/W): #WAIT
- 00 (R/W): P64 (default)

0x3003AE: P70–P73 Port Function Select Register (pP7_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P70–P73 port function select register (pP7_03_CFP)	003003AE (B)	D7	CFP731	P73 port extended function	CFP73[1:0]	Function	0	R/W
		D6	CFP730		1*	reserved	0	
					01	AIN3		
					00	P73		
		D5	CFP721	P72 port extended function	CFP72[1:0]	Function	0	R/W
		D4	CFP720		1*	reserved	0	
					01	AIN2		
					00	P72		
		D3	CFP711	P71 port extended function	CFP71[1:0]	Function	0	R/W
		D2	CFP710		1*	reserved	0	
					01	AIN1		
					00	P71		
D1	CFP701	P70 port extended function	CFP70[1:0]	Function	0	R/W		
D0	CFP700		1*	reserved	0			
			01	AIN0				
			00	P70				

This register selects the functions of ports P70 to P73.

D[7:6] CFP73[1:0]: P73 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): AIN3
- 00 (R/W): P73 (default)

D[5:4] CFP72[1:0]: P72 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): AIN2
- 00 (R/W): P72 (default)

D[3:2] CFP71[1:0]: P71 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): AIN1
- 00 (R/W): P71 (default)

D[1:0] CFP70[1:0]: P70 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): AIN0
- 00 (R/W): P70 (default)

0x3003AF: P74 Port Function Select Register (pP7_4_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P74 port function select register (pP7_4_CFP)	003003AF (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	CFP741	P74 port extended function	CFP74[1:0]	Function	0	R/W
		D0	CFP740		11	reserved	0	
					10	EXCL5		
					01	AIN4		
				00	P74			

This register selects the function of port P74.

D[7:2] Reserved

D[1:0] CFP74[1:0]: P74 Port Extended Function Select Bits

11 (R/W): Reserved

10 (R/W): EXCL5

01 (R/W): AIN4

00 (R/W): P74 (default)

0x3003B0: P80–P83 Port Function Select Register (pP8_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P80–P83 port function select register (pP8_03_CFP)	003003B0 (B)	D7	CFP831	P83 port extended function	CFP83[1:0]	Function	0	R/W
		D6	CFP830		11	BCLK	0	
					10	TFT_CTL1		
					01	FPDRDY		
				00	P83			
		D5	CFP821	P82 port extended function	CFP82[1:0]	Function	0	R/W
		D4	CFP820		1*	reserved	0	
					01	FPSHIFT		
					00	P82		
		D3	CFP811	P81 port extended function	CFP81[1:0]	Function	0	R/W
		D2	CFP810		1*	reserved	0	
					01	FPLINE		
					00	P81		
		D1	CFP801	P80 port extended function	CFP80[1:0]	Function	0	R/W
		D0	CFP800		1*	reserved	0	
					01	FPFRAME		
		00	P80					

This register selects the functions of ports P80 to P83.

D[7:6] CFP83[1:0]: P83 Port Extended Function Select Bits

- 11 (R/W): BCLK
- 10 (R/W): TFT_CTL1
- 01 (R/W): FPDRDY
- 00 (R/W): P83 (default)

D[5:4] CFP82[1:0]: P82 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): FPSHIFT
- 00 (R/W): P82 (default)

D[3:2] CFP81[1:0]: P81 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): FPLINE
- 00 (R/W): P81 (default)

D[1:0] CFP80[1:0]: P80 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): FPFRAME
- 00 (R/W): P80 (default)

0x3003B1: P84–P85 Port Function Select Register (pP8_45_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P84–P85 port function select register (pP8_45_CFP)	003003B1 (B)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CFP851	P85 port extended function	CFP85[1:0]	Function	0	R/W
		D2	CFP850		1*	reserved	0	
					01	DCSIO1		
					00	P85		
		D1	CFP841	P84 port extended function	CFP84[1:0]	Function	0	R/W
		D0	CFP840		11	reserved	0	
					10	FPDAT11		
		01	DCSIO0					
				00	P84			

This register selects the functions of ports P84 to P85.

D[7:4] Reserved**D[3:2] CFP85[1:0]: P85 Port Extended Function Select Bits**

11 (R/W): Reserved

10 (R/W): Reserved

01 (R/W): DCSIO1

00 (R/W): P85 (default)

D[1:0] CFP84[1:0]: P84 Port Extended Function Select Bits

11 (R/W): Reserved

10 (R/W): FPDAT11

01 (R/W): DCSIO0

00 (R/W): P84 (default)

0x3003B2: P90–P93 Port Function Select Register (pP9_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P90–P93 port function select register (pP9_03_CFP)	003003B2 (B)	D7	CFP931	P93 port extended function	CFP93[1:0]	Function	0	R/W
		D6	CFP930		1*	reserved	0	
					01	FPDAT3		
					00	P93		
		D5	CFP921	P92 port extended function	CFP92[1:0]	Function	0	R/W
		D4	CFP920		1*	reserved	0	
					01	FPDAT2		
					00	P92		
		D3	CFP911	P91 port extended function	CFP91[1:0]	Function	0	R/W
		D2	CFP910		1*	reserved	0	
					01	FPDAT1		
					00	P91		
D1	CFP901	P90 port extended function	CFP90[1:0]	Function	0	R/W		
D0	CFP900		1*	reserved	0			
			01	FPDAT0				
			00	P90				

This register selects the functions of ports P90 to P93.

D[7:6] CFP93[1:0]: P93 Port Extended Function Select Bits

11 (R/W): Reserved
 10 (R/W): Reserved
 01 (R/W): FPDAT3
 00 (R/W): P93 (default)

D[5:4] CFP92[1:0]: P92 Port Extended Function Select Bits

11 (R/W): Reserved
 10 (R/W): Reserved
 01 (R/W): FPDAT2
 00 (R/W): P92 (default)

D[3:2] CFP91[1:0]: P91 Port Extended Function Select Bits

11 (R/W): Reserved
 10 (R/W): Reserved
 01 (R/W): FPDAT1
 00 (R/W): P91 (default)

D[1:0] CFP90[1:0]: P90 Port Extended Function Select Bits

11 (R/W): Reserved
 10 (R/W): Reserved
 01 (R/W): FPDAT0
 00 (R/W): P90 (default)

0x3003B3: P94–P97 Port Function Select Register (pP9_47_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P94–P97 port function select register (pP9_47_CFP)	003003B3 (B)	D7 D6	CFP971	P97 port extended function	CFP97[1:0]	Function	0	R/W
			CFP970		1*	reserved	0	
					01	FPDAT7	0	
					00	P97		
		D5 D4	CFP961	P96 port extended function	CFP96[1:0]	Function	0	R/W
			CFP960		1*	reserved	0	
					01	FPDAT6	0	
					00	P96		
		D3 D2	CFP951	P95 port extended function	CFP95[1:0]	Function	0	R/W
			CFP950		1*	reserved	0	
					01	FPDAT5	0	
					00	P95		
		D1 D0	CFP941	P94 port extended function	CFP94[1:0]	Function	0	R/W
			CFP940		1*	reserved	0	
					01	FPDAT4	0	
					00	P94		

This register selects the functions of ports P94 to P97.

D[7:6] CFP97[1:0]: P97 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): FPDAT7
- 00 (R/W): P97 (default)

D[5:4] CFP96[1:0]: P96 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): FPDAT6
- 00 (R/W): P96 (default)

D[3:2] CFP95[1:0]: P95 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): FPDAT5
- 00 (R/W): P95 (default)

D[1:0] CFP94[1:0]: P94 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): FPDAT4
- 00 (R/W): P94 (default)

0x300C20: PA0–PA3 Port Function Select Register (pPA_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PA0–PA3 port function select register (pPA_03_CFP)	00300C20 (B)	D7	CFPA31	PA3 port extended function	CFPA3[1:0]	Function	0	R/W
		D6	CFPA30		11	CARD0	0	
					10	TFT_CTL2		
					01	FPDAT10		
					00	PA3		
		D5	CFPA21	PA2 port extended function	CFPA2[1:0]	Function	0	R/W
		D4	CFPA20		11	reserved	0	
					10	TFT_CTL1		
					01	FPDAT9		
					00	PA2		
		D3	CFPA11	PA1 port extended function	CFPA1[1:0]	Function	0	R/W
		D2	CFPA10		11	reserved	0	
					10	TFT_CTL0		
					01	FPDAT8		
					00	PA1		
		D1	CFPA01	PA0 port extended function	CFPA0[1:0]	Function	0	R/W
D0	CFPA00	1*	reserved		0			
		01	TFT_CTL0					
		00	PA0					

Note: The PA0 to PA3 ports are not available in the QFP24-144pin package model.

This register selects the functions of ports PA0 to PA3.

D[7:6] CFPA3[1:0]: PA3 Port Extended Function Select Bits

- 11 (R/W): CARD0
- 10 (R/W): TFT_CTL2
- 01 (R/W): FPDAT10
- 00 (R/W): PA3 (default)

D[5:4] CFPA2[1:0]: PA2 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): TFT_CTL1
- 01 (R/W): FPDAT9
- 00 (R/W): PA2 (default)

D[3:2] CFPA1[1:0]: PA1 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): TFT_CTL0
- 01 (R/W): FPDAT8
- 00 (R/W): PA1 (default)

D[1:0] CFPA0[1:0]: PA0 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): TFT_CTL0
- 00 (R/W): PA0 (default)

0x300C21: PA4 Port Function Select Register (pPA_4_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PA4 port function select register (pPA_4_CFP)	00300C21 (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	CFPA41	PA4 port extended function	CFPA4[1:0]	Function	0	R/W
		D0	CFPA40		11	CARD1	0	
					10	TFT_CTL3		
					01	FPDAT11		
				00	PA4			

Note: The PA4 port is not available in the QFP24-144pin package model.

This register selects the function of port PA4.

D[7:2] Reserved

D[1:0] CFPA4[1:0]: PA4 Port Extended Function Select Bits

11 (R/W): CARD1

10 (R/W): TFT_CTL3

01 (R/W): FPDAT11

00 (R/W): PA4 (default)

0x300C22: PB0–PB3 Port Function Select Register (pPB_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PB0–PB3 port function select register (pPB_03_CFP)	00300C22 (B)	D7	CFPB31	PB3 port extended function	CFPB3[1:0]	Function	0	R/W
		D6	CFPB30		11	CARD5	0	
					10	I2S_MCLK		
					01	FPDAT11		
					00	PB3		
		D5	CFPB21	PB2 port extended function	CFPB2[1:0]	Function	0	R/W
		D4	CFPB20		11	CARD4	0	
					10	I2S_SCK		
					01	FPDAT10		
					00	PB2		
		D3	CFPB11	PB1 port extended function	CFPB1[1:0]	Function	0	R/W
		D2	CFPB10		11	CARD3	0	
					10	I2S_WS		
					01	FPDAT9		
					00	PB1		
		D1	CFPB01	PB0 port extended function	CFPB0[1:0]	Function	0	R/W
D0	CFPB00	11	CARD2		0			
		10	I2S_SDO					
		01	FPDAT8					
		00	PB0					

Note: The PB0 to PB3 ports are not available in the QFP24-144pin package model.

This register selects the functions of ports PB0 to PB3.

D[7:6] CFPB3[1:0]: PB3 Port Extended Function Select Bits

- 11 (R/W): CARD5
- 10 (R/W): I2S_MCLK
- 01 (R/W): FPDAT11
- 00 (R/W): PB3 (default)

D[5:4] CFPB2[1:0]: PB2 Port Extended Function Select Bits

- 11 (R/W): CARD4
- 10 (R/W): I2S_SCK
- 01 (R/W): FPDAT10
- 00 (R/W): PB2 (default)

D[3:2] CFPB1[1:0]: PB1 Port Extended Function Select Bits

- 11 (R/W): CARD3
- 10 (R/W): I2S_WS
- 01 (R/W): FPDAT9
- 00 (R/W): PB1 (default)

D[1:0] CFPB0[1:0]: PB0 Port Extended Function Select Bits

- 11 (R/W): CARD2
- 10 (R/W): I2S_SDO
- 01 (R/W): FPDAT8
- 00 (R/W): PB0 (default)

0x300C24: PC0–PC3 Port Function Select Register (pPC_03_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PC0–PC3 port function select register (pPC_03_CFP)	00300C24 (B)	D7	CFPC31	PC3 port extended function	CFPC3[1:0]	Function	0	R/W
		D6	CFPC30		1*	reserved	0	
					01	PC3		
					00	D11		
		D5	CFPC21	PC2 port extended function	CFPC2[1:0]	Function	0	R/W
		D4	CFPC20		1*	reserved	0	
					01	PC2		
					00	D10		
		D3	CFPC11	PC1 port extended function	CFPC1[1:0]	Function	0	R/W
		D2	CFPC10		1*	reserved	0	
					01	PC1		
					00	D9		
	D1	CFPC01	PC0 port extended function	CFPC0[1:0]	Function	0	R/W	
	D0	CFPC00		1*	reserved	0		
				01	PC0			
				00	D8			

This register selects the functions of ports PC0 to PC3.

D[7:6] CFPC3[1:0]: PC3 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): PC3
- 00 (R/W): D11 (default)

D[5:4] CFPC2[1:0]: PC2 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): PC2
- 00 (R/W): D10 (default)

D[3:2] CFPC1[1:0]: PC1 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): PC1
- 00 (R/W): D9 (default)

D[1:0] CFPC0[1:0]: PC0 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): PC0
- 00 (R/W): D8 (default)

0x300C25: PC4–PC7 Port Function Select Register (pPC_47_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PC4–PC7 port function select register (pPC_47_CFP)	00300C25 (B)	D7	CFPC71	PC7 port extended function	CFPC7[1:0]	Function	0	R/W
		D6	CFPC70		1*	reserved	0	
					01	PC7		
					00	D15		
		D5	CFPC61	PC6 port extended function	CFPC6[1:0]	Function	0	R/W
		D4	CFPC60		1*	reserved	0	
					01	PC6		
					00	D14		
		D3	CFPC51	PC5 port extended function	CFPC5[1:0]	Function	0	R/W
		D2	CFPC50		1*	reserved	0	
					01	PC5		
					00	D13		
D1	CFPC41	PC4 port extended function	CFPC4[1:0]	Function	0	R/W		
D0	CFPC40		1*	reserved	0			
			01	PC4				
			00	D12				

This register selects the functions of ports PC4 to PC7.

D[7:6] CFPC7[1:0]: PC7 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): PC7
- 00 (R/W): D15 (default)

D[5:4] CFPC6[1:0]: PC6 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): PC6
- 00 (R/W): D14 (default)

D[3:2] CFPC5[1:0]: PC5 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): PC5
- 00 (R/W): D13 (default)

D[1:0] CFPC4[1:0]: PC4 Port Extended Function Select Bits

- 11 (R/W): Reserved
- 10 (R/W): Reserved
- 01 (R/W): PC4
- 00 (R/W): D12 (default)

I.3.4 Input/Output Cells and Input/Output Characteristics

Table I.3.4.1 Pin Characteristics

Signal name	I/O	I/O cell name	Input level	Ioh/Iol	Pull-up/down	Power source	Remarks
MCLKI	I	LLINY	Transparent	–	–	V _{DD}	note 2
MCLKO	O	LLOTY	–	–	–	V _{DD}	note 2
RTC_CLKI	I	LLINY	Transparent	–	–	V _{DD}	note 2
RTC_CLKO	O	LLOTY	–	–	–	V _{DD}	note 2
VCP	O	LLOTY	–	–	–	PLV _{DD}	note 4
#RESET	I	HIBHP1TY	Schmitt	–	50 kΩ up	V _{DDH}	
#NMI	I	HIBHP1TY	Schmitt	–	50 kΩ up	V _{DDH}	
BOOT1	I	HIBASP2TY	LVC MOS	–	100 kΩ up	V _{DDH}	
BOOT0	I	HIBHP1TY	Schmitt	–	50 kΩ up	V _{DDH}	note 5
BURNIN	I	HIBHY	Schmitt	–	50 kΩ down	V _{DDH}	note 5
DSIO (P34)	I/O	HBBH2BP1TY	Schmitt	4 mA	50 kΩ up	V _{DDH}	
DCLK (P35)	I/O	HBBH2BP1TY	Schmitt	4 mA	50 kΩ up	V _{DDH}	
DST2 (P36)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	
D0	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D1	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D2	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D3	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D4	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D5	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D6	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D7	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D8 (PC0)	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D9 (PC1)	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D10 (PC2)	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D11 (PC3)	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D12 (PC4)	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D13 (PC5)	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D14 (PC6)	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
D15 (PC7)	I/O	HBBT2AHTY	LV TTL	4 mA	Bus-hold latch	V _{DDH}	
A0/#BSL	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A1	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A2	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A3	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A4	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A5	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A6	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A7	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A8	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A9	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A10	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A11 (P47)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A12	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A13	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A14	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A15	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A16/DQML	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A17/DQMH	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A18 (P46/TFT_CTL2)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A19 (P45/FPDAT11)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A20 (P44/FPDAT10)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A21 (P43/FPDAT9)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A22 (P42/FPDAT8)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A23 (P41/#SDRAS/EXCL3)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
A24 (P40/#SDCAS/EXCL4)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
#RD	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
#WRL	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
#WRH/#BSH	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
#CE10 (P57)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
#CE4 (P50/CARD0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
#CE5 (P51/CARD1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1

Signal name	I/O	I/O cell name	Input level	I _{OH} /I _{OL}	Pull-up/down	Power source	Remarks
P52 (BCLK/#CE6/CMU_CLK)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
#CE7 (P53/SDA10)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
#CE8 (P54/CARD1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
#CE9 (P55/CARD0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
#CE11 (P56)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P00 (SIN0/#DMAACK2)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P01 (SOUT0/#DMAACK3)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P02 (#SCLK0/#DMAEND2)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P03 (#SRDY0/#DMAEND3)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P04 (SIN1/I2S_SDO)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P05 (SOUT1/I2S_WS)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P06 (#SCLK1/I2S_SCK)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P07 (#SRDY1/I2S_MCLK)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P10 (TM0/SIN0/#DMAEND0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P11 (TM1/SOUT0/#DMAEND1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P12 (TM2/#SCLK0/#DMAACK0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P13 (TM3/#SRDY0/#DMAACK1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P14 (TM4/SIN1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
DST0 (P15/TM5/SOUT1/TFT_CTL0)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
DST1 (P16/DCSIO0/#SCLK1/TFT_CTL3)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
DPC0 (P17/DCSIO1/#SRDY1/TFT_CTL2)	I/O	HBBH2BP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P20 (SDCKE)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
SDCLK (P21)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P22 (#SDCS)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P23 (#SDRAS/TFT_CTL1)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P24 (#SDCAS)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P25 (#SDWE)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P26 (DQML)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P27 (DQMH)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1
P30 (CARD2/#DMAREQ0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P31 (CARD3/#DMAREQ1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P32 (CARD4/#DMAREQ2)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P33 (CARD5/#DMAREQ3)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P60 (SIN2/DCSIO0/EXCL0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P61 (SOUT2/DCSIO1/EXCL1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P62 (#SCLK2/#ADTRG/CMU_CLK)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P63 (#SRDY2/WDT_CLK/#WDT_NMI)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P64 (#WAIT/EXCL2)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P65 (SDI/FPDAT8)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P66 (SDO/FPDAT9)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P67 (SPI_CLK/FPDAT10)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P70 (AIN0)	I	HIBASP2TY	LVC MOS	–	100 kΩ up	AV _{DD}	note 1, 3
P71 (AIN1)	I	HIBASP2TY	LVC MOS	–	100 kΩ up	AV _{DD}	note 1, 3
P72 (AIN2)	I	HIBASP2TY	LVC MOS	–	100 kΩ up	AV _{DD}	note 1, 3
P73 (AIN3)	I	HIBASP2TY	LVC MOS	–	100 kΩ up	AV _{DD}	note 1, 3
P74 (AIN4/EXCL5)	I	HIBASP2TY	LVC MOS	–	100 kΩ up	AV _{DD}	note 1, 3
P80 (FPFRAME)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P81 (FPLINE)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P82 (FPSHIFT)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P83 (FPDRDY/TFT_CTL1/BCLK)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P84 (DCSIO0/FPDAT11)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P85 (DCSIO1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P90 (FPDAT0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P91 (FPDAT1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P92 (FPDAT2)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P93 (FPDAT3)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P94 (FPDAT4)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P95 (FPDAT5)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P96 (FPDAT6)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1
P97 (FPDAT7)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1

I S1C33E08 SPECIFICATIONS: PIN DESCRIPTION

Signal name	I/O	I/O cell name	Input level	I _{OH} /I _{OL}	Pull-up/down	Power source	Remarks
PA0 (TFT_CTL0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1, 5
PA1 (FPDAT8/TFT_CTL0)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1, 5
PA2 (FPDAT9/TFT_CTL1)	I/O	HBBH2AP2TY	Schmitt	4 mA	100 kΩ up	V _{DDH}	note 1, 5
PA3 (FPDAT10/TFT_CTL2/CARD0)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1, 5
PA4 (FPDAT11/TFT_CTL3/CARD1)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1, 5
PB0 (FPDAT8/I2S_SDO/CARD2)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1, 5
PB1 (FPDAT9/I2S_WS/CARD3)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1, 5
PB2 (FPDAT10/I2S_SCK/CARD4)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1, 5
PB3 (FPDAT11/I2S_MCLK/CARD5)	I/O	HBBH1BP2TY	Schmitt	2 mA	100 kΩ up	V _{DDH}	note 1, 5
USBDP	I/O	–	–	–	–	V _{DDH}	
USBDM	I/O	–	–	–	–	V _{DDH}	
USBVBUS	I	–	–	–	–	V _{DDH}	
TEST0	I	LITST1Y	LVC MOS	–	60 kΩ down	V _{DDH}	

- Notes:**
- 1 Pull-ups can be enabled or disabled by setting the pin control registers.
 - 2 This pin must be used in input voltage range $0\text{ V} \leq V_{IN} \leq V_{DD}$.
 - 3 This pin must be used in input voltage range $0\text{ V} \leq V_{IN} \leq AV_{DD}$.
 - 4 This pin must be used in input voltage range $0\text{ V} \leq V_{IN} \leq PLV_{DD}$.
 - 5 These pins are not available in the QFP24-144pin package model.

I.3.5 Package

I.3.5.1 QFP24-144pin Package

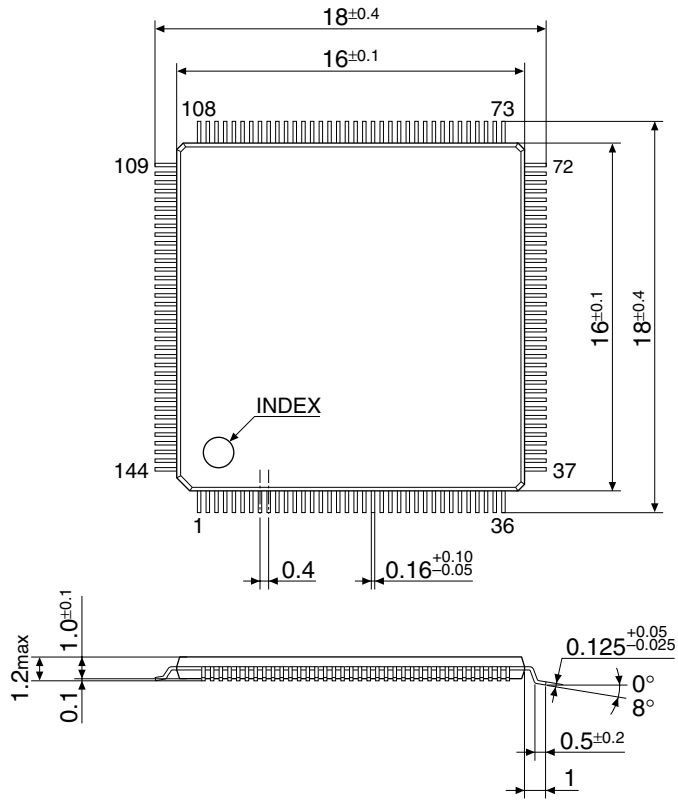


Figure I.3.5.1.1 QFP24-144pin Package Dimensions

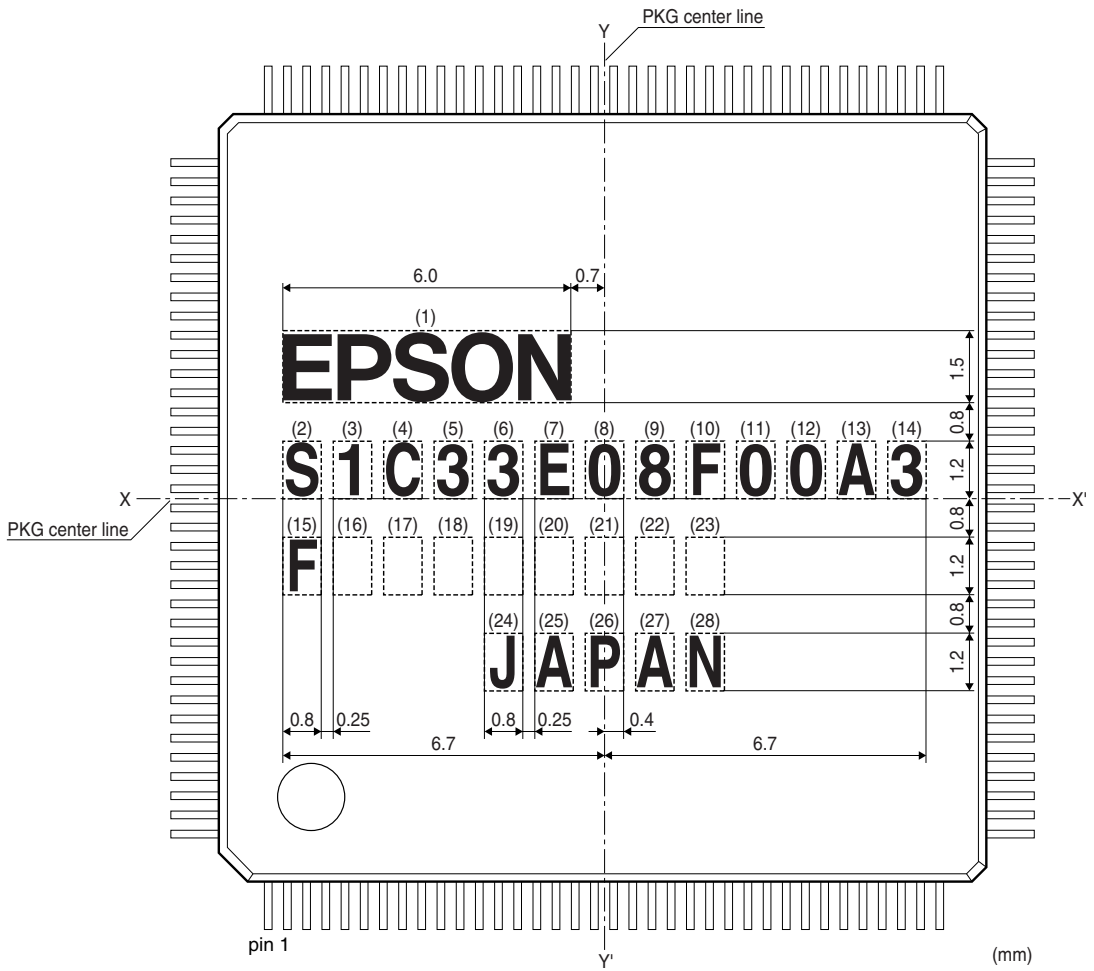
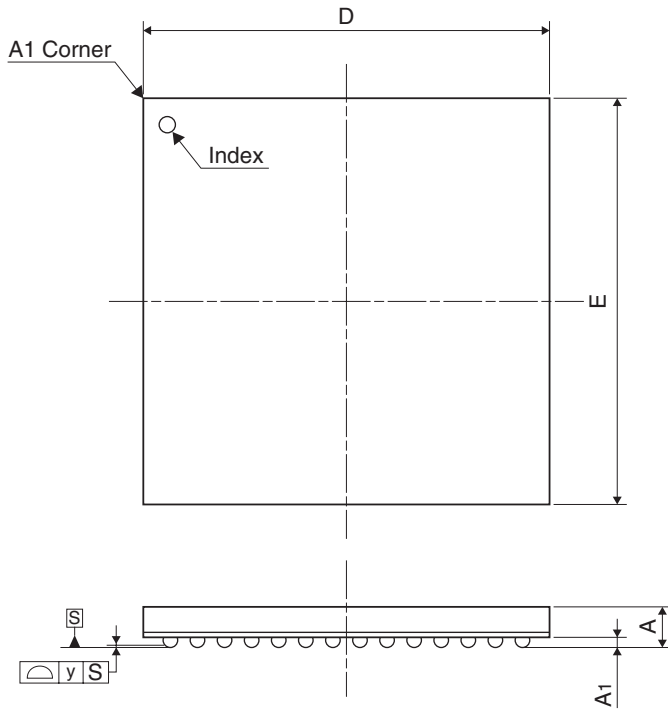


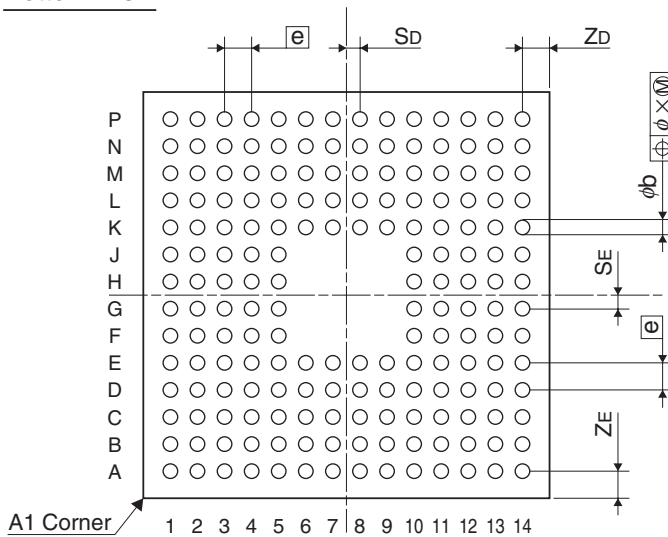
Figure I.3.5.1.2 QFP24-144pin Package Marking

I.3.5.2 PFBGA-180pin Package

Top View



Bottom View



Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	11.8	12.0	12.2
E	11.8	12.0	12.2
A			1.20
A1	0.25	0.30	0.35
e		0.80	
b	0.38	0.43	0.48
X			0.08
Y			0.10
S _d		0.40	
S _E		0.40	
Z _d		0.80	
Z _E		0.80	

Figure I.3.5.2.1 PFBGA-180pin Package Dimensions

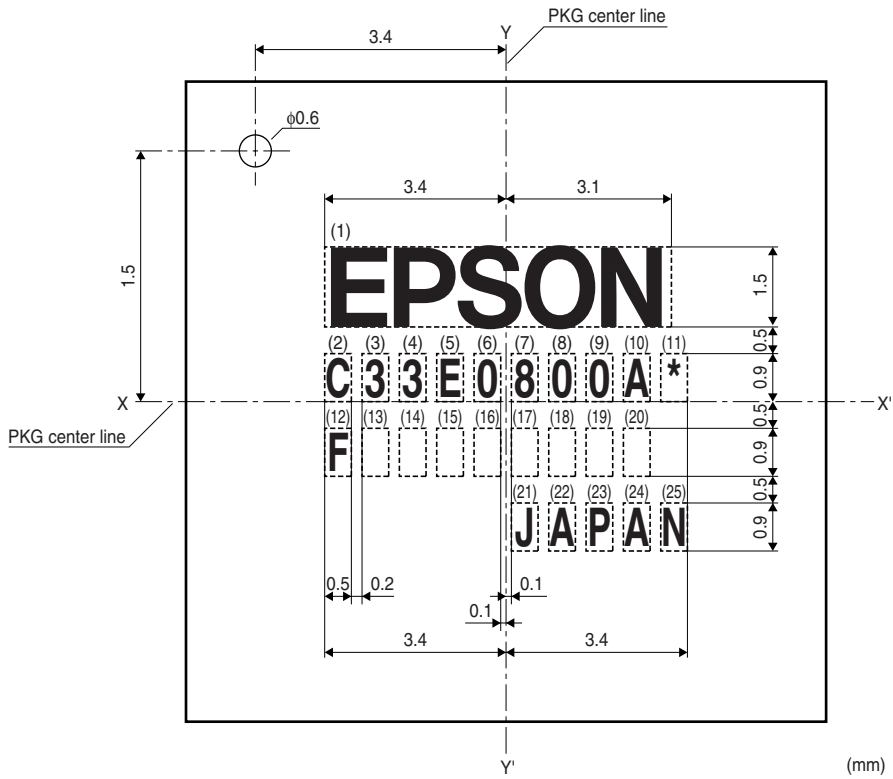


Figure I.3.5.2.2 PFBGA-180pin Package Marking

(mm)

I.3.5.3 Thermal Resistance of the Package

The chip temperature of LSI devices tends to increase with the power consumed on the chip. The chip temperature when encapsulated in a package is calculated from its ambient temperature (T_a), the thermal resistance of the package (θ), and power dissipation (P_D).

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

When used under normal operating conditions, make sure that the chip temperature (T_j) is 100°C or less.

Thermal resistance of the QFP24-144pin package

1. When mounted on a board (windless condition)

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

This value indicates the thermal resistance of the package when measured under a windless condition, with the sample mounted on a measurement board (size: 114 × 76 × 1.6 mm thick, FR4/4 layered board).

2. When suspended alone (windless condition)

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

This value indicates the thermal resistance of the package when measured under a windless condition, with the sample suspended alone.

Thermal resistance of the PFBGA-180pin package

1. When mounted on a board (windless condition)

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

This value indicates the thermal resistance of the package when measured under a windless condition, with the sample mounted on a measurement board (size: 114.5 × 101.5 × 1.6 mm thick, FR4/4 layered board).

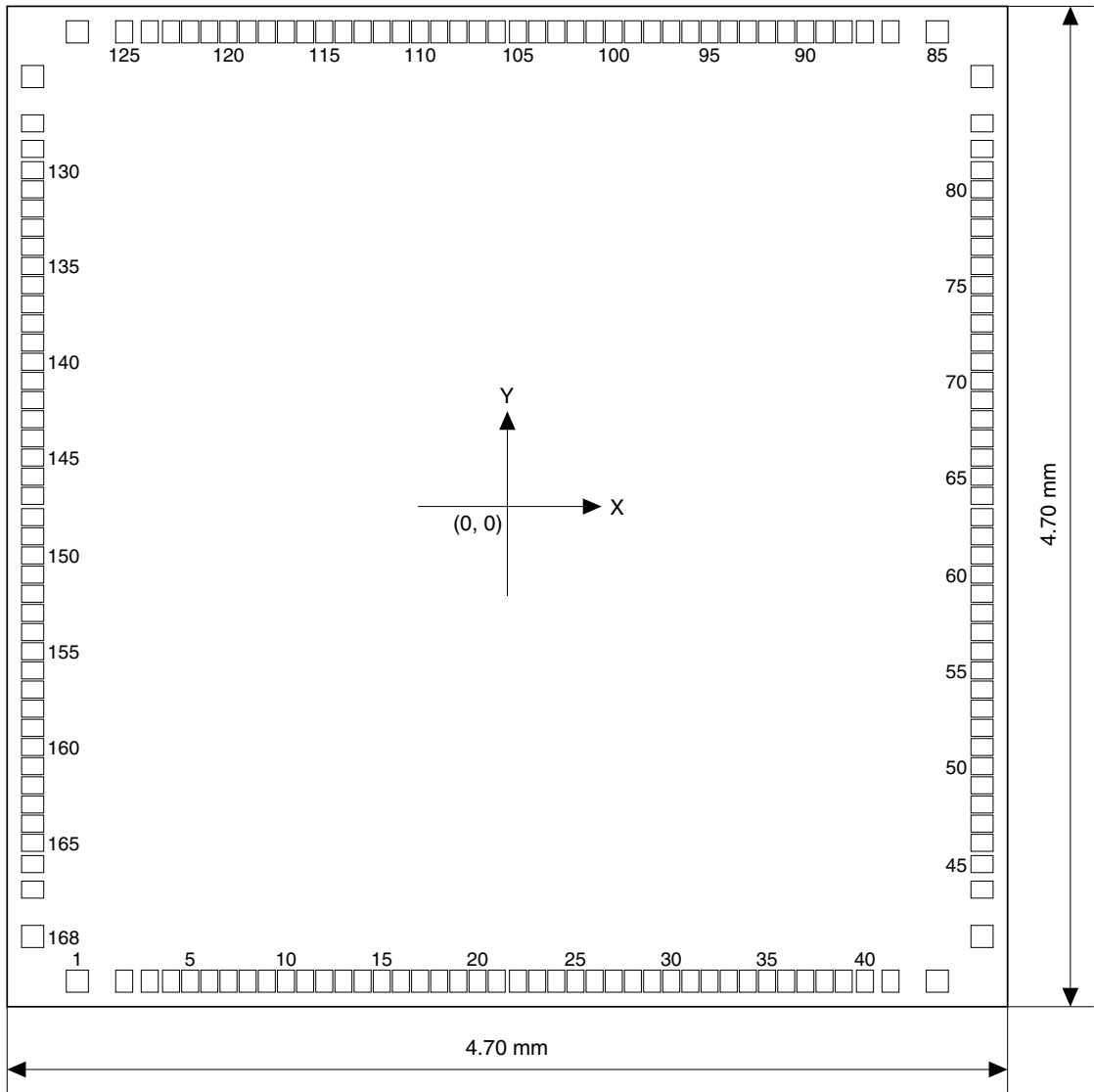
2. When suspended alone (windless condition)

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

This value indicates the thermal resistance of the package when measured under a windless condition, with the sample suspended alone.

Note: The thermal resistance of the package varies significantly depending on how it is mounted on the board and whether forcibly air-cooled.

I.3.6 Pad Layout



Pad opening (X × Y) 104 μm × 104 μm: PAD No. 1, 42, 43, 84, 85, 126, 127, 168
 80 μm × 104 μm: PAD No. 2–41, 86–125
 104 μm × 80 μm: PAD No. 44–83, 128–167

Figure I.3.6.1 Pad Layout Diagram

Table I.3.6.1 Pad Coordinate (unit: mm)

No.	Pad name	X	Y	No.	Pad name	X	Y
1	#CE9 (P55/CARD0)	-2.02	-2.23	51	TEST0	2.23	-1.13
2	A18 (P46/TFT_CTL2)	-1.80	-2.23	52	BOOT1	2.23	-1.04
3	A19 (P45/FPDAT11)	-1.68	-2.23	53	AVdd	2.23	-0.95
4	A20 (P44/FPDAT10)	-1.58	-2.23	54	P74 (AIN4/EXCL5)	2.23	-0.86
5	A21 (P43/FPDAT9)	-1.49	-2.23	55	P73 (AIN3)	2.23	-0.77
6	A22 (P42/FPDAT8)	-1.40	-2.23	56	P72 (AIN2)	2.23	-0.68
7	Vss	-1.31	-2.23	57	P71 (AIN1)	2.23	-0.59
8	P30 (CARD2/#DMAREQ0)	-1.22	-2.23	58	P70 (AIN0)	2.23	-0.50
9	P31 (CARD3/#DMAREQ1)	-1.13	-2.23	59	P80 (FPFRAME)	2.23	-0.41
10	P32 (CARD4/#DMAREQ2)	-1.04	-2.23	60	P81 (FPLINE)	2.23	-0.32
11	P33 (CARD5/#DMAREQ3)	-0.95	-2.23	61	P82 (FPSHIFT)	2.23	-0.23
12	P00 (SIN0/#DMAACK2)	-0.86	-2.23	62	P83 (FPDRDY/TFT_CTL1/BCLK)	2.23	-0.14
13	P01 (SOUT0/#DMAACK3)	-0.77	-2.23	63	P84 (DCSIO0/FPDAT11)	2.23	-0.05
14	P02 (#SCLK0/#DMAEND2)	-0.68	-2.23	64	P85 (DCSIO1)	2.23	0.05
15	P03 (#SRDY0/#DMAEND3)	-0.59	-2.23	65	Vdd	2.23	0.14
16	VDDH	-0.50	-2.23	66	MCLKI	2.23	0.23
17	P04 (SIN1/I2S_SDO)	-0.41	-2.23	67	MCLKO	2.23	0.32
18	P05 (SOUT1/I2S_WS)	-0.32	-2.23	68	Vss	2.23	0.41
19	P06 (#SCLK1/I2S_SCK)	-0.23	-2.23	69	BURNIN	2.23	0.50
20	P07 (#SRDY1/I2S_MCLK)	-0.14	-2.23	70	P90 (FPDAT0)	2.23	0.59
21	P10 (TM0/SIN0/#DMAEND0)	-0.05	-2.23	71	P91 (FPDAT1)	2.23	0.68
22	P11 (TM1/SOUT0/#DMAEND1)	0.05	-2.23	72	P92 (FPDAT2)	2.23	0.77
23	Vdd	0.14	-2.23	73	P93 (FPDAT3)	2.23	0.86
24	P12 (TM2/#SCLK0/#DMAACK0)	0.23	-2.23	74	VDDH	2.23	0.95
25	P13 (TM3/#SRDY0/#DMAACK1)	0.32	-2.23	75	PA0 (TFT_CTL0)	2.23	1.04
26	P14 (TM4/SIN1)	0.41	-2.23	76	PA1 (FPDAT4)	2.23	1.13
27	DST0 (P15/TM5/SOUT1/TFT_CTL0)	0.50	-2.23	77	PA1 (FPDAT8/TFT_CTL0)	2.23	1.22
28	DST1 (P16/DCSIO0/#SCLK1/TFT_CTL3)	0.59	-2.23	78	P95 (FPDAT5)	2.23	1.31
29	DPCO (P17/DCSIO1/#SRDY1/TFT_CTL2)	0.68	-2.23	79	PA2 (FPDAT9/TFT_CTL1)	2.23	1.40
30	VDDH	0.77	-2.23	80	P96 (FPDAT6)	2.23	1.49
31	VDDH	0.86	-2.23	81	PA3 (FPDAT10/TFT_CTL2/CARD0)	2.23	1.58
32	DSIO (P34)	0.95	-2.23	82	P97 (FPDAT7)	2.23	1.68
33	DST2 (P36)	1.04	-2.23	83	PA4 (FPDAT11/TFT_CTL3/CARD1)	2.23	1.80
34	DLCK (P35)	1.13	-2.23	84	#RESET	2.23	2.02
35	PB0 (FPDAT8/I2S_SDO/CARD2)	1.22	-2.23	85	#NMI	2.02	2.23
36	P64 (#WAIT)	1.31	-2.23	86	Vdd	1.80	2.23
37	PB1 (FPDAT9/I2S_WS/CARD3)	1.40	-2.23	87	USBDM	1.68	2.23
38	Vss	1.49	-2.23	88	USBDP	1.58	2.23
39	P65 (SDI/FPDAT8)	1.58	-2.23	89	N.C.	1.49	2.23
40	PB2 (FPDAT10/I2S_SCK/CARD4)	1.68	-2.23	90	USBVBUS	1.40	2.23
41	P66 (SDO/FPDAT9)	1.80	-2.23	91	N.C.	1.31	2.23
42	PB3 (FPDAT11/I2S_MCLK/CARD5)	2.02	-2.23	92	VDDH	1.22	2.23
43	P67 (SPL_CLK/FPDAT10)	2.23	-2.02	93	P60 (SIN2/DCSIO0/EXCL0)	1.13	2.23
44	Vdd	2.23	-1.80	94	P61 (SOUT2/DCSIO1/EXCL1)	1.04	2.23
45	RTC_CLKI	2.23	-1.68	95	P62 (#SCLK2/#ADTRG/CMU_CLK)	0.95	2.23
46	RTC_CLKO	2.23	-1.58	96	P63 (#SRDY2/WDT_CLK/#WDT_NMI)	0.86	2.23
47	Vss	2.23	-1.49	97	BOOT0	0.77	2.23
48	PLVss	2.23	-1.40	98	Vss	0.68	2.23
49	VCP	2.23	-1.31	99	D0	0.59	2.23
50	PLVdd	2.23	-1.22	100	D1	0.50	2.23

I S1C33E08 SPECIFICATIONS: PIN DESCRIPTION

No.	Pad name	X	Y	No.	Pad name	X	Y
101	D2	0.41	2.23	135	Vss	-2.23	1.13
102	D3	0.32	2.23	136	Vss	-2.23	1.04
103	VDD	0.23	2.23	137	A1	-2.23	0.95
104	D4	0.14	2.23	138	A2	-2.23	0.86
105	D5	0.05	2.23	139	A3	-2.23	0.77
106	D6	-0.05	2.23	140	A4	-2.23	0.68
107	D7	-0.14	2.23	141	A5	-2.23	0.59
108	VDDH	-0.23	2.23	142	VDD	-2.23	0.50
109	D8	-0.32	2.23	143	VDD	-2.23	0.41
110	D9	-0.41	2.23	144	A6	-2.23	0.32
111	D10	-0.50	2.23	145	A7	-2.23	0.23
112	D11	-0.59	2.23	146	A8	-2.23	0.14
113	VDD	-0.68	2.23	147	A9	-2.23	0.05
114	D12	-0.77	2.23	148	VDDH	-2.23	-0.05
115	D13	-0.86	2.23	149	VDDH	-2.23	-0.14
116	D14	-0.95	2.23	150	A10	-2.23	-0.23
117	D15	-1.04	2.23	151	A11 (P47)	-2.23	-0.32
118	Vss	-1.13	2.23	152	A12	-2.23	-0.41
119	P20 (SDCKE)	-1.22	2.23	153	A13	-2.23	-0.50
120	SDCLK (P21)	-1.31	2.23	154	A23 (P41/#SDRAS/EXCL3)	-2.23	-0.59
121	P22 (#SDCS)	-1.40	2.23	155	A24 (P40/#SDCAS/ EXCL4)	-2.23	-0.68
122	VDDH	-1.49	2.23	156	Vss	-2.23	-0.77
123	P23 (#SDRAS/TFT_CTL1)	-1.58	2.23	157	Vss	-2.23	-0.86
124	P24 (#SDCAS)	-1.68	2.23	158	#CE10 (P57)	-2.23	-0.95
125	P25 (#SDWE)	-1.80	2.23	159	#RD	-2.23	-1.04
126	Vss	-2.02	2.23	160	#WRL	-2.23	-1.13
127	P26 (DQML)	-2.23	2.02	161	#WRH/#BSH	-2.23	-1.22
128	P27 (DQMH)	-2.23	1.80	162	#CE11 (P56)	-2.23	-1.31
129	A14	-2.23	1.68	163	P52 (BCLK/#CE6/CMU_CLK)	-2.23	-1.40
130	A15	-2.23	1.58	164	VDD	-2.23	-1.49
131	A16/DQML	-2.23	1.49	165	#CE4 (P50/CARD0)	-2.23	-1.58
132	A17/DQMH	-2.23	1.40	166	#CE5 (P51/CARD1)	-2.23	-1.68
133	#CE7 (P53/SDA10)	-2.23	1.31	167	#CE8 (P54/CARD1)	-2.23	-1.80
134	A0/#BSL	-2.23	1.22	168	VDDH	-2.23	-2.02

I.4 Power Supply

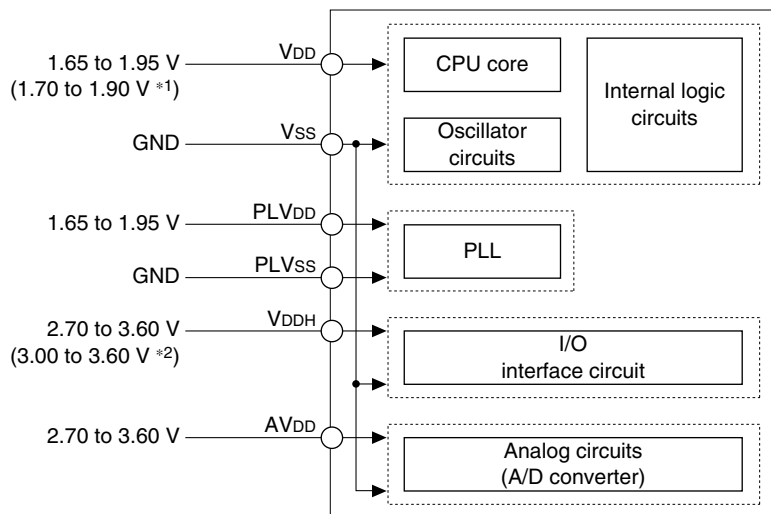
This section explains the operating voltage of the S1C33E08.

I.4.1 Power Supply Pins

The S1C33E08 has the power supply pins shown in Table I.4.1.1.

Table I.4.1.1 Power Supply Pins

Pin name	Pin No.		Function
	QFP	PFBGA	
V _{DD}	23,38,59,86,96,123	D7,E6,E7,F10,F11,G4,G11,H4,H5,K7,N2,N8	Power supply (+) for core (1.8 V)
V _{DDH}	16,30,67,91,105,128	D8,E8,E9,F4,F5,G5,G10,H10,H11,K8,K9,L9	Power supply (+) for I/O (3.3 V)
V _{SS}	7,41,62,81,101,117,135	D4,D5,D6,D11,E5,E10,E11,J5,J10,J11,K5,K6,K10,K11,N3,N9	Power supply (-); GND
PLV _{DD}	44	P5	Power supply (+) for PLL (PLV _{DD} = V _{DD})
PLV _{SS}	42	P4	Power supply (-) for PLL (PLV _{SS} = V _{SS})
AV _{DD}	47	P6	Power supply (+) for analog system and AIN0–AIN4 (3.3 V, AV _{DD} = V _{DDH})



*1 When the ceramic oscillator circuit is used

*2 When the USB function controller is used

Figure I.4.1.1 Power Supply System

I.4.2 Operating Voltage (V_{DD} , V_{SS})

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

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

or

$V_{DD} = 1.70 \text{ V to } 1.90 \text{ V}$ ($1.8 \text{ V} \pm 0.10 \text{ V}$, $V_{SS} = \text{GND}$) when the ceramic oscillator circuit is used

Note: The S1C33E08 QFP package has six V_{DD} pins and seven V_{SS} pins; the PFBGA package has 12 V_{DD} pins and 16 V_{SS} pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.

I.4.3 Power Supply for PLL (PLV_{DD} , PLV_{SS})

The PLL power supply pins (PLV_{DD} , PLV_{SS}) are provided separately from the V_{DD} and V_{SS} pins in order that the digital circuits do not affect the PLL circuit. Supply the same voltage level as the V_{DD} to the PLV_{DD} pin.

$PLV_{DD} = V_{DD}$, $PLV_{SS} = V_{SS}$

Noise on the PLL power lines decrease the PLL output precision, so use a stabilized power supply and make the board pattern with consideration given to that.

I.4.4 Power Supply for I/O Interface (V_{DDH})

The V_{DDH} voltage is used for interfacing with external I/O signals. For the output interface of the S1C33E08, the V_{DDH} voltage is used as high level and the V_{SS} voltage as low level. The V_{SS} pin is used for the ground common with V_{DD} . The following voltage is enabled for V_{DDH} :

$V_{DDH} = 2.70 \text{ V to } 3.60 \text{ V}$ ($V_{SS} = \text{GND}$) when the USB function controller is not used

or

$V_{DDH} = 3.00 \text{ V to } 3.60 \text{ V}$ ($3.3 \text{ V} \pm 0.3 \text{ V}$, $V_{SS} = \text{GND}$) when the USB function controller is used

Notes:

- The S1C33E08 QFP package has six V_{DDH} pins; the PFBGA package has 12 V_{DDH} pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.
- When an external clock is input to the MCLKI or RTC_CLKI pin, the clock signal level must be V_{DD} .

I.4.5 Power Supply for Analog Circuits (AV_{DD})

The analog power supply pin (AV_{DD}) is provided separately from the V_{DD} and V_{DDH} pins in order that the digital circuits do not affect the analog circuit (A/D converter). The AV_{DD} 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_{DD} :

$AV_{DD} = 2.70 \text{ V to } 3.60 \text{ V}$ ($3.0/3.3 \text{ V} \pm 0.3 \text{ V}$, $V_{SS} = \text{GND}$)

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

I.4.6 Precautions on Power Supply

Power-on sequence

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

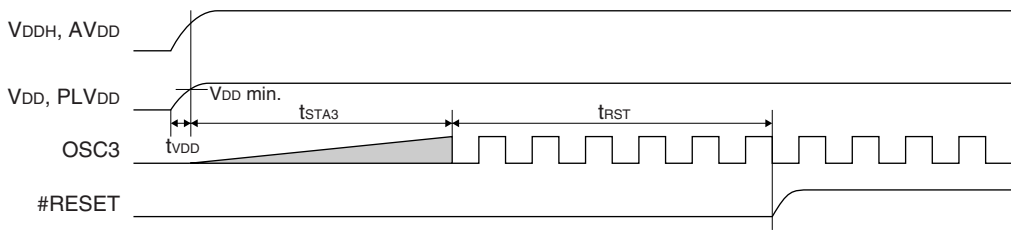


Figure I.4.6.1 Power-On Sequence

- (1) t_{VDD} : Elapsed time until the power supply stabilizes after power-on
Supply power in the following sequence (or simultaneously).
Power-on: V_{DD} and PLV_{DD} (Internal) \rightarrow V_{DDH} and AV_{DD} (I/O) \rightarrow Apply the input signal
- (2) t_{STA3} : Time at which OSC3 oscillation starts
- (3) t_{RST} : Minimum reset pulse width
Time at which the clock supplied to the chip 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 \rightarrow AV_{DD} and V_{DDH} (I/O) \rightarrow PLV_{DD} and V_{DD} (Internal)

Latch-up

The CMOS device may be in the latch-up condition. This is the phenomenon caused by conduction of the parasitic PNP junction (thyristor) contained in the CMOS IC, resulting in a large current between V_{DD} and V_{SS} 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 V_{DD} 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 V_{DD} and V_{SS} 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 V_{SS} . 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 V_{DD} , V_{DDH} , AV_{DD} , or V_{SS} .
- (4) No outputs should be shorted.

THIS PAGE IS BLANK.

I.5 CPU Core and Bus Architecture

The S1C33E08 contains the C33 PE Core as its core processor.

The C33 PE (Processor Element) Core is a Seiko Epson original 32-bit RISC-type core processor for the S1C33 Family microprocessors. Based on the C33 STD Core CPU features, some useful C33 ADV Core functions/instructions were added and some of the infrequently used ones in general applications are removed to realize a high cost-performance core unit with high processing speed.

The C33 PE Core has been designed with optimization for embedded applications (full RTL design) in mind to short development time and to reduce cost.

As the principal instructions are object-code compatible with the C33 STD Core CPU, the software assets that the user has accumulated in the past can be effectively utilized.

For details of the C33 PE Core, refer to the “S1C33 Family C33 PE Core Manual.”

I.5.1 Features of the C33 PE Core

Processor type

- Seiko Epson original 32-bit RISC processor
- 32-bit internal data processing
- Contains a 32-bit × 8-bit multiplier

Operating-clock frequency

- DC to 66 MHz or higher (depending on the processor model and process technology)

Instruction set

- | | |
|-----------------------------------|--|
| • Code length | 16-bit fixed length |
| • Number of instructions | 125 |
| • Execution cycle | Main instructions executed in one cycles |
| • Extended immediate instructions | Immediate extended up to 32 bits |
| • Multiplication instructions | Multiplications for 16 × 16 and 32 × 32 bits supported |

Register set

- 32-bit general-purpose registers
- 32-bit special registers

Memory space and external bus

- Instruction, data, and I/O coexisting linear space
- Up to 4G bytes of memory space
- Harvard architecture using separated instruction bus and data bus

Interrupts

- Reset, NMI, and 240 external interrupts supported
- Four software exceptions
- Three instruction execution exceptions
- Direct branching from vector table to interrupt handler routine

Power-down mode

- HALT mode
- SLEEP mode

I.5.2 CPU Registers

The C33 PE Core contains 16 general-purpose registers and 8 special registers.

Special registers		General-purpose registers	
	bit 31	bit 31	bit 0
#15	PC	#15	R15
#11	DBBR	#14	R14
#10	IDIR	#13	R13
#8	TTBR	#12	R12
#3	AHR	#11	R11
#2	ALR	#10	R10
#1	SP	#9	R9
#0	PSR	#8	R8
		#7	R7
		#6	R6
		#5	R5
		#4	R4
		#3	R3
		#2	R2
		#1	R1
		#0	R0

Figure I.5.2.1 Registers

I.5.3 Instruction Set

The C33 PE Core instruction set consists of the function-extended instruction set of the C33 STD Core CPU and the new instructions, in addition to the conventional S1C33-series instructions. Some instructions of the C33 STD Core CPU are deleted. As the C33 PE Core is object-code compatible with the C33 STD Core CPU, software assets can be transported from the S1C33 series to the C33 PE model easily, with minimal modifications required.

All of the instruction codes are fixed to 16 bits in length which, combined with pipelined processing, allows most important instructions to be executed in one cycle. For details, refer to the “S1C33 Family C33 PE Core Manual.”

Table I.5.3.1 S1C33-Series-Compatible Instructions

Classification	Mnemonic	Function	
Arithmetic operation	add	$\%rd, \%rs$	Addition between general-purpose registers
		$\%rd, imm6$	Addition of a general-purpose register and immediate
		$\%sp, imm10$	Addition of SP and immediate (with immediate zero-extended)
	adc	$\%rd, \%rs$	Addition with carry between general-purpose registers
	sub	$\%rd, \%rs$	Subtraction between general-purpose registers
		$\%rd, imm6$	Subtraction of general-purpose register and immediate
		$\%sp, imm10$	Subtraction of SP and immediate (with immediate zero-extended)
	sbc	$\%rd, \%rs$	Subtraction with carry between general-purpose registers
	cmp	$\%rd, \%rs$	Arithmetic comparison between general-purpose registers
		$\%rd, sign6$	Arithmetic comparison of general-purpose register and immediate (with immediate zero-extended)
	mlt.h	$\%rd, \%rs$	Signed integer multiplication (16 bits \times 16 bits \rightarrow 32 bits)
	mltu.h	$\%rd, \%rs$	Unsigned integer multiplication (16 bits \times 16 bits \rightarrow 32 bits)
	mlt.w	$\%rd, \%rs$	Signed integer multiplication (32 bits \times 32 bits \rightarrow 64 bits)
mltu.w	$\%rd, \%rs$	Unsigned integer multiplication (32 bits \times 32 bits \rightarrow 64 bits)	
Branch	jrgt	$sign8$	PC relative conditional jump Branch condition: $!Z \ \& \ !(N \wedge V)$
	jrgt.d		Delayed branching possible
	jrge	$sign8$	PC relative conditional jump Branch condition: $!(N \wedge V)$
	jrge.d		Delayed branching possible
	jrlt	$sign8$	PC relative conditional jump Branch condition: $N \wedge V$
	jrlt.d		Delayed branching possible
	jrle	$sign8$	PC relative conditional jump Branch condition: $Z \ \ N \wedge V$
	jrle.d		Delayed branching possible
	jrugt	$sign8$	PC relative conditional jump Branch condition: $!Z \ \& \ !C$
	jrugt.d		Delayed branching possible
	jruge	$sign8$	PC relative conditional jump Branch condition: $!C$
	jruge.d		Delayed branching possible
	jrult	$sign8$	PC relative conditional jump Branch condition: C
	jrult.d		Delayed branching possible
	jrule	$sign8$	PC relative conditional jump Branch condition: $Z \ \ C$
	jrule.d		Delayed branching possible
	jqreq	$sign8$	PC relative conditional jump Branch condition: Z
	jqreq.d		Delayed branching possible
	jrne	$sign8$	PC relative conditional jump Branch condition: $!Z$
	jrne.d		Delayed branching possible
	jp	$sign8$	PC relative jump Delayed branching possible
	jp.d	$\%rb$	Absolute jump Delayed branching possible
call	$sign8$	PC relative subroutine call Delayed call possible	
call.d	$\%rb$	Absolute subroutine call Delayed call possible	
ret		Subroutine return	
ret.d		Delayed return possible	
reti		Return from interrupt or exception handling	
ret.d		Return from the debug processing routine	
int	$imm2$	Software exception	
brk		Debug exception	

I S1C33E08 SPECIFICATIONS: CPU CORE AND BUS ARCHITECTURE

Classification	Mnemonic	Function	
Data transfer	ld.b	$\%rd, \%rs$	General-purpose register (byte) → general-purpose register (sign-extended)
		$\%rd, [\%rb]$	Memory (byte) → general-purpose register (sign-extended)
		$\%rd, [\%rb] +$	Postincrement possible
		$\%rd, [\%sp+imm6]$	Stack (byte) → general-purpose register (sign-extended)
		$[\%rb], \%rs$	General-purpose register (byte) → memory
		$[\%rb] +, \%rs$	Postincrement possible
	ld.ub	$[\%sp+imm6], \%rs$	General-purpose register (byte) → stack
		$\%rd, \%rs$	General-purpose register (byte) → general-purpose register (zero-extended)
		$\%rd, [\%rb]$	Memory (byte) → general-purpose register (zero-extended)
	ld.h	$\%rd, [\%rb] +$	Postincrement possible
		$\%rd, [\%sp+imm6]$	Stack (byte) → general-purpose register (zero-extended)
		$\%rd, \%rs$	General-purpose register (halfword) → general-purpose register (sign-extended)
		$\%rd, [\%rb]$	Memory (halfword) → general-purpose register (sign-extended)
		$\%rd, [\%rb] +$	Postincrement possible
		$\%rd, [\%sp+imm6]$	Stack (halfword) → general-purpose register (sign-extended)
	ld.uh	$[\%rb], \%rs$	General-purpose register (halfword) → memory
		$[\%rb] +, \%rs$	Postincrement possible
		$[\%sp+imm6], \%rs$	General-purpose register (halfword) → stack
		$\%rd, \%rs$	General-purpose register (halfword) → general-purpose register (zero-extended)
		$\%rd, [\%rb]$	Memory (halfword) → general-purpose register (zero-extended)
		$\%rd, [\%rb] +$	Postincrement possible
	ld.w	$\%rd, [\%sp+imm6]$	Stack (halfword) → general-purpose register (zero-extended)
		$\%rd, \%rs$	General-purpose register (word) → general-purpose register
		$\%rd, sign6$	Immediate → general-purpose register (sign-extended)
$\%rd, [\%rb]$		Memory (word) → general-purpose register	
$\%rd, [\%rb] +$		Postincrement possible	
$\%rd, [\%sp+imm6]$		Stack (word) → general-purpose register	
$[\%rb], \%rs$		General-purpose register (word) → memory	
$[\%rb] +, \%rs$		Postincrement possible	
System control	$[\%sp+imm6], \%rs$	General-purpose register (word) → stack	
	nop	No operation	
	halt	HALT	
Immediate extension	slp	SLEEP	
	ext	$imm13$	Extend operand in the following instruction
Bit manipulation	btst	$[\%rb], imm3$	Test a specified bit in memory data
	bclr	$[\%rb], imm3$	Clear a specified bit in memory data
	bset	$[\%rb], imm3$	Set a specified bit in memory data
	bnot	$[\%rb], imm3$	Invert a specified bit in memory data
Other	swap	$\%rd, \%rs$	Byte-wise swap on byte boundary in word
	pushn	$\%rs$	Push general-purpose registers $\%rs-\%r0$ onto the stack
	popn	$\%rd$	Pop data for general-purpose registers $\%rd-\%r0$ off the stack

The symbols in the above table each have the meanings specified below.

Table I.5.3.2 Symbol Meanings

Symbol	Description
$\%rs$	General-purpose register, source
$\%rd$	General-purpose register, destination
$\%ss$	Special register, source
$\%sd$	Special register, destination
$[\%rb]$	General-purpose register, indirect addressing
$[\%rb] +$	General-purpose register, indirect addressing with postincrement
$\%sp$	Stack pointer
$imm2, imm4, imm3,$ $imm5, imm6, imm10,$ $imm13$	Unsigned immediate (numerals indicating bit length)
$sign6, sign8$	However, numerals in shift instructions indicate the number of bits shifted, while those in bit manipulation indicate bit positions.
	Signed immediate (numerals indicating bit length)

Table I.5.3.3 Function Extended Instructions

Classification	Mnemonic	Function	Extended function	
Logical operation	and	$\%rd, \%rs$	Logical AND between general-purpose registers	The V flag is cleared after the instruction has been executed.
		$\%rd, sign6$	Logical AND of general-purpose register and immediate	
	or	$\%rd, \%rs$	Logical OR between general-purpose registers	
		$\%rd, sign6$	Logical OR of general-purpose register and immediate	
	xor	$\%rd, \%rs$	Exclusive OR between general-purpose registers	
		$\%rd, sign6$	Exclusive OR of general-purpose register and immediate	
	not	$\%rd, \%rs$	Logical inversion between general-purpose registers (1's complement)	
		$\%rd, sign6$	Logical inversion of general-purpose register and immediate (1's complement)	
Shift and rotate	srl	$\%rd, \%rs$	Logical shift to the right (Bits 0–31 shifted as specified by the register)	For rotate/shift operation, it has been made possible to shift 9–31 bits.
		$\%rd, imm5$	Logical shift to the right (Bits 0–31 shifted as specified by immediate)	
	sll	$\%rd, \%rs$	Logical shift to the left (Bits 0–31 shifted as specified by the register)	
		$\%rd, imm5$	Logical shift to the left (Bits 0–31 shifted as specified by immediate)	
	sra	$\%rd, \%rs$	Arithmetic shift to the right (Bits 0–31 shifted as specified by the register)	
		$\%rd, imm5$	Arithmetic shift to the right (Bits 0–31 shifted as specified by immediate)	
	sla	$\%rd, \%rs$	Arithmetic shift to the left (Bits 0–31 shifted as specified by the register)	
		$\%rd, imm5$	Arithmetic shift to the left (Bits 0–31 shifted as specified by immediate)	
	rr	$\%rd, \%rs$	Rotate to the right (Bits 0–31 rotated as specified by the register)	
		$\%rd, imm5$	Rotate to the right (Bits 0–31 rotated as specified by immediate)	
	rl	$\%rd, \%rs$	Rotate to the left (Bits 0–31 rotated as specified by the register)	
		$\%rd, imm5$	Rotate to the left (Bits 0–31 rotated as specified by immediate)	
Data transfer	ld.w	$\%rd, \%ss$	Special register (word) → general-purpose register	The number of special registers that can be used to load data has been increased.
		$\%sd, \%rs$	General-purpose register (word) → special register	

Table I.5.3.4 Instructions Added to the C33 PE Core

Classification	Mnemonic		Function
Branch	jpr	%rb	PC relative jump
	jpr.d		Delayed branching possible
System control	psrset	imm5	Set a specified bit in PSR
	psrclr	imm5	Clear a specified bit in PSR
Coprocessor control	ld.c	%rd, imm4	Load data from coprocessor
	ld.c	imm4, %rs	Store data in coprocessor
	do.c	imm6	Execute coprocessor
	ld.cf		Load C, V, Z, and N flags from coprocessor
Other	swaph	%rd, %rs	Bitwise swap on halfword boundary in word
	push	%rs	Push single general-purpose register
	pop	%rd	Pop single general-purpose register
	pushs	%ss	Push special registers %ss-ALR onto the stack
	pop	%sd	Pop data for special registers %sd-ALR off the stack

Table I.5.3.5 Instructions Removed

Classification	Mnemonic		Function
Arithmetic operation	div0s	%rs	First step in signed integer division
	div0u	%rs	First step in unsigned integer division
	div1	%rs	Execution of step division
	div2s	%rs	Data correction for the result of signed integer division 1
	div3s		Data correction for the result of signed integer division 2
Other	mirror	%rd, %rs	Bitwise swap every byte in word
	mac	%rs	Multiply-accumulate operation 16 bits × 16 bits + 64 bits → 64 bits
	scan0	%rd, %rs	Search for bits whose value = 0
	scan1	%rd, %rs	Search for bits whose value = 1

I.5.4 Trap Table

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

After an initial reset, the TTBR register is set to 0xC00000.

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

Bits 9 to 0 in the TTBR register are fixed at 0. Therefore, the trap table starting address always begins with a 1KB boundary address.

Table I.5.4.1 Trap Table

Vector number (Hex address)	Exception/interrupt name (peripheral circuit)	Cause of exception/interrupt	IDMA Ch.	Priority
0(Base)	Reset	Low input to the reset pin	–	1
1	reserved	–	–	–
2(Base+8)	ext exception	ext instruction (illegal use)	–	4
3(Base+0C)	Undefined instruction exception	Undefined instruction	–	3
4–5	reserved	–	–	–
6(Base+18)	Address misaligned exception	Memory access instruction	–	2
0x60000	Debugging exception	brk instruction, etc.	–	5
7(Base+1C)	NMI	Low input to the #NMI pin or watchdog timer overflow	–	6
8–10	reserved	–	–	–
11(Base+2C)	Illegal interrupt exception	Occurrence of illegal interrupt from ITC	–	High ↑
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)	Intelligent DMA	Intelligent DMA, end of transfer	–	
27–29	reserved	–	–	
30(Base+78)	16-bit timer 0	Timer 0 compare-match B	7	
31(Base+7C)		Timer 0 compare-match A	8	
32–33	reserved	–	–	
34(Base+88)	16-bit timer 1	Timer 1 compare-match B	9	
35(Base+8C)		Timer 1 compare-match A	10	
36–37	reserved	–	–	
38(Base+98)	16-bit timer 2	Timer 2 compare-match B	11	
39(Base+9C)		Timer 2 compare-match A	12	
40–41	reserved	–	–	
42(Base+A8)	16-bit timer 3	Timer 3 compare-match B	13	
43(Base+AC)		Timer 3 compare-match A	14	
44–45	reserved	–	–	
46(Base+B8)	16-bit timer 4	Timer 4 compare-match B	15	
47(Base+BC)		Timer 4 compare-match A	16	
48–49	reserved	–	–	
50(Base+C8)	16-bit timer 5	Timer 5 compare-match B	17	
51(Base+CC)		Timer 5 compare-match A	18	
52–55	reserved	–	–	Low ↓

I S1C33E08 SPECIFICATIONS: CPU CORE AND BUS ARCHITECTURE

Vector number (Hex address)	Exception/interrupt name (peripheral circuit)	Cause of exception/interrupt	IDMA Ch.	Priority
56(Base+E0)	Serial interface Ch.0	Receive error	–	High ↑
57(Base+E4)		Receive buffer full	23	
58(Base+E8)		Transmit buffer empty	24	
59	reserved	–	–	
60(Base+F0)	Serial interface Ch.1	Receive error	–	
61(Base+F4)		Receive buffer full	25	
62(Base+F8)		Transmit buffer empty	26	
63(Base+FC)	A/D converter	Result out of range (upper-limit and lower-limit)	–	
64(Base+100)		End of conversion	27	
65(Base+104)	RTC	1/64 second, 1 second, 1 minuet, or 1 hour 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)	reserved	–	–	
73(Base+124)	LCDC	End of frame	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	
79–80	reserved	–	–	
81(Base+144)	SPI	Receive DMA request	36	
82(Base+148)		Transmit DMA request	37	
83	reserved	–	–	
84(Base+150)	Port input interrupt 8	Edge (rising or falling) or level (High or Low)	38	
	SPI	SPI interrupt (D[1:0]/0x3003C4 = 0x10)		
85(Base+154)	Port input interrupt 9	Edge (rising or falling) or level (High or Low)	39	
	USB PDREQ	USB DMA request (D[3:2]/0x3003C4 = 0x10)		
86(Base+158)	Port input interrupt 10	Edge (rising or falling) or level (High or Low)	40	
	USB	USB interrupt (D[5:4]/0x3003C4 = 0x10)		
87(Base+15C)	Port input interrupt 11	Edge (rising or falling) or level (High or Low)	41	
	DCSIO	DCSIO interrupt (D[7:6]/0x3003C4 = 0x10)		
88(Base+160)	Port input interrupt 12	Edge (rising or falling) or level (High or Low)	42	
89(Base+164)	Port input interrupt 13	Edge (rising or falling) or level (High or Low)	43	
90(Base+168)	Port input interrupt 14	Edge (rising or falling) or level (High or Low)	44	
91(Base+16C)	Port input interrupt 15	Edge (rising or falling) or level (High or Low)	45	
92–93	reserved	–	–	
94(Base+178)	I ² S interface	I ² S FIFO empty	46	↓ Low
95–107	reserved	–	–	

- Notes:**
- The MP3 decoder BIOS (MP3 decoder module) uses the HSDMA Ch.0, Ch.1 and I²S and their interrupts for output data requests. Therefore, these DMA channels and interrupts cannot be used in the user program.
 - IDMA Ch.19–22, 32, 47–53 are reserved.

I.5.5 Power-Down Mode

The C33 PE Core supports two power-down modes: HALT and SLEEP modes.

HALT mode

Program execution is halted at the same time that the C33 PE Core executes the `halt` instruction, and the processor enters HALT mode.

HALT mode commonly turns off only the C33 PE Core operation. See Section III.1.11, “Standby Modes,” for details.

SLEEP mode

Program execution is halted at the same time the C33 PE Core executes the `slp` instruction, and the processor enters SLEEP mode.

SLEEP mode commonly turns off the C33 PE Core and on-chip peripheral circuit operations, thereby it significantly reduces the current consumption in comparison to the HALT mode. See Section III.1.11, “Standby Modes,” for details.

Canceling HALT or SLEEP mode

Initial reset, maskable external interrupts, NMI, and debug exceptions are commonly used for canceling HALT and SLEEP modes.

The interrupt enable/disable status set in the processor does not affect the cancellation of HALT or SLEEP mode even if an interrupt signal is used as the cancellation. In other words, interrupt signals are able to cancel HALT and SLEEP modes even if the IE flag in PSR or the interrupt enable bits in the interrupt controller (depending on the implementation) are set to disable interrupts.

When the processor is taken out of HALT or SLEEP mode using an interrupt that has been enabled (by the interrupt controller and IE flag), the corresponding interrupt handler routine is executed. Therefore, when the interrupt handler routine is terminated by the `reti` instruction, the processor returns to the instruction next to `halt` or `slp`.

When the interrupt has been disabled, the processor restarts the program from the instruction next to `halt` or `slp` after the processor is taken out of HALT or SLEEP mode.

I.5.6 Debug Mode

The C33 PE Core has debug mode to assist in software development by the user.

The debug mode provides the following functions:

- **Instruction break**
A debug exception is generated before the set instruction address is executed. An instruction break can be set at three addresses.
- **Data break**
A debug exception is generated when the set address is accessed for read or write. A data break can be set at only one address.
- **Single step**
A debug exception is generated every instruction executed.
- **Forcible break**
A debug exception is generated by an external input signal.
- **Bus break**
A debug exception is generated when the data of the selected bus matches the set value.
- **Bus trace**
The value of the selected bus is traced.
- **PC trace**
The status of instruction execution by the processor is traced.

When a debug exception occurs, the processor performs the following processing:

- (1) Suspends the instruction currently being executed.
A debug exception is generated at the end of the E stage of the currently executed instruction, and is accepted at the next rise of the system clock.
- (2) Saves the contents of the PC and R0, in that order, to the addresses specified below.
PC → 0x00060008
R0 → 0x0006000C
- (3) Loads the debug exception vector located at the address 0x00060000 to PC and branches to the debug exception handler routine.

In the exception handler routine, the `retd` instruction should be executed at the end of processing to return to the suspended instruction. When returning from the exception by the `retd` instruction, the processor restores the saved data in order of the R0 and the PC.

Neither hardware interrupts nor NMI interrupts are accepted during a debug exception.

I.5.7 Bus Architecture

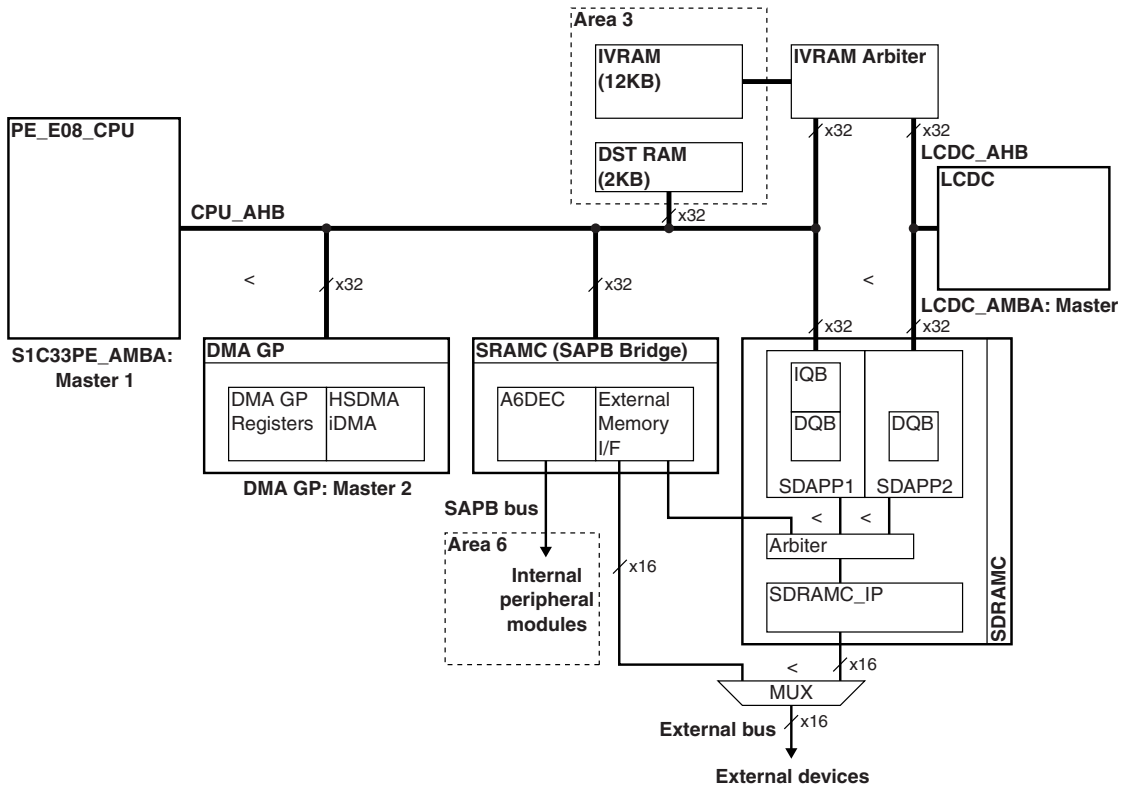


Figure I.5.7.1 S1C33E08 Bus Architecture

I.5.7.1 32-bit High-Speed Bus

Since the IVRAM (internal video RAM) or an external SDRAM may be simultaneously accessed from the CPU and the LCDC, the S1C33E08 adopts a dual 32-bit high-speed bus system that consists of the CPU_AHB bus and LCDC_AHB bus to reduce bus occupancy by one bus master.

The C33 PE Core, DMAC, SRAMC, instruction/data caches in SDRAMC, IVRAM arbitrer and DST RAM in Area 3 are connected to the CPU_AHB bus. Usually the C33 PE Core is the bus master of the CPU_AHB bus and the ownership of the bus is delegated to the DMAC when a DMA request is generated.

The LCDC, IVRAM arbitrer and LCDC data cache in SDRAMC are connected to the LCDC_AHB bus. The LCDC_AHB bus master is always the LCDC.

The IVRAM arbitrer is connected to both the CPU_AHB and LCDC_AHB buses and it arbitrates accesses to the IVRAM from the CPU and LCDC. The LCDC has higher priority than the CPU for the IVRAM access authorization as it must refresh the LCD display. For details of the IVRAM arbitrer, see Section VIII.2, “IVRAM and IVRAM Arbitrer.”

Likewise the SDRAMC is connected to the both buses allowing the CPU and LCDC to access the external SDRAM. Also in this case the LCDC has higher priority. The bus arbitrer in the SDRAMC arbitrates the ownership of the bus between the CPU and LCDC. For details of the bus arbitrer, see Section II.4.3, “Bus Arbitrer.”

I.5.7.2 SAPB Bus

The SAPB bus is used to access the internal peripheral modules located in Area 6. The SRAMC functions as a bridge between the CPU_AHB bus and SAPB bus.

I.5.7.3 External Bus

The S1C33E08 external bus is configured with a 25-bit address bus and a 16-bit data bus.

The external SDRAM is accessed via the SDRAMC and other devices are accessed via the SRAMC. The bus arbiter in the SDRAMC arbitrates the access authorization for the external bus.

For more information on the external memory, see Section I.6, “Memory Map,” Section II.3, “SRAM Controller (SRAMC),” and Section II.4, “SDRAM Controller (SDRAMC).”

I.5.8 Chip ID

The S1C33E08 has chip ID bits shown below that allow the application software to identify CPU type, model, and chip version.

Core ID Bits (D[7:0]/0x20000)

These bits provide an 8-bit ID code that indicates the chip core type.

ID	Chip Core Type
0x02	C33 standard macro core (C33 STD Core)
0x03	C33 mini-macro core
0x04	C33 advanced macro core (C33 ADV Core)
0x05	C33 PE Core
0x06	C33 PE little endian core

The S1C33E08 has adopted the C33 PE little endian core, so the chip core ID is 0x06.

Product Series ID Bits (D[7:0]/0x20001)

These bits provide an 8-bit ID code that indicates the product series of the S1C33 Family.

ID	Product Series
0x03	S1C333xx Series
0x04	S1C334xx Series
0x0E	S1C33Exx Series
0x15	S1C33Lxx Series

The product series ID of the S1C33E08 is 0x0E.

Model ID Bits (D[7:0]/0x20002)

These bits provide an 8-bit ID code that indicates the model.

The model ID of the S1C33E07/E08 is 0x07.

Version Bits (D[7:0]/0x20003)

These bits provide an 8-bit ID code that indicates the version number.

0x22 is a version number.

THIS PAGE IS BLANK.

I.6 Memory Map

Figure I.6.1 shows a memory map of the entire address space of the S1C33E08. Figure I.6.2 shows a memory map of internal memory and the internal I/O space of the S1C33E08.

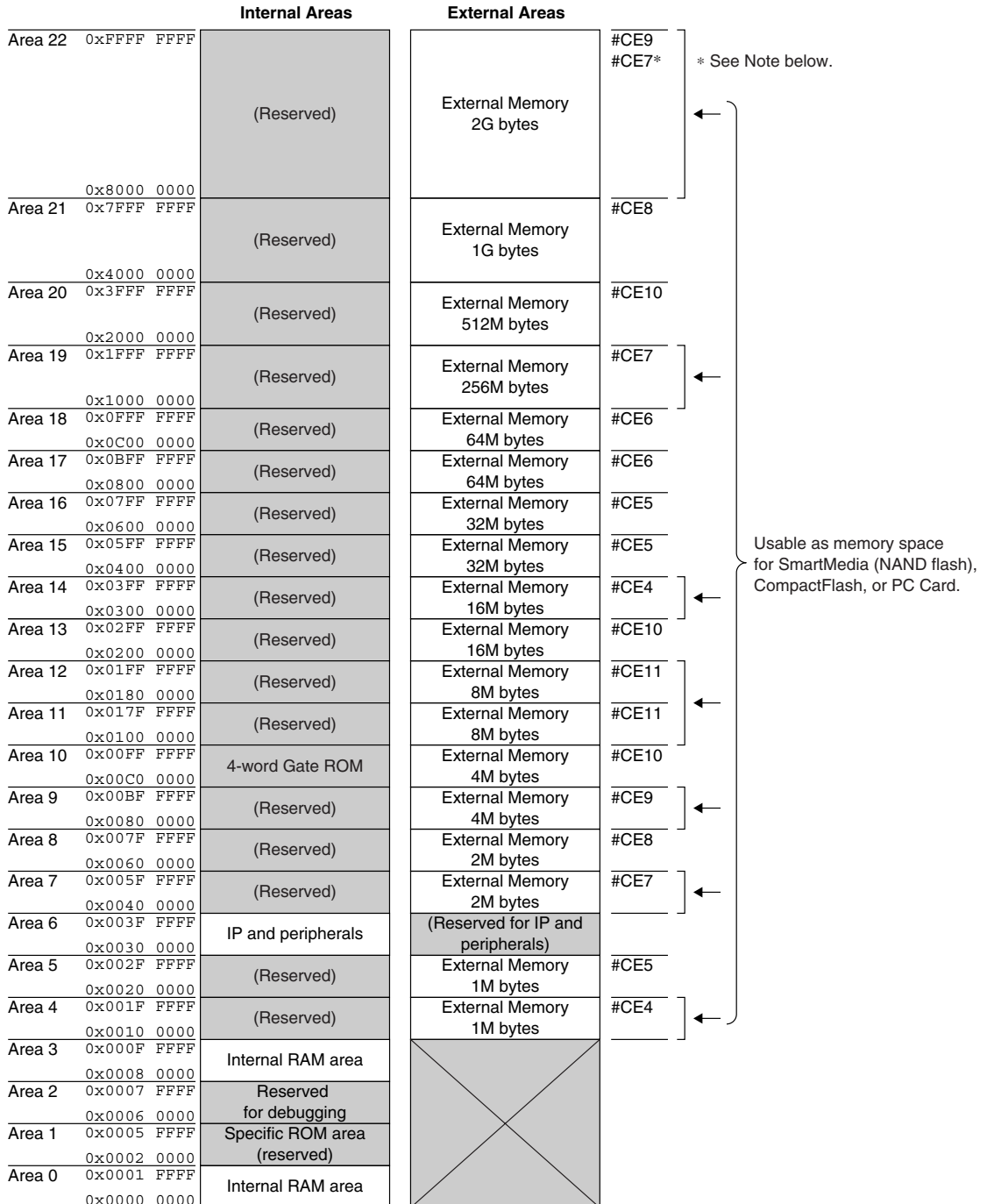


Figure I.6.1 General Memory Map

Notes:

- Area 22 is assigned to #CE9 in default settings. Note that Area 22 will be reassigned to #CE7 when the SDRAMC is enabled.

- A minimum 20K bytes of external RAM is required for using the MP3 decoder BIOS.

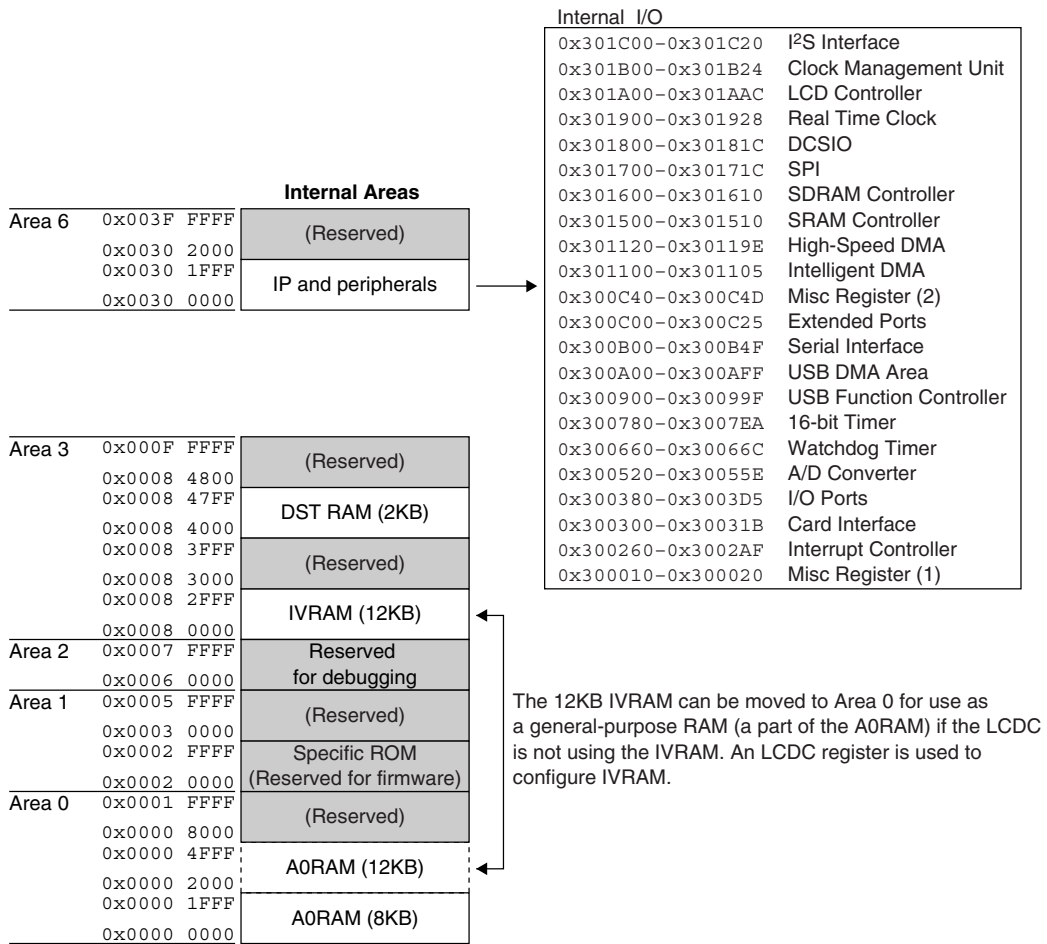


Figure I.6.2 Internal Area Map (when MP3 decoder BIOS is not used)

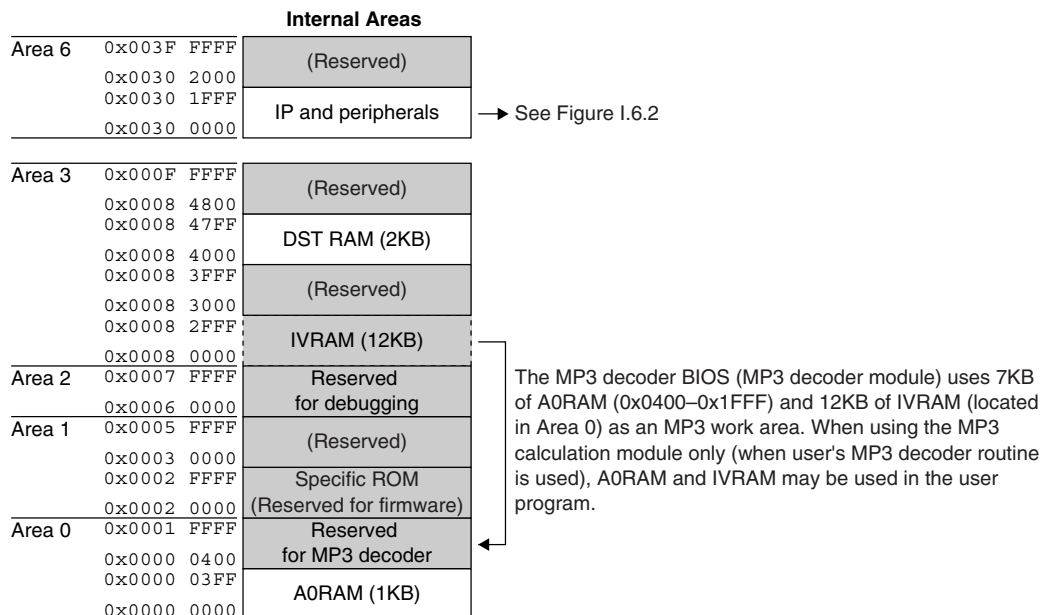


Figure I.6.3 Internal Area Map (when MP3 decoder BIOS is used)

The following describes the area configuration of the S1C33E08.

I.6.1 Boot Address and Gate ROM

When the chip is powered on or reset, the boot address is set to 0xC00000 (initial value of TTBR) by the C33 PE Core. In the S1C33E08, a 4-word Gate ROM is located at address 0xC00000 in Area 10 (internal area), and the vectors written to it are used to start a boot sequence regardless whether an external memory is connected to Area 10 or not. The vector written in the Gate ROM, which boots up the system, is determined by the input status of the BOOT pins. Also the #CE10 pin is used to set the boot mode. By setting these pins, the S1C33E08 boots up from a NAND Flash, SPI-EEPROM, PC (RS-232C), or an external NOR-Flash/ROM. The boot sequence has been programmed in the specific ROM located in Area 1.

Tables I.6.1.1 and I.6.1.2 list the pin configurations and the boot mode selected. For more information on boot, see Appendix D, “Boot.”

Table I.6.1.1 Boot Mode Configuration (PFBGA-180pin or die model)

BOOT1 pin	BOOT0 pin	#CE10	Boot mode	Boot code start address	MBR execution address
1	1	1 (Input)	SPI-EEPROM	0x20010 in the internal ROM (Area 1)	0x0 in AORAM
		0 (Input)	PC RS232C		
1	0	Output	NOR Flash/external ROM	0x2000C in the internal ROM (Area 1)	Depending on the contents in 0xC00000
0	1	–	reserved	–	–
0	0	1 (Input)	Large-page NAND Flash (> 1024 + 32 bytes/page)	0x20004 in the internal ROM (Area 1)	0x0 in AORAM
		0 (Input)	Small-page NAND Flash (< 1024 + 32 bytes/page)		

Table I.6.1.2 Boot Mode Configuration (QFP-144pin model)

BOOT1 pin	#CE10	Boot mode	Boot code start address	MBR execution address
1	Output	NOR Flash/external ROM	0x2000C in the internal ROM (Area 1)	Depending on the contents in 0xC00000
0	1 (Input)	Large-page NAND Flash (> 1024 + 32 bytes/page)	0x20004 in the internal ROM (Area 1)	0x0 in AORAM
	0 (Input)	Small-page NAND Flash (< 1024 + 32 bytes/page)		

* The QFP-144pin does not have the BOOT0 pin due to the limited number of pins available (BOOT0 has been pulled down to Vss inside the IC).

I.6.2 Area 0 (AORAM)

Area 0 contains an 8K-byte high-speed RAM (AORAM). Its location address ranges from 0x0 to 0x1FFF.

Moreover, the S1C33E08 has a built-in 12K-byte RAM (IVRAM) to be used as a video RAM for the LCDC, which is located in Area 3 by default. If IVRAM is not required for use as a video RAM (e.g. when the LCDC is not used or when an external SDRAM is used as a video RAM), IVRAM can be moved to Area 0 to expand AORAM into 20K bytes. The LCDC provides the control bit IRAM (D0/0x301A64) for this switch over. When IRAM = 0 (default), IVRAM is located in Area 3 and when IRAM = 1, IVRAM is located at 0x2000 to 0x4FFF in Area 0 (immediately following 8K-byte AORAM).

* **IRAM:** IRAM Assign Bit in the IRAM Select Register (D0/0x301A64)

Since AORAM (including IVRAM located in Area 0) is accessed directly from the CPU without passing through the AHB bus, no wait cycles are inserted. AORAM is accessed in one cycle (with no wait cycle), regardless of whether accessed in units of bytes, half-words, or words.

Moreover, due to a Harvard architecture, AORAM can be accessed simultaneously with the fetching of instructions from external memory (cache).

- Notes:**
- A0RAM cannot contain IDMA control words or be specified as the source or destination of DMA transfer.
 - When the debug monitor S5U1C330M2D1 (MON33) is used to debug, addresses 0x0 to 0xF are configured as an area for debugging. The S5U1C330M2D1 does not allow the user program to access this address range. When only the S5U1C33001H (ICD33) is used for debugging, this area can be accessed by the user.
 - The MP3 decoder BIOS (MP3 decoder module) uses 7KB of A0RAM (0x0400–0x1FFF) and 12KB of IVRAM (located in Area 0) as an MP3 work area. When using the MP3 calculation module only (when user's MP3 decoder routine is used), A0RAM and IVRAM may be used in the user program.

I.6.3 Area 1 (Specific ROM for Firmware)

Area 1 contains a 64K-byte mask ROM. This ROM is reserved for firmware to support booting and the MP3 decoder BIOS, therefore it does not contain a user program.

The MP3 decoder BIOS provides the MP3 data decoding and playback API functions. For details of the MP3 decoder, refer to Section X.1, "MP3 Decoder."

I.6.4 Area 2 (Debug Area)

Area 2 is a debugging-only area allocated for debugging resources. This area can only be accessed for writing in debug mode.

Make sure this area is not accessed from the user program or debugger.

I.6.5 Area 3 (IVRAM)

0x80000 to 0x82FFF in Area 3 is allocated to the 12K-byte IVRAM (Internal Video RAM) by the default configuration. IVRAM can be used as a video RAM for the LCDC. The LCDC and CPU access IVRAM through their respective 32-bit AHB bus and the IVRAM Arbiter that resolves bus conflicts between the LCDC and CPU. IVRAM located in Area 3 can be accessed in a minimum of 2-wait cycles.

As described in Section I.6.2, IVRAM can be located in Area 0 for use as a high-speed general-purpose RAM that allows no wait access when it is not used as a video RAM.

- Notes:**
- A program cannot be executed in the IVRAM located in Area 3.
 - When using the MP3 decoder BIOS (MP3 decoder module), the IVRAM will be relocated to Area 0 to be used as an MP3 work area by an API function. In this case, an external SDRAM must be used as the VRAM.

I.6.6 Area 3 (DST RAM)

0x84000 to 0x847FF in Area 3 is allocated to the 2K-byte DST RAM (Descriptor Table RAM). DST RAM is provided for storing IDMA control words as A0RAM cannot contain them. The memory space other than control words can be used as a general-purpose RAM and may also be specified as the source and destination of DMA transfer.

DST RAM can be accessed in a minimum of 2-wait cycles.

Note: The upper 256 bytes (0x84700 to 0x847FF) in DST RAM are reserved for use as the debugging area. The user program must be prohibited from accessing this area.

However, specify 0x84780 for the debug RAM address of the `c33 das` command in the debugger.

```
c33 das 0x60000 0x84780 1
```

I.6.7 Area 6 (I/O Area)

Area 6 is allocated to the I/O area for S1C33E08 IP and peripheral circuits.

Although Area 6 is one of the external memory areas, external memory cannot be accessed.

For details of the internal peripheral circuits mapped to this area, see the description in Chapters III to IX. For details of a control register list, see “I/O Map” in the Appendix.

I.6.8 External Memory Areas

Areas 4, 5, and 7 to 22 can be used for external memory and other external devices. Set up the SRAMC or SDRAMC according to specifications of the devices connected.

Although the internal address and internal data buses of the S1C33E08 are both 32 bits wide, the maximum external data bus width is 16 bits (D[15:0]) and the maximum external address bus width is 25 bits (A[24:0]) due to the limited number of pins available.

Note: The MP3 decoder BIOS (MP3 decoder module) uses a minimum of 20KB area in the external RAM. When using the MP3 calculation module only (when user's MP3 decoder routine is used), no external RAM is required for the MP3 decoder BIOS.

THIS PAGE IS BLANK.

I.7 Electrical Characteristics

I.7.1 Absolute Maximum Rating

(V_{SS}=PLV_{SS}=0V)

Item	Symbol	Condition	Rated value	Unit	*
Internal logic power voltage	V _{DD}		-0.3 to +2.5	V	
PLL power voltage	PLV _{DD}		-0.3 to +2.5	V	
I/O power voltage	V _{DDH}		-0.3 to +4.0	V	
Analog power voltage	AV _{DD}		-0.3 to +4.0	V	
Input voltage	V _I		-0.3 to V _{DDH} +0.5	V	
Analog input voltage	AV _{IN}		-0.3 to AV _{DD} +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	

E char

I.7.2 Recommended Operating Conditions

(V_{SS}=PLV_{SS}=0V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Internal logic power voltage	V _{DD}	Crystal or external clock	1.65	1.80	1.95	V	
		Ceramic oscillation	1.70	1.80	1.90	V	
PLL power voltage	PLV _{DD}		1.65	1.80	1.95	V	
I/O power voltage	V _{DDH}	When USB is not used	2.70	–	3.60	V	
		When USB is used	3.00	3.30	3.60	V	
Analog power voltage	AV _{DD}		2.70	–	3.60	V	
Input voltage	HV _I		V _{SS}	–	V _{DDH}	V	
	LV _I		V _{SS}	–	V _{DD}	V	
Analog input voltage	AV _{IN}		V _{SS}	–	AV _{DD}	V	
CPU operating clock frequency	f _{CPU}		–	–	60	MHz	
Bus operating clock frequency	f _{BUS}		–	–	60	MHz	
OSC3 oscillation frequency	f _{OSC3}		5	–	48	MHz	
OSC3 external input clock frequency	f _{ECLK3}		2	–	48	MHz	
OSC1 oscillation frequency	f _{OSC1}		–	32.768	–	kHz	
Operating temperature	T _a	Crystal or external clock	-40	25	85	°C	
		Ceramic oscillation	0	25	70	°C	
Input rise time (normal input)	t _{ri}		–	–	50	ns	
Input fall time (normal input)	t _{fi}		–	–	50	ns	
Input rise time (Schmitt input)	t _{ri}		–	–	5	ms	
Input fall time (Schmitt input)	t _{fi}		–	–	5	ms	

I.7.3 DC Characteristics

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

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

Note: See Section I.3.4, "Input/Output Cells and Input/Output Characteristics," for pin characteristics.

I.7.4 Current Consumption

Operating current

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

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Current consumption during CPU running (MCLK=SDCLK)	IDD1	20MHz	–	8.0	–	mA	1
		25MHz	–	10.0	–	mA	1
		33MHz	–	13.0	–	mA	1
		48MHz	–	19.0	–	mA	1
		60MHz	–	23.5	–	mA	2
Current consumption during CPU running (MCLK/SDCLK) (MCLK=1/2SDCLK)*1	IDD2	24/48MHz	–	10.0	–	mA	1
		30/60MHz	–	14.0	–	mA	2
		40/80MHz	–	20.0	–	mA	2
		45/90MHz	–	21.0	–	mA	2
Current consumption in HALT mode	IDD3	20MHz	–	1.3	–	mA	1
		25MHz	–	1.6	–	mA	1
		33MHz	–	2.1	–	mA	1
		48MHz	–	3.0	–	mA	1
		60MHz	–	4.0	–	mA	2
Current consumption in HALT mode (Operating clock=48MHz×1/n)*2	IDD4	1/1 (=48MHz)	–	3.0	–	mA	3
		1/2 (=24MHz)	–	2.2	–	mA	3
		1/4 (=12MHz)	–	1.8	–	mA	3
		1/8 (=6MHz)	–	1.6	–	mA	3
		1/16 (=3MHz)	–	1.5	–	mA	3
		1/32 (=1.5MHz)	–	1.4	–	mA	3

Current consumption measurement condition: $V_{IH}=V_{DDH}$, $V_{IL}=0V$, output pins are open, V_{DD} power current only

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

* note)

No.	OSC3	OSC1	CPU	Other peripheral circuits
1	On	Off	Normal operation *3	Stopped
2	On	Off	Normal operation *3	PLL (×2) only is used when the OSC3 frequency is 30MHz or more. Others are stopped.
3	On	Off	HALT mode	Stopped

*1: The MCLK (CPU operating clock) frequency is set to half of the SDCLK (SDRAM clock).

*2: The operating clock actually used is the MCLK divided clock specified with the CMU_CLKSEL register.

The CMU_CLKSEL register can be rewritten anytime necessary, note, however, that a hazard may occur on the clock output when the register is changed.

The lower the operating frequency, the lower the current consumption is realized. However, it affects the clock frequency supplied to the peripheral circuits, therefore, it is necessary to control the clock supply (on/off) or to reset the prescaler.

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

Current consumption in SLEEP mode

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

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Current consumption in SLEEP mode	IDS1	48MHz	–	1.0	–	μA	
	IDS2	32kHz (System clock = OSC1)	–	2.0	–	μA	

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

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

Peripheral circuit operating currents

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

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
SRAM controller operating current	ISR	When SRAM clock is supplied. (48MHz)	–	3.4	–	mA	4
SDRAM controller operating current	ISD	When SDRAM clock is supplied. (48MHz)	–	5.4	–	mA	4
DMA controller operating current	IDMA	When DMA clock is supplied. (48MHz)	–	3.9	–	mA	4
LCD controller operating current	ILCD	When LCDC clock is supplied. (48MHz)	–	5.3	–	mA	4
USB controller operating current	IUSB	Idle state when USB clock is supplied. (48MHz)	–	10.0	–	mA	4
RTC operating current	IRTC	OSC1 oscillation (32kHz)	–	2.0	–	μA	4
OSC3 operating current	IOSC3	OSC3 oscillation (48MHz)	–	1.3	–	mA	4
SSCG operating current	ISSCG	SSCG input clock (48MHz)	–	400.0	–	μA	4
PLL operating current	IPLL	PLL output clock (90MHz)	–	2.1	–	mA	5
A/D converter operating current	IAD	When A/D converter is enabled	–	260.0	–	μA	6
MP3 decoder operating current	IMP3	When MP3 decoder is on playback	–	8.0	–	mA	7

- * note 4) V_{DD} power current consumption in idle status when the clock is supplied
 5) PLV_{DD} power current consumption
 6) AV_{DD} power current consumption
 7) Except on stop or pause state

1.7.5 A/D Converter Characteristics

(Unless otherwise specified: V_{DDH}=AV_{DD}=2.7V to 3.6V, V_{DD}=1.65V to 1.95V, V_{SS}=0V, T_a=-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.

E char

A/D conversion error

V[001]h = Ideal voltage at zero-scale point (=0.5LSB)

V'[001]h = Actual voltage at zero-scale point

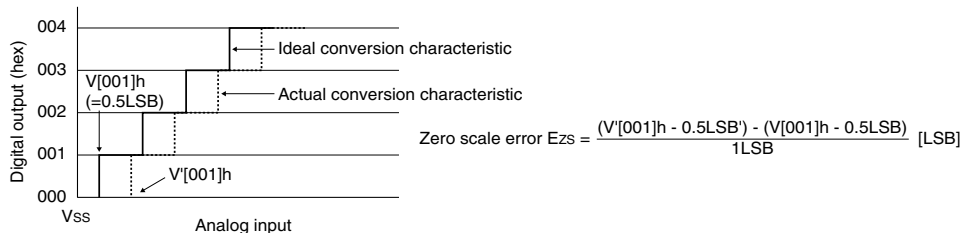
V[3FF]h = Ideal voltage at full-scale point (=1022.5LSB)

V'[3FF]h = Actual voltage at full-scale point

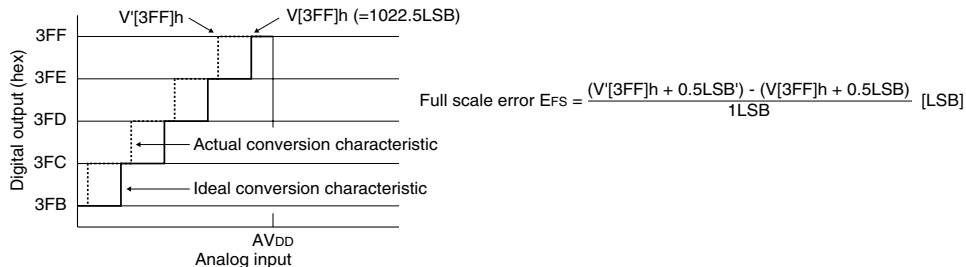
$$1\text{LSB} = \frac{AV_{DD} - V_{SS}}{2^{10} - 1}$$

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

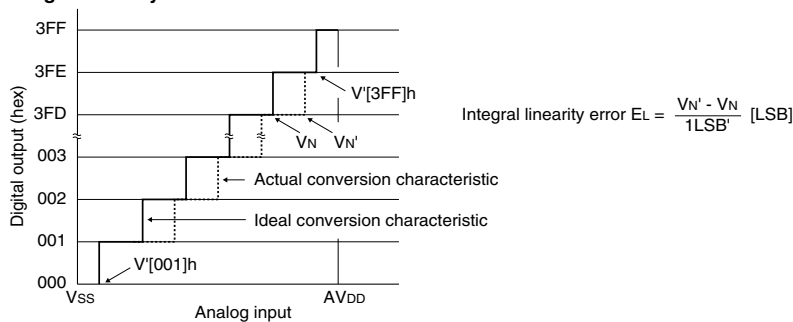
Zero scale error



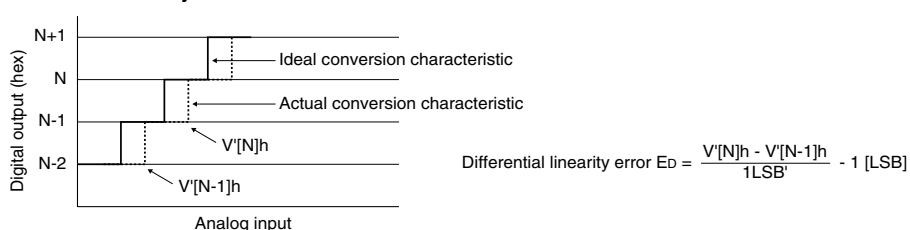
Full scale error



Integral linearity error



Differential linearity error



I.7.6 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 resistor (R_f , R_d) and capacitor (C_G , C_D) values are finally decided.

OSC1 oscillation (crystal)

(Unless otherwise specified: $V_{DD}=1.65V$ to $1.95V$, $V_{SS}=0V$, $T_a=25^\circ C$)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Oscillation start time	t_{STA1}				3	s	

OSC3 oscillation (crystal/ceramic)

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

(Unless otherwise specified: $V_{DD}=1.70V$ to $1.90V$, $V_{SS}=0V$, $T_a=25^\circ C$)

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

I.7.7 PLL Characteristics

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

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Input frequency	f _{PLLIN}		5		50	MHz	1
Output frequency	f _{PLLOUT}		20		90	MHz	2
Output stabilization time	t _{PLL}				200	μs	

* note 1) Input clock source divider: OSC3 ×1, ×1/2, ×1/3, ×1/4, ×1/5, ×1/6, ×1/7, ×1/8, ×1/9, ×1/10

2) Multiplication rate: ×1, ×2, ×4, ×8, ×10, ×15

E char

I.7.8 AC Characteristics

I.7.8.1 Symbol Description

tcyc: Bus-clock cycle time

Indicates the cycle time of the bus clock.

I.7.8.2 AC Characteristics Measurement Condition

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

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

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: Pins other than SDCLK: $C_L = 50 \text{ pF}$
 SDCLK pin: $C_L = 20 \text{ pF}$

I.7.8.3 SRAMC 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: $V_{DDH}=2.7V$ to $3.6V$, $V_{DD}=1.65V$ to $1.95V$, $V_{SS}=0V$, $T_a=-40^{\circ}C$ to $+85^{\circ}C$)

Item	Symbol	Min.	Max.	Unit	*
High-speed clock cycle time	t _{C3}	16.7	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}		25	ns	
BCLK low-level output delay time	t _{CD2}		25	ns	

E char

BCLK clock output characteristics

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

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

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

Bus access cycle

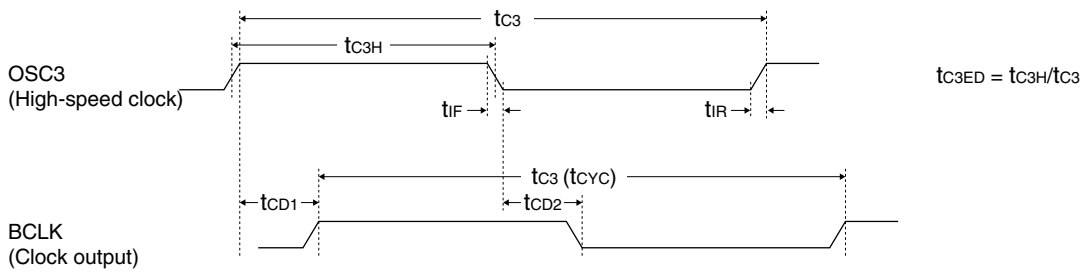
(Unless otherwise specified: $V_{DDH}=2.7V$ to $3.6V$, $V_{DD}=1.65V$ to $1.95V$, $V_{SS}=0V$, External load=50pF, $T_a=-40^{\circ}C$ to $+85^{\circ}C$)

Item	Symbol	Min.	Max.	Unit	*
CMU_CLK output delay time	t _{CD}		20	ns	
Address delay time	t _{AD}		14	ns	
#CE _x delay time	t _{CED}		14	ns	
Write delay time	t _{WRD}		14	ns	
Write data delay time	t _{WRDD}		14	ns	
Write data hold time	t _{WRDH}	-2		ns	
Read delay time	t _{RDD}		14	ns	
Read data setup time	t _{RDS}	12		ns	
Read data hold time	t _{RDDH}	0		ns	
Write signal pulse width	t _{WRW}	t _{CYC} (1+WC)-14		ns	
Read signal pulse width	t _{RDW}	t _{CYC} (1+WC)-14		ns	
Read address access time	t _{ACC}		t _{CYC} (1+WC)-26	ns	
Chip enable access time	t _{CEAC}		t _{CYC} (1+WC)-26	ns	
Read signal access time	t _{RDAC}		t _{CYC} (1+WC)-26	ns	
#WAIT setup time	t _{WTS}	12		ns	
#WAIT hold time	t _{WTH}	0		ns	

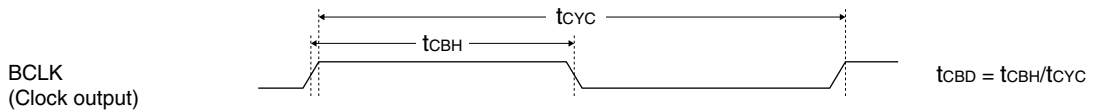
I.7.8.4 SRAMC AC Characteristic Timing Charts

Clock

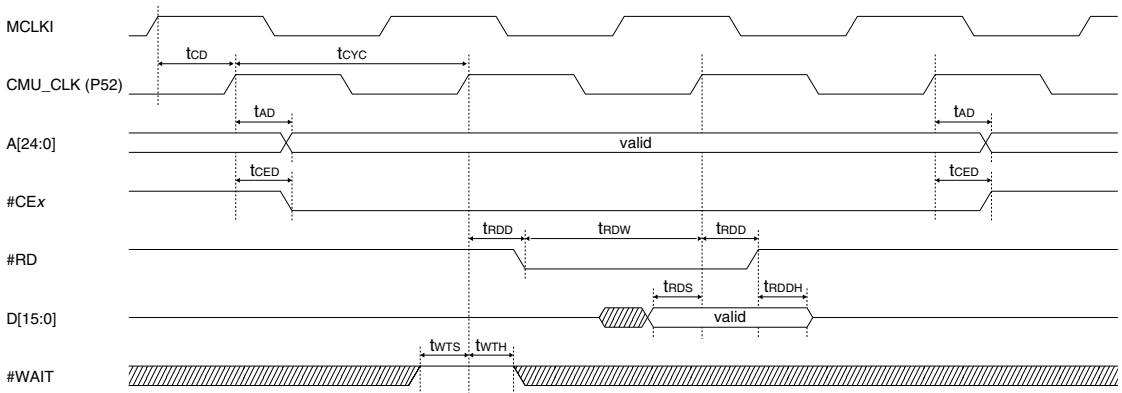
(1) When an external clock is input:



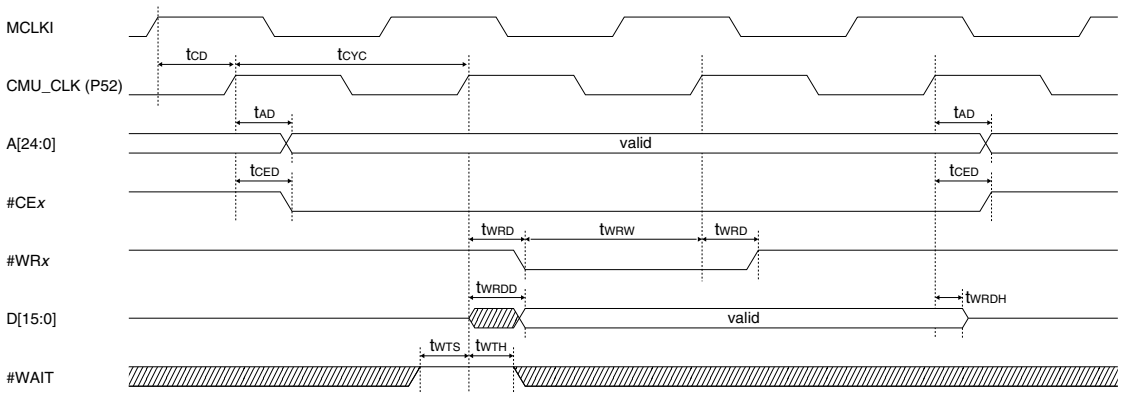
(2) When the high-speed oscillation circuit is used for the operating clock:



SRAM read cycle



SRAM write cycle



I.7.8.5 SDRAM Interface AC Characteristics

Normal mode (SDCLK = MCLK, 60MHz Max.)

(Unless otherwise specified: $V_{DDH}=2.7V$ to $3.3V$, $V_{DD}=1.65V$ to $1.95V$, $T_a=-40^{\circ}C$ to $+85^{\circ}C$
External load conditions: Address bus/data bus=50pF, SDCLK/control signals=20pF)

Item	Symbol	Min.	Max.	Unit	*
Address delay time	t _{AD}		10	ns	
Address hold time	t _{AH}	1.5		ns	
SDA10 delay time	t _{A10D}		10	ns	
SDA10 hold time	t _{A10H}	1.5		ns	
#SDCS delay time	t _{CSD}		10	ns	
#SDCS hold time	t _{CSH}	1.5		ns	
#SDRAS signal delay time	t _{RASD}		10	ns	
#SDRAS signal hold time	t _{RASH}	1.5		ns	
#SDCAS signal delay time	t _{CASD}		10	ns	
#SDCAS signal hold time	t _{CASH}	1.5		ns	
DQMH, DQML signal delay time	t _{DQMD}		10	ns	
DQMH, DQML signal hold time	t _{DQMH}	1.5		ns	
SDCKE signal delay time	t _{CKED}		10	ns	
SDCKE signal hold time	t _{CKEH}	1.5		ns	
#SDWE signal delay time	t _{WED}		10	ns	
#SDWE signal hold time	t _{WEH}	1.5		ns	
Read data setup time	t _{RDS}	4		ns	
Read data hold time	t _{RDH}	0		ns	
Write data delay time	t _{WDD}		11	ns	
Write data hold time	t _{WDH}	1.5		ns	

E char

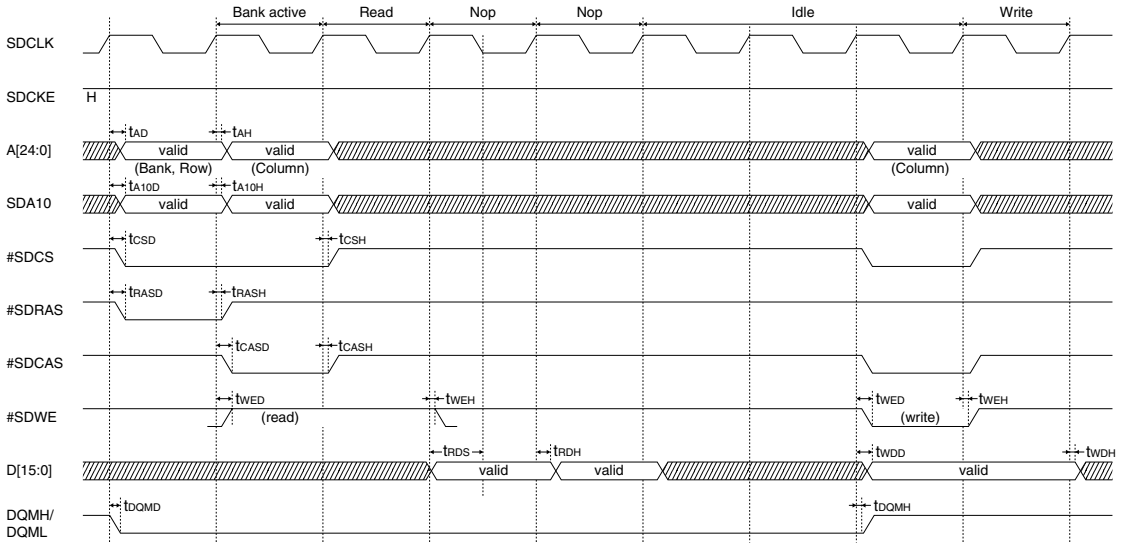
Double frequency mode (SDCLK = 2 × MCLK, 90MHz Max.)

(Unless otherwise specified: $V_{DDH}=2.7V$ to $3.3V$, $V_{DD}=1.65V$ to $1.95V$, $T_a=-40^{\circ}C$ to $+85^{\circ}C$
External load conditions: Address bus/data bus=30pF, SDCLK/control signals=20pF)

Item	Symbol	Min.	Max.	Unit	*
Address delay time	t _{AD}		8	ns	
Address hold time	t _{AH}	1.5		ns	
SDA10 delay time	t _{A10D}		8	ns	
SDA10 hold time	t _{A10H}	1.5		ns	
#SDCS delay time	t _{CSD}		8	ns	
#SDCS hold time	t _{CSH}	1.5		ns	
#SDRAS signal delay time	t _{RASD}		8	ns	
#SDRAS signal hold time	t _{RASH}	1.5		ns	
#SDCAS signal delay time	t _{CASD}		8	ns	
#SDCAS signal hold time	t _{CASH}	1.5		ns	
DQMH, DQML signal delay time	t _{DQMD}		8	ns	
DQMH, DQML signal hold time	t _{DQMH}	1.5		ns	
SDCKE signal delay time	t _{CKED}		8	ns	
SDCKE signal hold time	t _{CKEH}	1.5		ns	
#SDWE signal delay time	t _{WED}		8	ns	
#SDWE signal hold time	t _{WEH}	1.5		ns	
Read data setup time	t _{RDS}	4		ns	
Read data hold time	t _{RDH}	0		ns	
Write data delay time	t _{WDD}		8	ns	
Write data hold time	t _{WDH}	1.5		ns	

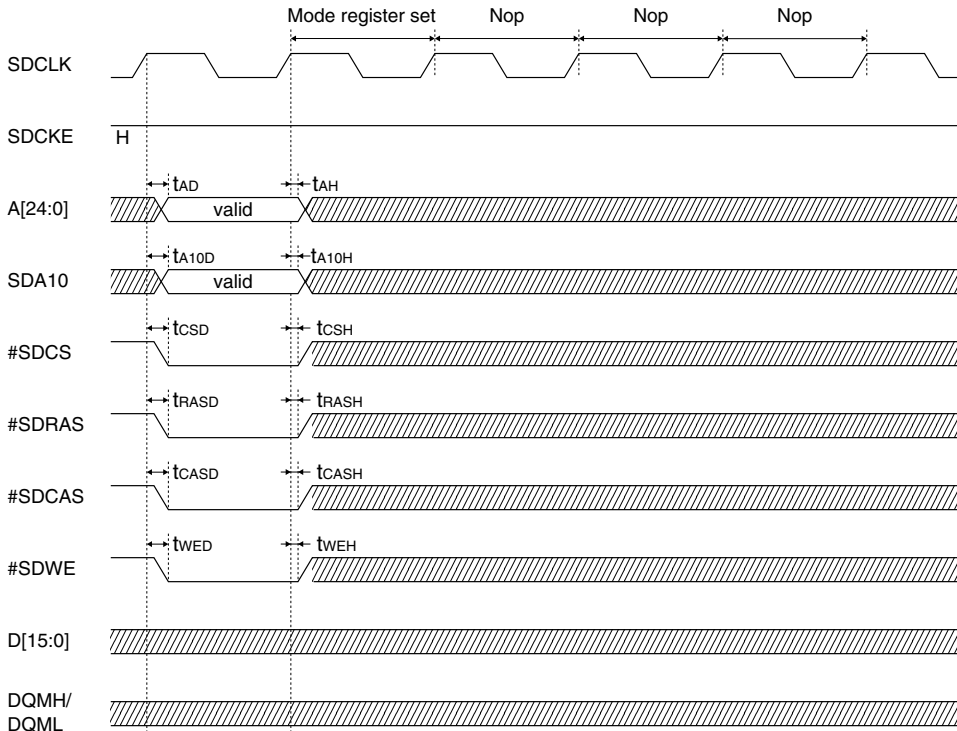
Note: All the signals change at the rising edge of the SDRAM clock.

SDRAM access cycle

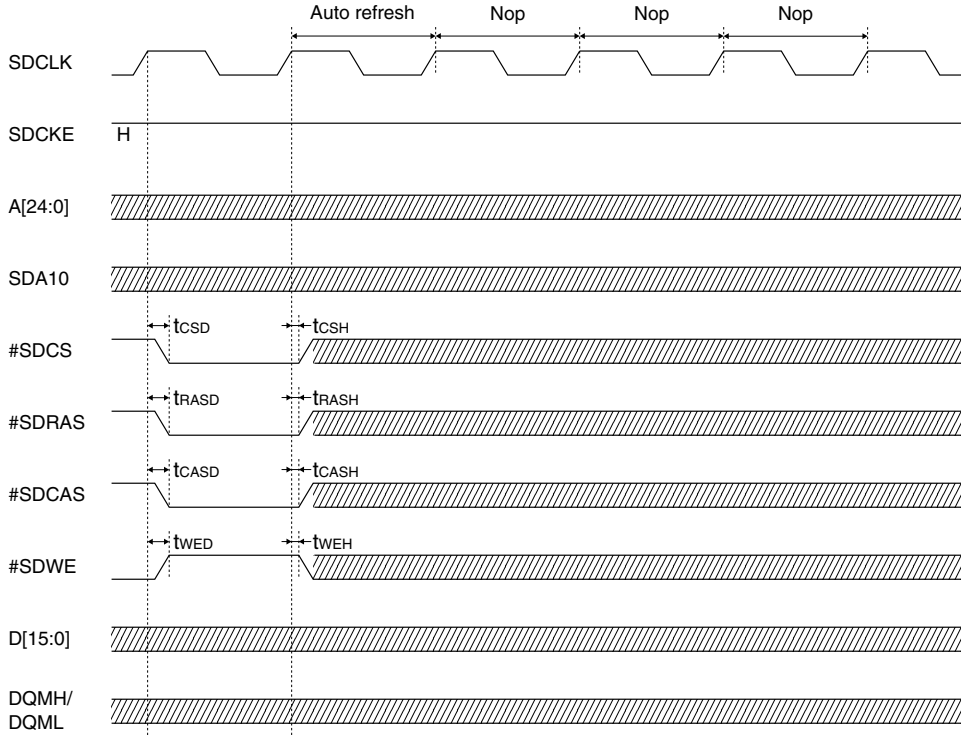


* Read: CAS latency = 2, burst length = 2 Write: single write

SDRAM mode-register-set cycle

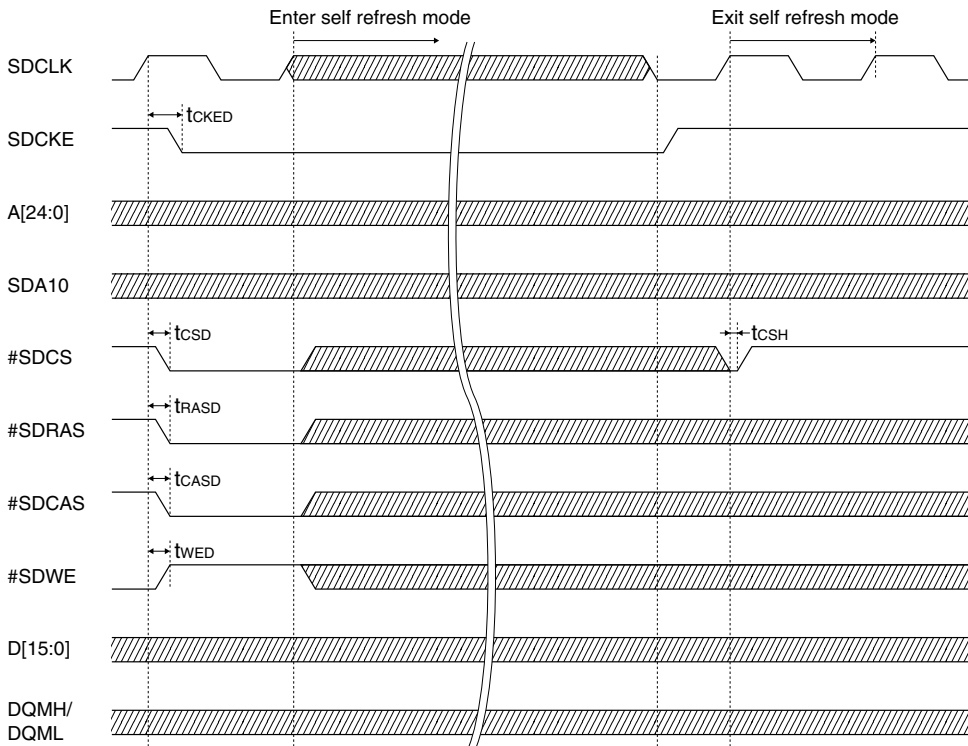


SDRAM auto-refresh cycle



* 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.

I.7.8.6 LCDC AC Characteristics

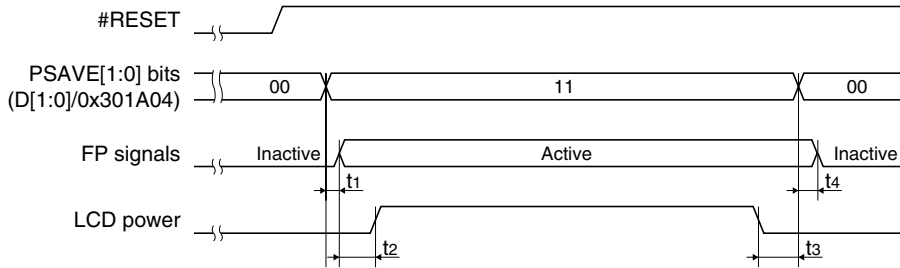
Conditions: $V_{DDH} = 2.7V$ to $3.6V$

$T_a = -40^{\circ}C$ to $85^{\circ}C$

Trise and Tfall for all outputs should be < 5 ns (10% ~ 90%)

$C_L = 60$ pF (LCD panel interface)

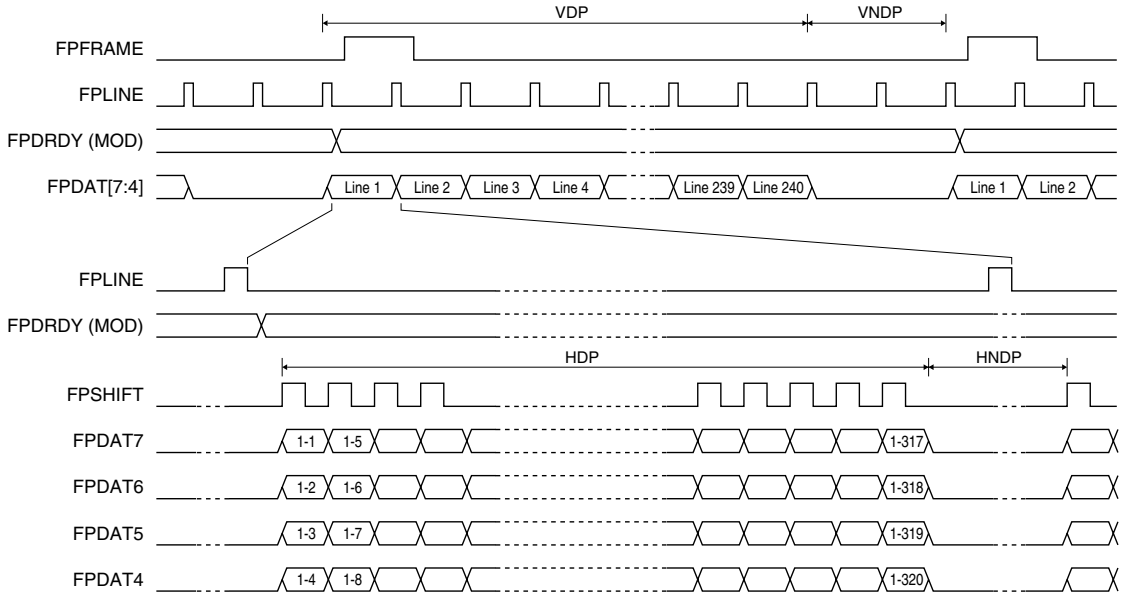
Power up/down timing



Symbol	Parameter	Min.	Typ.	Max.	Unit
t ₁	Power Save inactive to FPLINE, FPFRAME, FPSHIFT, FPDAT, FPDRDY active			1	Frame
t ₂	FPLINE, FPFRAME, FPSHIFT, FPDAT, FPDRDY active to LCD power on	1			Frame
t ₃	Power Save active to LCD power off	1			Frame
t ₄	Power Save active to FPLINE, FPFRAME, FPSHIFT, FPDAT, FPDRDY inactive			1	Frame

note) Any I/O port can be used for controlling the power supply to the LCD panel. Note, however, that the t₂ and t₃ timing conditions must be satisfied when controlling the signal.

4-bit single monochrome panel timing

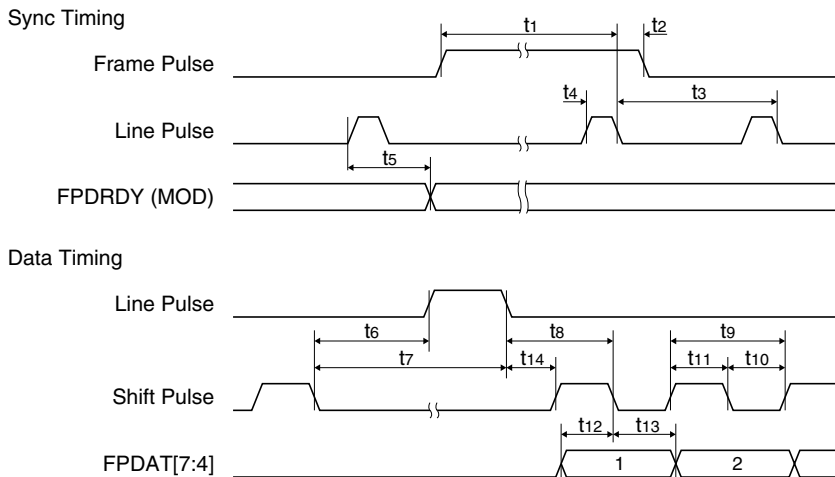


* Diagram drawn with 2 FPLINE vertical blank period

Example timing for a 320 × 240 panel

For this timing diagram FPSMASK (D29/0x301A60) is set to 1

- VDP = Vertical Display Period = $\text{VDPCNT}[9:0] + 1$ (lines)
 $\text{VDPCNT}[9:0]$ (D[9:0]/0x301A14)
- VNDP = Vertical Non-Display Period = $\text{VTCNT}[9:0] - \text{VDPCNT}[9:0]$ (lines)
 $\text{VTCNT}[9:0]$ (D[25:16]/0x301A14)
- HDP = Horizontal Display Period = $(\text{HDPCNT}[6:0] + 1) \times 8$ (Ts)
 $\text{HDPCNT}[6:0]$ (D[6:0]/0x301A10)
- HNDP = Horizontal Non-Display Period = $(\text{HTCNT}[6:0] - \text{HDPCNT}[6:0]) \times 8$ (Ts)
 $\text{HTCNT}[6:0]$ (D[22:16]/0x301A10)



Note: For this timing diagram FPSMASK (D29/0x301A60) 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

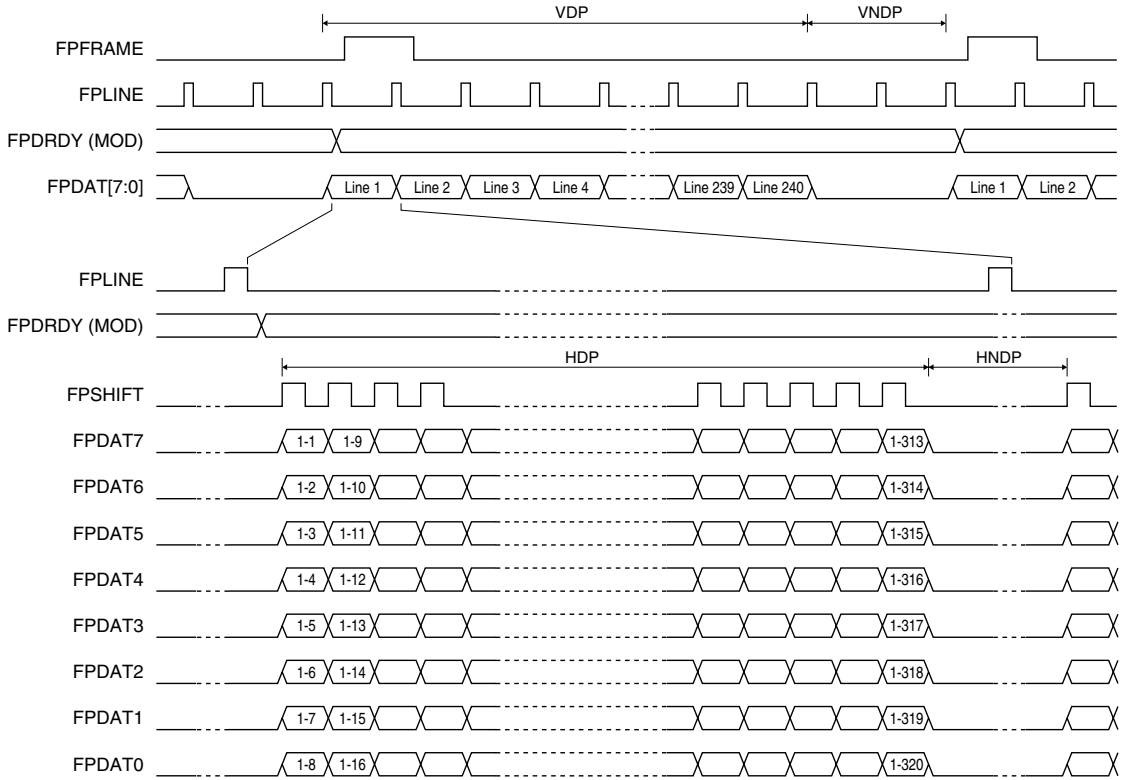
2. $t_{1min} = t_{3min} - 9$ (Ts)

3. $t_{3min} = (HTCNT[6:0] + 1) \times 8$ (Ts)

4. $t_{6min} = (HTCNT[6:0] - HDPCNT[6:0]) \times 8 + 2$ (Ts)

5. $t_{7min} = (HTCNT[6:0] - HDPCNT[6:0]) \times 8 + 11$ (Ts)

8-bit single monochrome panel timing

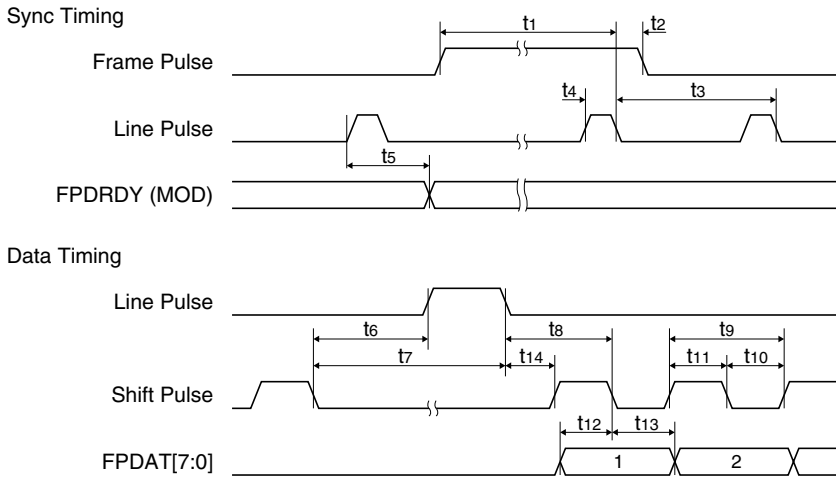


* Diagram drawn with 2 FPLINE vertical blank period

Example timing for a 320 × 240 panel

For this timing diagram FPSMASK (D29/0x301A60) is set to 1

- VDP = Vertical Display Period = $VDCNT[9:0] + 1$ (lines)
 $VDCNT[9:0]$ (D[9:0]/0x301A14)
- VNDP = Vertical Non-Display Period = $VTCNT[9:0] - VDCNT[9:0]$ (lines)
 $VTCNT[9:0]$ (D[25:16]/0x301A14)
- HDP = Horizontal Display Period = $(HDPCNT[6:0] + 1) \times 8$ (Ts)
 $HDPCNT[6:0]$ (D[6:0]/0x301A10)
- HNDP = Horizontal Non-Display Period = $(HTCNT[6:0] - HDPCNT[6:0]) \times 8$ (Ts)
 $HTCNT[6:0]$ (D[22:16]/0x301A10)



Note: For this timing diagram FPSMASK (D29/0x301A60) 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

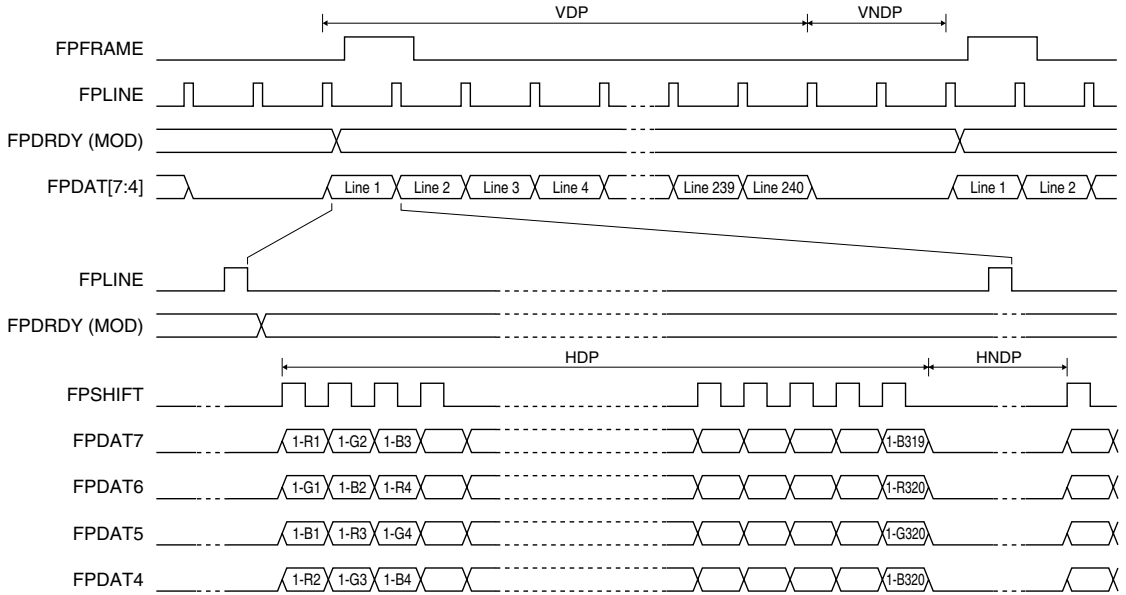
2. $t_{1min} = t_{3min} - 9$ (Ts)

3. $t_{3min} = (HTCNT[6:0] + 1) \times 8$ (Ts)

4. $t_{6min} = (HTCNT[6:0] - HDPCNT[6:0]) \times 8 + 4$ (Ts)

5. $t_{7min} = (HTCNT[6:0] - HDPCNT[6:0]) \times 8 + 13$ (Ts)

4-bit single color panel timing



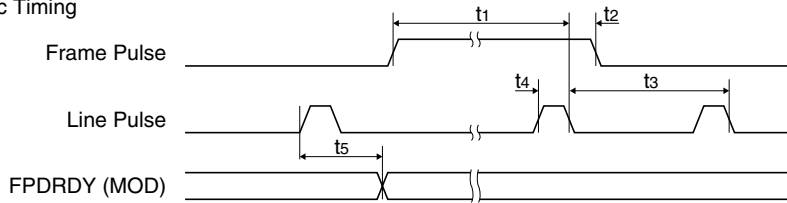
* Diagram drawn with 2 FPLINE vertical blank period

Example timing for a 320 × 240 panel

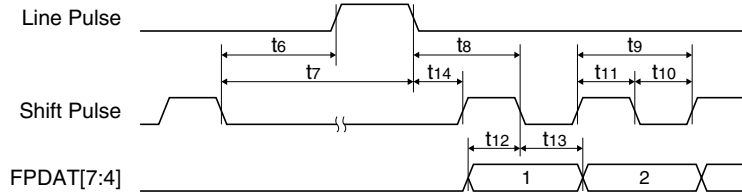
$$\begin{aligned}
 \text{VDP} &= \text{Vertical Display Period} &&= \text{VDPCNT}[9:0] + 1 \text{ (lines)} \\
 &&&\text{VDPCNT}[9:0] \text{ (D}[9:0\text{)}/0\text{x301A14)} \\
 \text{VNDP} &= \text{Vertical Non-Display Period} &&= \text{VTCNT}[9:0] - \text{VDPCNT}[9:0] \text{ (lines)} \\
 &&&\text{VTCNT}[9:0] \text{ (D}[25:16\text{)}/0\text{x301A14)} \\
 \text{HDP} &= \text{Horizontal Display Period} &&= (\text{HDPCNT}[6:0] + 1) \times 8 \text{ (Ts)} \\
 &&&\text{HDPCNT}[6:0] \text{ (D}[6:0\text{)}/0\text{x301A10)} \\
 \text{HNDP} &= \text{Horizontal Non-Display Period} &&= (\text{HTCNT}[6:0] - \text{HDPCNT}[6:0]) \times 8 \text{ (Ts)} \\
 &&&\text{HTCNT}[6:0] \text{ (D}[22:16\text{)}/0\text{x301A10)}
 \end{aligned}$$

I S1C33E08 SPECIFICATIONS: ELECTRICAL CHARACTERISTICS

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

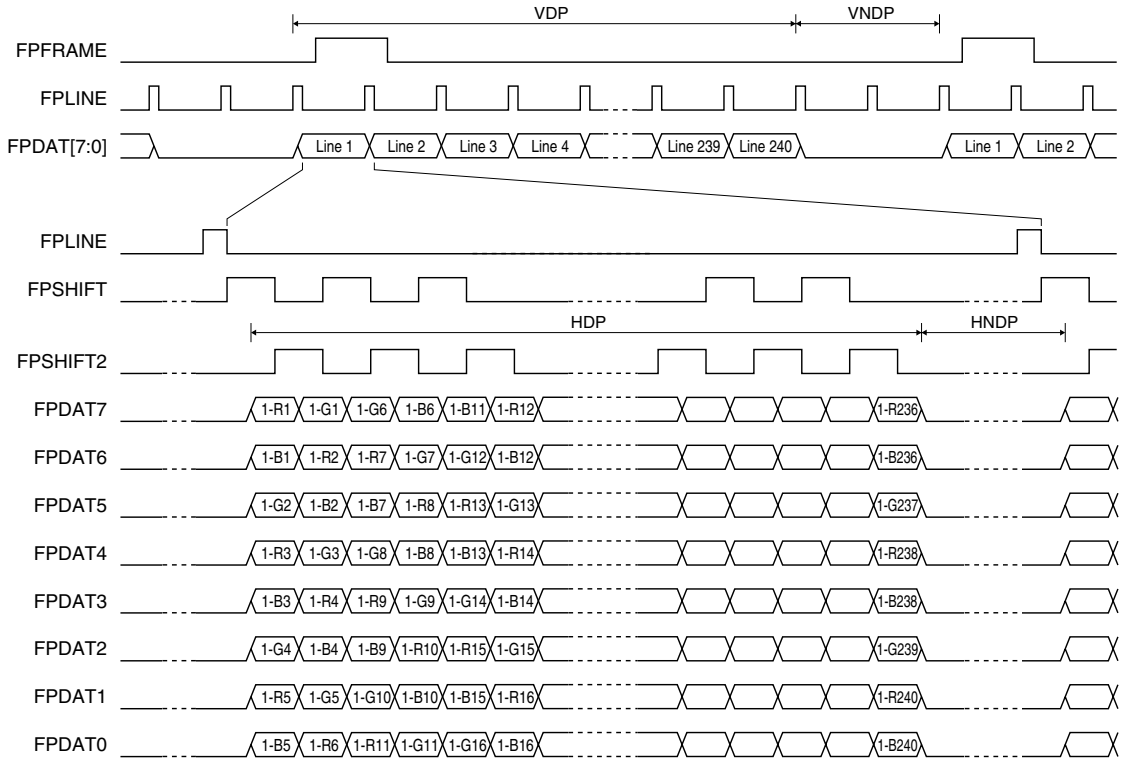
2. $t_{1min} = t_{3min} - 9 (Ts)$

3. $t_{3min} = (HTCNT[6:0] + 1) \times 8 (Ts)$

4. $t_{6min} = (HTCNT[6:0] - HDPCNT[6:0]) \times 8 + 1.5 (Ts)$

5. $t_{7min} = (HTCNT[6:0] - HDPCNT[6:0]) \times 8 + 10.5 (Ts)$

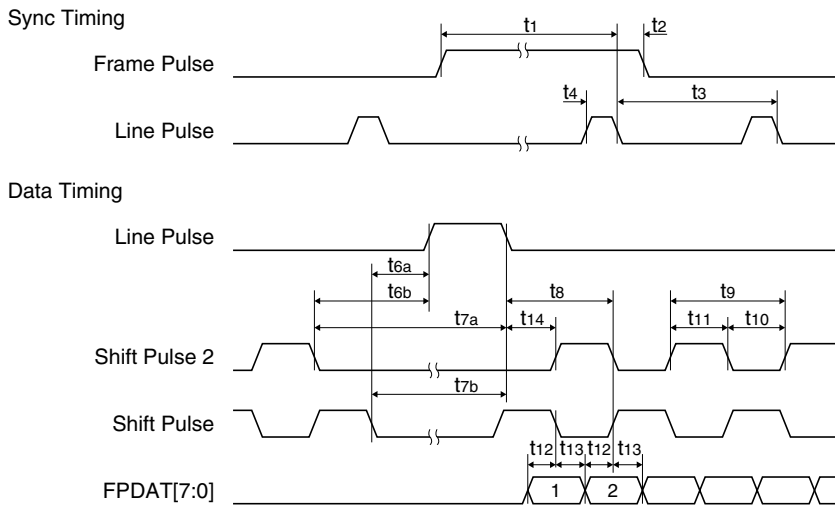
8-bit single color panel timing (Format 1)



* Diagram drawn with 2 FPLINE vertical blank period

Example timing for a 320 × 240 panel

- VDP = Vertical Display Period = $VDP_{CNT}[9:0] + 1$ (lines)
 $VDP_{CNT}[9:0]$ (D[9:0]/0x301A14)
- VNDP = Vertical Non-Display Period = $VTC_{CNT}[9:0] - VDP_{CNT}[9:0]$ (lines)
 $VTC_{CNT}[9:0]$ (D[25:16]/0x301A14)
- HDP = Horizontal Display Period = $(HDP_{CNT}[6:0] + 1) \times 8$ (Ts)
 $HDP_{CNT}[6:0]$ (D[6:0]/0x301A10)
- HNDP = Horizontal Non-Display Period = $(HTC_{CNT}[6:0] - HDP_{CNT}[6:0]) \times 8$ (Ts)
 $HTC_{CNT}[6:0]$ (D[22:16]/0x301A10)



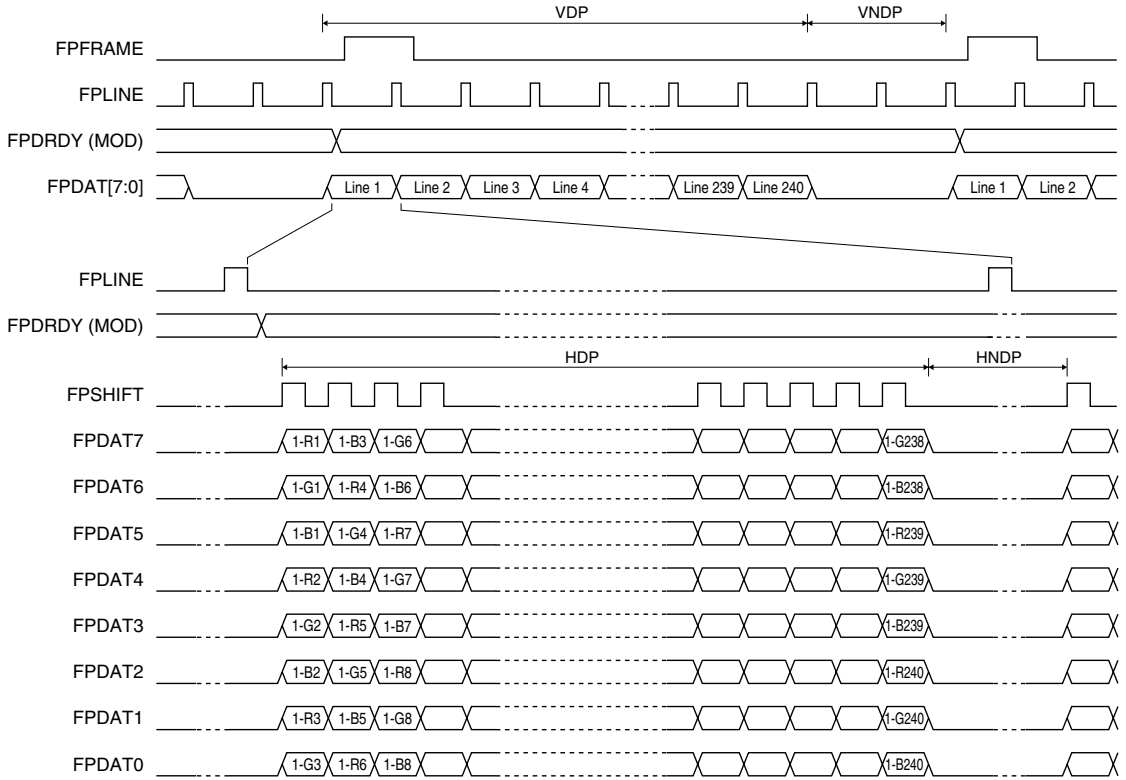
8-bit Single Color Panel AC Timing (Format 1)

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

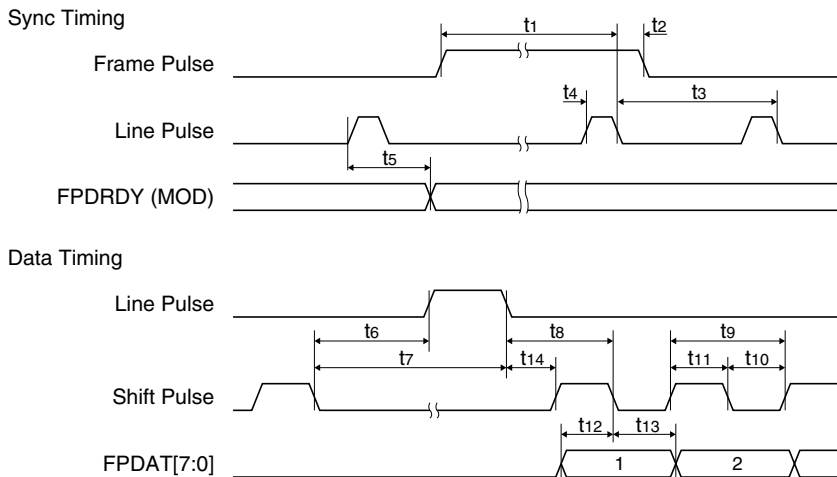
2. t_{1min} = t_{3min} - 9 (Ts)
3. t_{3min} = (HTCNT[6:0] + 1) × 8 (Ts)
4. t_{6amin} = (HTCNT[6:0] - HDPCNT[6:0]) × 8 + t₁₃ - t₁₀ + 1 (Ts)
5. t_{6bmin} = (HTCNT[6:0] - HDPCNT[6:0]) × 8 + t₁₃ + 1 (Ts)
6. t_{7amin} = (HTCNT[6:0] - HDPCNT[6:0]) × 8 + 11 (Ts)
7. t_{7amin} = (HTCNT[6:0] - HDPCNT[6:0]) × 8 + 11 - t₁₀ (Ts)

8-bit single color panel timing (Format 2)



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

- VDP = Vertical Display Period = $\text{VDPCNT}[9:0] + 1$ (lines)
 $\text{VDPCNT}[9:0]$ (D[9:0]/0x301A14)
- VNDP = Vertical Non-Display Period = $\text{VTCNT}[9:0] - \text{VDPCNT}[9:0]$ (lines)
 $\text{VTCNT}[9:0]$ (D[25:16]/0x301A14)
- HDP = Horizontal Display Period = $(\text{HDPCNT}[6:0] + 1) \times 8$ (Ts)
 $\text{HDPCNT}[6:0]$ (D[6:0]/0x301A10)
- HNDP = Horizontal Non-Display Period = $(\text{HTCNT}[6:0] - \text{HDPCNT}[6:0]) \times 8$ (Ts)
 $\text{HTCNT}[6:0]$ (D[22:16]/0x301A10)



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

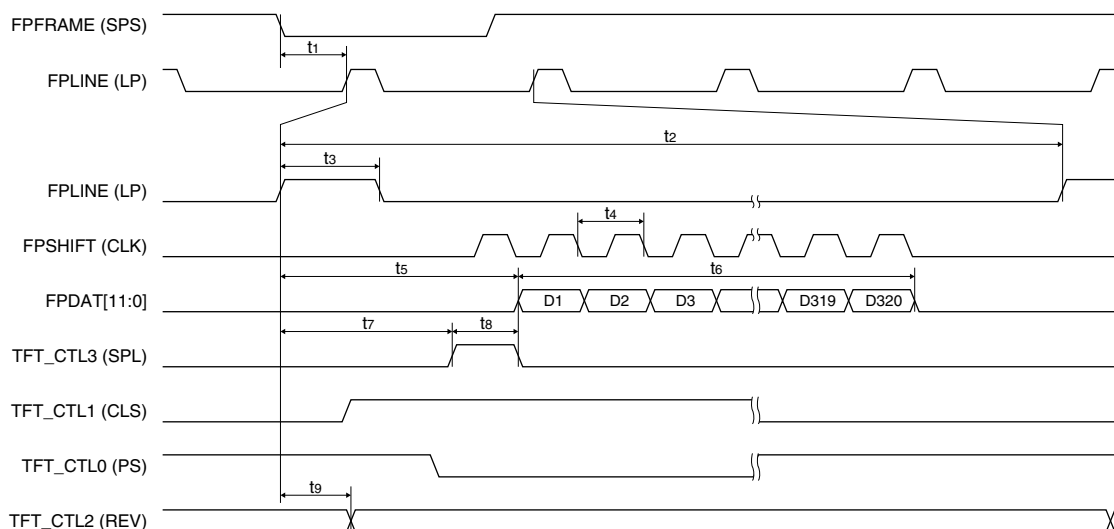
Symbol	Parameter	Min.	Typ.	Max.	Unit
t_1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t_2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t_3	Line Pulse period	note 3			
t_4	Line Pulse width	9			Ts
t_5	MOD delay from Line Pulse rising edge	1			Ts
t_6	Shift Pulse falling edge to 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. $t_{3min} = (HTCNT[6:0] + 1) \times 8$ (Ts)
4. $t_{6min} = (HTCNT[6:0] - HDPCNT[6:0]) \times 8 + 1$ (Ts)
5. $t_{7min} = (HTCNT[6:0] - HDPCNT[6:0]) \times 8 + 10$ (Ts)

12-bit generic HR-TFT panel timing

(1) Generic HR-TFT panel horizontal timing



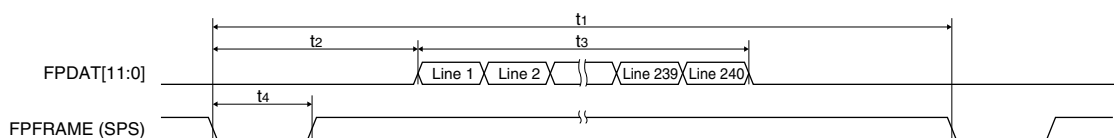
* Example timing for a 320 × 240 panel

Symbol	Parameter	Min.	Typ.	Max.	Unit
t ₁	FPLINE start position		note 2		(note 1)
t ₂	Total horizontal period	400	note 3	440	Ts
t ₃	FPLINE width		note 4		Ts
t ₄	FPSHIFT period		1		Ts
t ₅	Horizontal display start position		note 5		Ts
t ₆	Horizontal display period		note 6		Ts
t ₇	FPLINE rising edge to TFT_CTL3 rising edge		59		Ts
t ₈	TFT_CTL3 pulse width		1		Ts
t ₉	FPLINE rising edge to TFT_CTL2 change		11		Ts

note) 1. Ts = pixel clock period

- 2. $t_{1typ} = FPLST[9:0] + 1$ (Ts) FPLST[9:0] (D[25:16]/0x301A28)
- 3. $t_{2typ} = (HTCNT[6:0] + 1) \times 8$ (Ts) HTCNT[6:0] (D[22:16]/0x301A10)
- 4. $t_{3typ} = FPLWD[6:0] + 1$ (Ts) FPLWD[6:0] (D[6:0]/0x301A28)
- 5. $t_{5typ} = HDPCNT[9:0] + 1$ (Ts) HDPCNT[9:0] (D[9:0]/0x301A20)
- 6. $t_{6typ} = (HDPCNT[6:0] + 1) \times 8$ (Ts) HDPCNT[6:0] (D[6:0]/0x301A10)

(2) Generic HR-TFT panel vertical timing

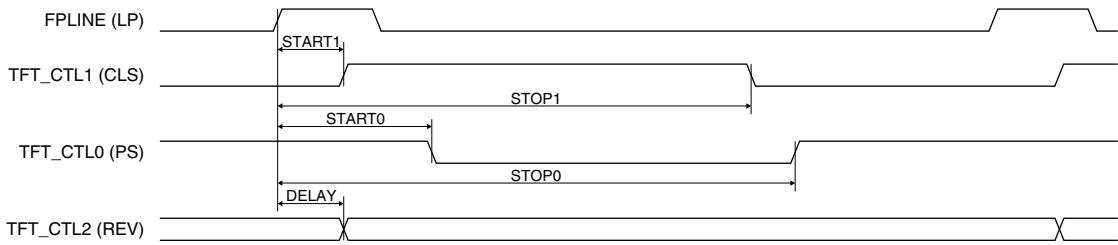


* Example timing for a 320 × 240 panel

Symbol	Parameter	Min.	Typ.	Max.	Unit
t ₁	Total vertical period	245	note 1	330	Lines
t ₂	Vertical display start position		note 2		Lines
t ₃	Vertical display period		note 3		Lines
t ₄	Vertical sync pulse width		2		Lines

- note) 1. $t_{1typ} = VTCNT[9:0] + 1$ (Lines) VTCNT[9:0] (D[25:16]/0x301A14)
- 2. $t_{2typ} = VDPCNT[9:0]$ (Lines) VDPCNT[9:0] (D[9:0]/0x301A24)
- 3. $t_{3typ} = VDPCNT[9:0] + 1$ (Lines) VDPCNT[9:0] (D[9:0]/0x301A14)

(3) Generic HR-TFT panel control signal offset timings



* When FPLST[9:0] (D[25:16]/0x301A28) = 0x0

- | | |
|-------------------------------|----------------------------------|
| START1 = CTL1ST[7:0] (Ts) | CTL1ST[7:0] (D[7:0]/0x301A44) |
| STOP1 = CTL1STP[7:0] + 1 (Ts) | CTL1STP[7:0] (D[23:16]/0x301A44) |
| START0 = CTL0ST[7:0] (Ts) | CTL0ST[7:0] (D[7:0]/0x301A48) |
| STOP0 = CTL0STP[7:0] + 1 (Ts) | CTL0STP[7:0] (D[23:16]/0x301A48) |
| DELAY = CTL2DLY[7:0] (Ts) | CTL2DLY[7:0] (D[7:0]/0x301A4C) |

I.7.9 USB DC and AC Characteristics

Input levels

(Unless otherwise specified: $V_{DD}=3.0V$ to $3.6V$, $V_{DD}=1.65V$ to $1.95V$, $T_a=-40^{\circ}C$ to $+85^{\circ}C$)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
VBUS input	V _{BUS}		4.40	–	5.25	V	1
High (driven)	V _{IH}		2.0	–	–	V	2
High (floating)	V _{IHZ}		2.7	–	3.6	V	2
Low	V _{IL}		–	–	0.8	V	2
Differential input sensitivity	V _{DI}	DP - DM	0.2	–	–	V	
Differential common mode range	V _{CM}	Include V _{DI} 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: $V_{DD}=3.0V$ to $3.6V$, $V_{DD}=1.65V$ to $1.95V$, $T_a=-40^{\circ}C$ to $+85^{\circ}C$)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Low	V _{OL}		0.0	–	0.3	V	3
High (driven)	V _{OH}		2.8	–	3.6	V	3
Output signal crossover voltage	V _{CRS}		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: $V_{DD}=3.0V$ to $3.6V$, $V_{DD}=1.65V$ to $1.95V$, $T_a=-40^{\circ}C$ to $+85^{\circ}C$)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Bus pull-up resistor on upstream facing port (idle Bus)	R _{PUI}		0.9	–	1.575	k Ω	5
Bus pull-up resistor on upstream facing port (receiving)	V _{PUA}		1.425	–	3.090	k Ω	5

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

Driver characteristics

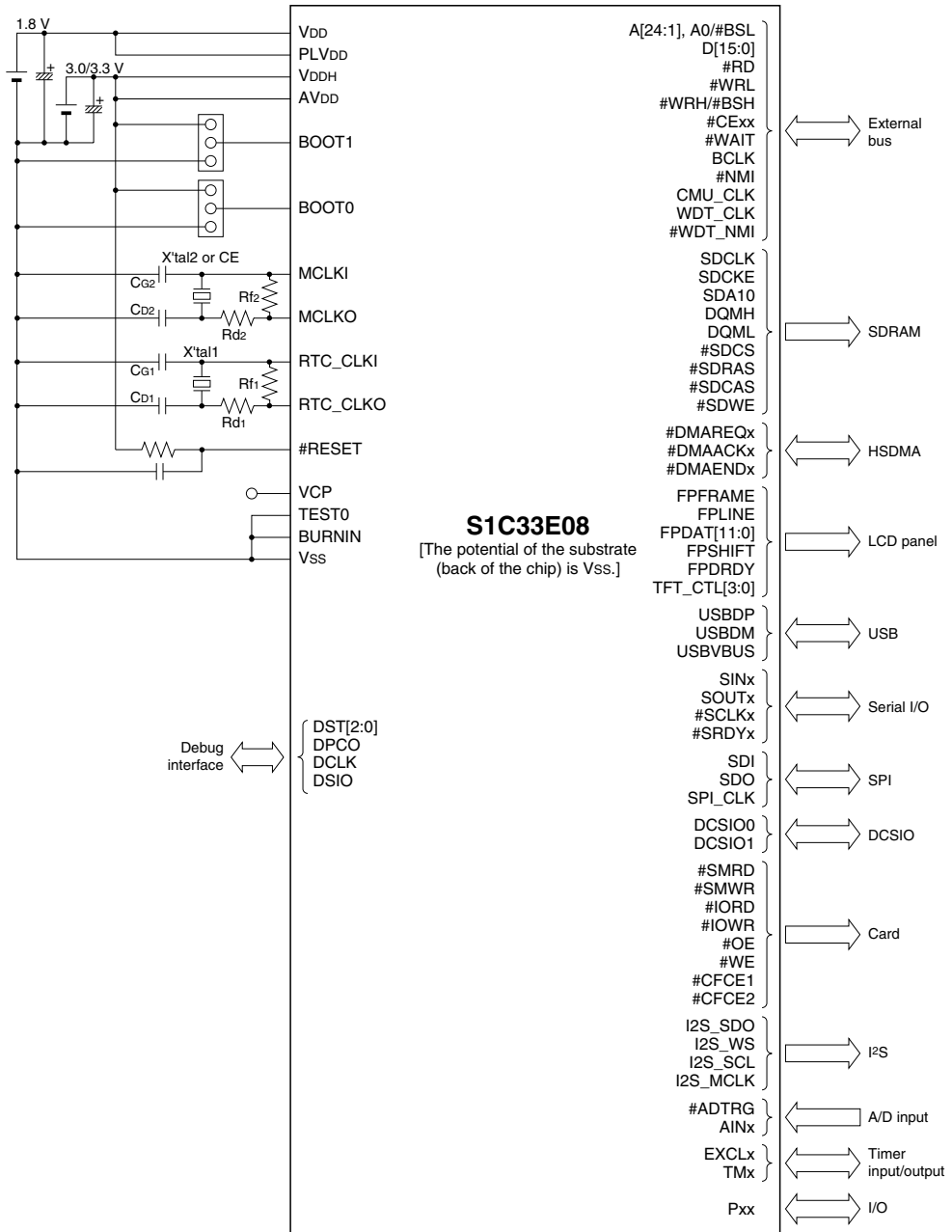
(Unless otherwise specified: $V_{DD}=3.0V$ to $3.6V$, $V_{DD}=1.65V$ to $1.95V$, $T_a=-40^{\circ}C$ to $+85^{\circ}C$)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	*
Rise time	T _{FR}		4	–	20	ns	4
Fall time	T _{FF}		4	–	20	ns	4
Differential rise and fall time matching	T _{FRFM}	T _{FR} /T _{FF}	90	–	111.11	%	
Driver output resistance	Z _{DRV}		28	–	44	Ω	
VBUS input impedance	Z _{VBUS}	R1 + R2	125	–	–	k Ω	
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.

THIS PAGE IS BLANK.

I.8 Basic External Wiring Diagram



X'tal1	Crystal resonator	32.768 kHz	
CG1	Gate capacitor	12 pF	*4
CD1	Drain capacitor	12 pF	*4
Rf1	Feedback resistor	10 MΩ	*4
Rd1	Drain resistor	0 Ω	*4

X'tal2 or CE	Resonator	Crystal	Ceramic *1
CG2	Gate capacitor	3 pF	(6 pF) *2, *3
CD2	Drain capacitor	4 pF	(6 pF) *2, *3
Rf2	Feedback resistor	1 MΩ	22 kΩ *2
Rd2	Drain resistor	0 Ω	47 Ω *2

Note: *1 CSTCW48M0X11***

*2 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.

*3 Capacitance built into the ceramic resonator

*4 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.

THIS PAGE IS BLANK.

I.9 Precautions on Mounting

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

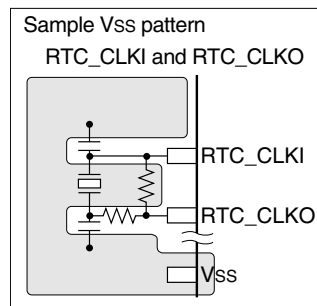
Oscillation Circuit

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

The measures against noise for the RTC_CLKO pin, and the components and lines connected to this pin is most essential, and similar measures must also be taken for the RTC_CLKI pin. The measures for the RTC_CLKI and RTC_CLKO pins are described below.

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

- (1) Components that are connected to the RTC_CLKI and RTC_CLKO 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 RTC_CLKI and RTC_CLKO 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 RTC_CLKI and RTC_CLKO pins and lines connected to those pins as well as the adjacent layers of the board using Vss. As shown in the figure on the right, 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 RTC_CLKI or MCLKI pin, the clock source should be connected to the RTC_CLKI or MCLKI pin in the shortest line. Furthermore, do not connect anything else to the RTC_CLKO or MCLKO 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 CMU_CLK pins with an oscilloscope. Check the waveform quality at the OSC3 or PLL output clock by measuring the CMU_CLK output. Ensure that the frequencies are as designed and that there is no noise or jitters. Check the waveform quality at the OSC1 clock by measuring the CMU_CLK output (after switching the system clock source to OSC1). 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.

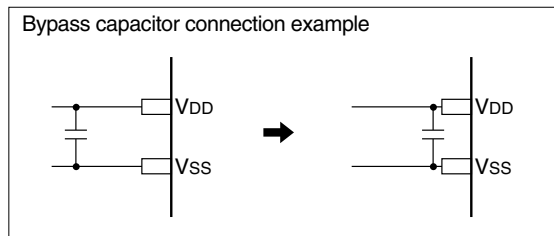
If conditions (1) to (3) are not satisfied, the OSC3 or PLL output may be jittery and the OSC1 output may be noisy. When the OSC3 or PLL output is jittery, the operating frequency will be lowered. When the OSC1 output is noisy, operation of the RTC using the OSC1 clock and the CPU core after the system clock is switched to OSC1 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 V_{DD}, V_{DDH}, V_{SS}, AV_{DD}, PLV_{DD} and PLV_{SS} pins with patterns as short and large as possible. In particular, the power supply for AV_{DD} affects A/D conversion precision.
 - (2) When connecting between the V_{DD} and V_{SS} pins with a bypass capacitor, the pins should be connected as short as possible.

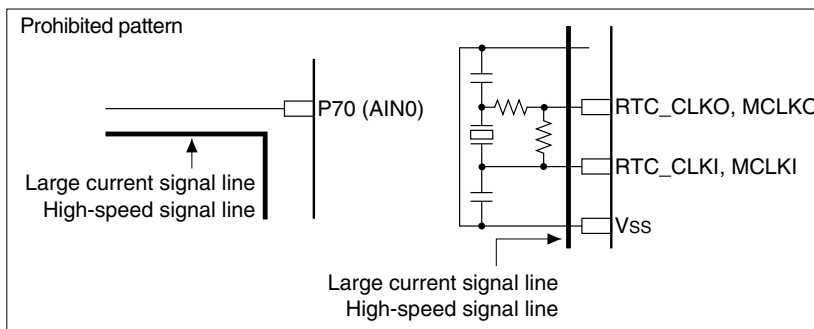


A/D Converter

- When the A/D converter is not used, the power supply pin AV_{DD} for the analog system should be connected to V_{DDH}.

Arrangement of Signal Lines

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



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.

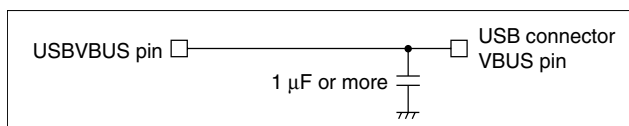
The receiver does not enter a floating state even when the USB cable is disconnected from the USB connector.

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.

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 μ F or more capacitor near the USB connector for decoupling the VBUS signal. Choose a ceramic capacitor for decoupling.



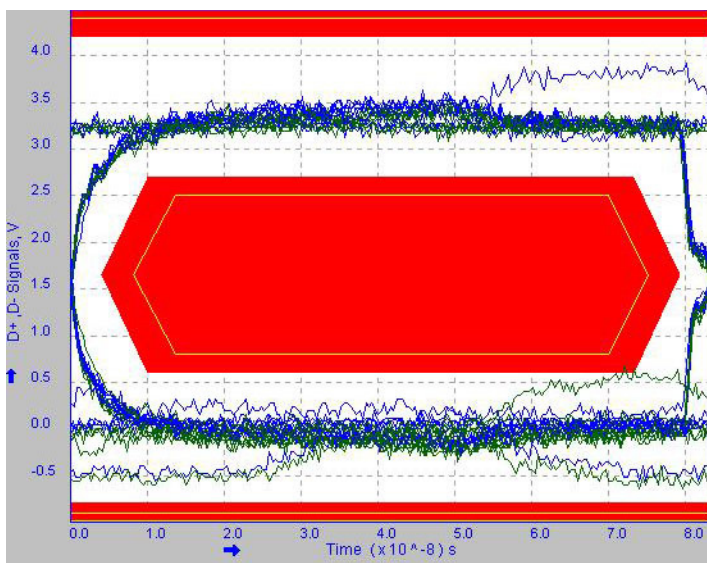
In addition to the above, verify the VBUS state completely on the actual circuit board using an oscilloscope or other device. Overshoot and other symptoms are more likely to occur when using a long USB cable and connecting it to the host side connector.

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 circuit board.



Sample EYE Diagram

Noise-Induced Erratic Operations

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

(1) TEST0 pin

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 TEST0 pin is connected to GND on the circuit board. Although the IC contains internal pull-down resistors, it is susceptible to noise because these resistors are high impedance (approximately 50 to 100 k Ω).

(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 V_{DD} or through a resistor of 10 k Ω or less.

Although the IC contains internal pull-up resistors, it is susceptible to noise because these resistors are high impedance (approximately 50 to 100 k Ω).

For details, refer to the "S1C33 Family Application Note."

(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."

(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) V_{DD}, V_{SS}, and V_{DDH} 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 k Ω 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 reasonable certainty be considered to be the source of the erratic operations.

The TEST0, DSIO, #RESET, and #NMI input circuits described above are designed to detect the edges of the input signal (#NMI can be changed to level sense mode), 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 k Ω 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 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 the oscillator input/output 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.

S1C33E08 Technical Manual

II BUS MODULES

II.1 High-Speed DMA (HSDMA)

II.1.1 Functional Outline of HSDMA

The S1C33E08 contains four channels of HSDMA (High-Speed DMA) circuits that support dual-address transfer and single-address transfer methods.

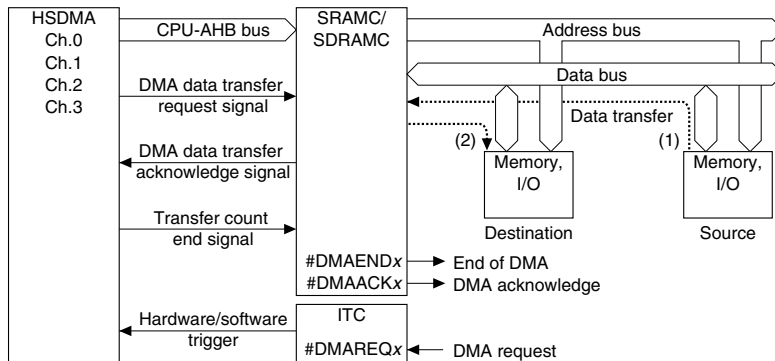
Since the control registers required for the HSDMA function are implemented with logic circuits (not located in a memory), HSDMA requests for data transfer can respond to instantaneously.

- Notes:**
- 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.
 - The MP3 decoder BIOS (MP3 decoder module) uses HSDMA Ch.0 and Ch.1, therefore, these two channels cannot be used while the MP3 decoder is active.

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.



- (1) Transfer data is read from the source memory or I/O device.
 (2) Transfer data is written to the destination memory or I/O device.

Figure II.1.1.1 Dual-Address Transfer Method

The features of dual-address transfer are outlined below.

- Source External memory and internal memory except Areas 0 and 1
- Destination External memory and internal memory except Areas 0 and 1
- Transfer data size 8, 16, or 32 bits
- Trigger
 1. Software trigger (register control)
 2. Hardware trigger (external trigger input, causes of interrupts)
- Transfer mode
 1. Single transfer (one unit of data is transferred by one trigger)
 2. Successive transfer (specified number of data are transferred by one trigger)
 3. Block transfer (data block of the specified size is transferred by one trigger)
- Transfer address control The source and/or destination addresses can be incremented or decremented in units of the transfer data size upon completion of transfer. In successive or block transfers, the address can be reset to the initial value upon completion of transfer.
- #DMAEND output Goes low at the last access of data transfer by each trigger.
- #DMAACK output Goes low when a DMA request is accepted.

II BUS MODULES: HIGH-SPEED DMA (HSDMA)

Note: A0RAM (area 0), Specific ROM (area 1), and IVRAM (area 0) cannot be specified as the source or destination for DMA transfer. While IVRAM (area 3), DST RAM (area 3), and the internal peripheral I/O registers (area 6) can be used for dual-address transfer.

Timing chart of dual-address mode

(1) SRAM

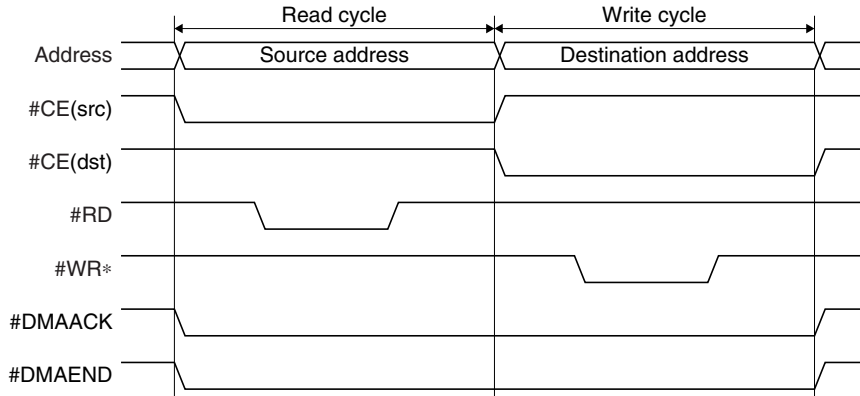


Figure II.1.1.2 #DMAACK/#DMAEND Signal Output Timing (SRAM, standard settings)

(2) SDRAM

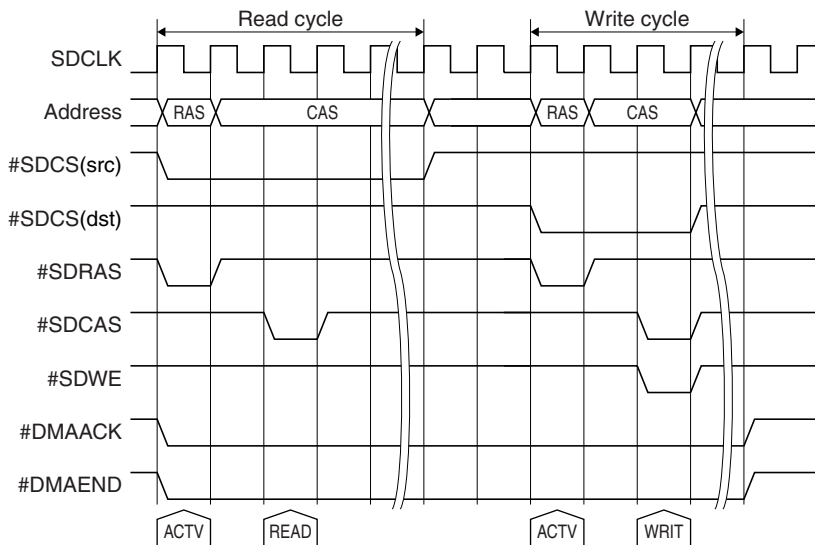


Figure II.1.1.3 #DMAACK/#DMAEND Signal Output Timing (SDRAM, standard settings)

Note: Two or more access cycles are generated when the device size of the external memory is smaller than the transfer data size. In this case, the #DMAACK/#DMAEND signal is asserted over these cycles.

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.

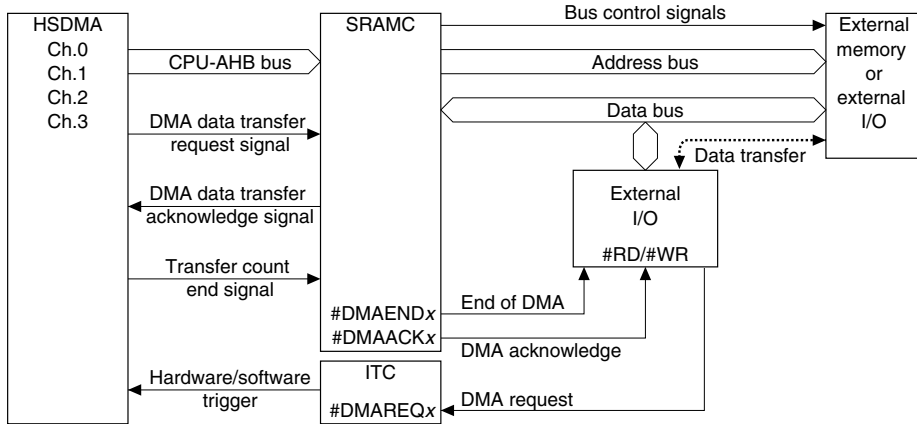


Figure II.1.1.4 Single-Address Transfer Method

The features of single-address transfer are outlined below.

- Source/destination
 1. Between an external I/O and an external memory (except SDRAM)
 2. Between an external I/O and another external I/O
- Transfer data size
 - 8, 16, or 32 bits
- Trigger
 1. Software trigger (register control)
 2. Hardware trigger (external trigger input, causes of interrupts)
- Transfer mode
 1. Single transfer (one unit of data is transferred by one trigger)
 2. Successive transfer (specified number of data are transferred by one trigger)
 3. Block transfer (data block of the specified size is transferred by one trigger)
- Transfer address control

The source and/or destination addresses can be incremented or decremented in units of the transfer data size upon completion of transfer.

In successive or block transfers, the address can be reset to the initial value upon completion of transfer.
- #DMAEND output

Goes low at the last access of data transfer by each trigger.
- #DMAACK output

Output for accessing the external I/O in every cycle during transfer.

- Notes:**
- A0RAM (area 0), Specific ROM (area 1), area 2, IVRAM (area 0 or area 3), DST RAM (area 3) and the internal peripheral I/O registers (area 6) cannot be used for single-address transfer.
 - Single-address mode does not allow data transfer between memory devices. An external logic circuit is required to perform single-address transfer between memory devices.
 - Single-address mode does not support the external memory area that is configured for SDRAM.

Timing chart of single-address mode

(1) SRAM

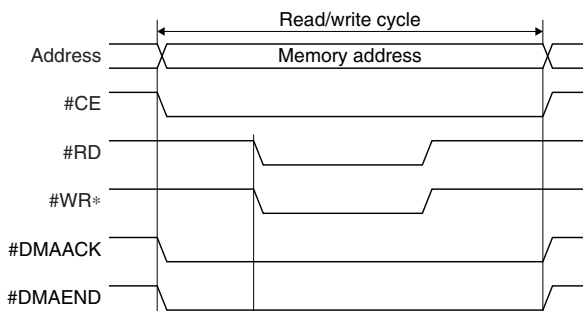


Figure II.1.1.5 #DMAACK/#DMAEND Signal Output Timing (SRAM, standard settings)

(2) SDRAM

The single-address mode does not support SDRAM.

II.1.2 I/O Pins of HSDMA

Table II.1.2.1 lists the I/O pins used for HSDMA.

Table II.1.2.1 I/O Pins of HSDMA

Pin name	I/O	Function
#DMAREQ0	I	DMA transfer request input pin for HSDMA Ch.0
#DMAREQ1	I	DMA transfer request input pin for HSDMA Ch.1
#DMAREQ2	I	DMA transfer request input pin for HSDMA Ch.2
#DMAREQ3	I	DMA transfer request input pin for HSDMA Ch.3
#DMAACK0	O	DMA acknowledge signal output pin for HSDMA Ch.0
#DMAACK1	O	DMA acknowledge signal output pin for HSDMA Ch.1
#DMAACK2	O	DMA acknowledge signal output pin for HSDMA Ch.2
#DMAACK3	O	DMA acknowledge signal output pin for HSDMA Ch.3
#DMAEND0	O	End-of-transfer signal output pin for HSDMA Ch.0
#DMAEND1	O	End-of-transfer signal output pin for HSDMA Ch.1
#DMAEND2	O	End-of-transfer signal output pin for HSDMA Ch.2
#DMAEND3	O	End-of-transfer signal output pin for HSDMA Ch.3

#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 a cause of interrupt can be selected for the HSDMA trigger source using the register in the interrupt controller.

#DMAACKx (DMA acknowledge signal output pin)

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 also output in dual-address mode.

See Figures II.1.1.2, II.1.1.3 and II.1.1.5 for the waveform of the #DMAACKx signal.

#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.

Note: The control pins above are shared with general-purpose input/output ports or other peripheral circuit input/output pins, so that functionality in the initial state is set to other than the HSDMA. Before the HSDMA signals assigned to these pins can be used, the functions of these pins must be switched for the HSDMA by setting each corresponding Port Function Select Register.

For details of pin functions and how to switch over, see Section I.3.3, "Switching Over the Multiplexed Pin Functions."

II.1.3 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.

II.1.3.1 Standard Mode and Advanced Mode

The HSDMA in the S1C33E08 is extended from that of the C33 STD models. The S1C33E08 HSDMA has two operating modes, the standard (STD) mode of which functions are compatible with the existing C33 STD models and an advanced (ADV) mode allowing use of the extended functions. Table II.1.3.1.1 shows differences between standard mode and advanced mode.

Table II.1.3.1.1 Differences between Standard Mode and Advanced Mode

Function	Standard mode	Advanced mode
Source/destination address bit width	28 bits	32 bits
Word (32-bit) data transfer	Unavailable	Available
Address decrement function with initialization	Unavailable	Available

To configure the HSDMA in advanced mode, set HSDMAADV (D0/0x30119C) to 1. The control registers (0x301162–0x30119A) for the extended functions are enabled to write after this setting. At initial reset, HSDMAADV (D0/0x30119C) is set to 0 and the HSDMA enters standard mode.

* **HSDMAADV**: Standard/Advanced Mode Select Bit in the HSDMA STD/ADV Mode Select Register (D0/0x30119C)

The following descriptions unless otherwise specified are common contents for both modes. The extended functions in advanced mode are explained assuming that HSDMAADV (D0/0x30119C) has been set to 1.

- Notes:**
- Be sure to use the control registers for advanced mode when the HSDMA is set to advanced mode.
 - The standard or advanced mode currently set is applied to all the HSDMA channels. It cannot be selected for each channel individually.

II.1.3.2 Sequential Access Time for IDMA and HSDMA

The DMAC (IDMA and HSDMA) has a higher priority for use of the AHB bus than the CPU, therefore the CPU must wait for use of the AHB bus while a DMA transfer is in progress until it has completed in default settings. Furthermore, the LCDC will be unable to read display data from the SDRAM if a DMA transfer between SDRAM addresses starts. To avoid a problem, such as degradation in graphics performance, caused by DMA, a DMA transfer that exceeds a specified number of cycles (or sequential access time) can be temporarily suspended to release the bus ownership to the CPU or LCDC. The sequential access time can be set to “Unlimited” or 64 to 940 cycles (in 64 MCLK cycle increments) using DMAACCTIME[3:0] (D[3:0]/0x30119E).

* **DMAACCTIME[3:0]**: IDMA and HSDMA Sequential Access Time Setup Bits in the DMA Sequential Access Time Register (D[3:0]/0x30119E)

Table II.1.3.2.1 Setting the Sequential Access Time

DMAACCTIME3	DMAACCTIME2	DMAACCTIME1	DMAACCTIME0	DMA sequential access time
1	1	1	1	960 cycles
1	1	1	0	896 cycles
1	1	0	1	832 cycles
1	1	0	0	768 cycles
1	0	1	1	704 cycles
1	0	1	0	640 cycles
1	0	0	1	576 cycles
1	0	0	0	512 cycles
0	1	1	1	448 cycles
0	1	1	0	384 cycles
0	1	0	1	320 cycles
0	1	0	0	256 cycles
0	0	1	1	192 cycles
0	0	1	0	128 cycles
0	0	0	1	64 cycles
0	0	0	0	Unlimited

(Default: 0b0000 = Unlimited)

When “Unlimited” is selected, the AHB bus will not be released until a DMA transfer has been completed after it starts. Specifying a number of cycles allows a DMA transfer to be temporarily suspended when the specified cycles of data transfer have been executed to release the bus. The CPU or LCDC can perform a bus access during the suspended status. After that, the DMAC resumes the data transfer that was being suspended.

II.1.3.3 Setting the Registers in Dual-Address Mode

Make sure that the HSDMA channel is disabled (HSx_EN (D0/0x30112C + 0x10•x) = 0) before setting the control information.

* **HSx_EN**: Ch.x Enable Bit in the HSDMA Ch.x Enable Register (D0/0x30112C + 0x10•x)

Address mode

The address mode select bit $DUALMx$ (D15/0x301122 + 0x10•x) should be set to 1 (dual-address mode). This bit is set to 0 (single-address mode) at initial reset.

* **DUALMx**: Ch.x Address Mode Select Bit in the HSDMA Ch.x Control Register (D15/0x301122 + 0x10•x)

Transfer mode

A transfer mode should be set using $DxMOD[1:0]$ (D[15:14]/0x30112A + 0x10•x).

* **DxMOD[1:0]**: Ch.x Transfer Mode Select Bits in the HSDMA Ch.x High-Order Destination Address Setup Register (D[15:14]/0x30112A + 0x10•x)

The following three transfer modes are available:

Single transfer mode ($DxMOD[1:0]$ (D[15:14]/0x30112A + 0x10•x) = 00, default)

In this mode, a transfer operation invoked by one trigger is completed after transferring one unit of data of the specified size. 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 ($DxMOD[1:0]$ (D[15:14]/0x30112A + 0x10•x) = 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 ($DxMOD[1:0]$ (D[15:14]/0x30112A + 0x10•x) = 10)

In this mode, a transfer operation invoked by one trigger is completed after transferring one block of data of the size set by $BLKLENx[7:0]$ (D[7:0]/0x301120 + 0x10•x). 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

Standard mode (HSDMAADV (D0/0x30119C) = 0, default)

$DATSIZEx$ (D14/0x301126 + 0x10•x) 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).

* **DATSIZEx**: Ch.x Transfer Data Size Select Bit in the HSDMA Ch.x High-Order Source Address Setup Register (D14/0x301126 + 0x10•x)

Advanced mode (HSDMAADV (D0/0x30119C) = 1)

In advanced mode, $WORDSIZEx$ (D0/0x301162 + 0x10•x) is provided to select word size (32 bits) in addition to half-word size and byte size that can be selected using $DATSIZEx$ (D14/0x301126 + 0x10•x).

* **WORDSIZEx**: Ch.x Transfer Data Size Select Bit in the HSDMA Ch.x Control Register for ADV mode (D0/0x301162 + 0x10•x)

Table II.1.3.3.1 Transfer Data Size Selectable in Advanced Mode

WORDSIZEx	DATSIZEx	Transfer data size
1	X	Word (32 bits)
0	1	Half-word (16 bits)
0	0	Byte (8 bits)

Block length

When using block transfer mode ($DxMOD[1:0]$ ($D[15:14]/0x30112A + 0x10 \bullet x$) = 10), the data block length (in units of the selected transfer data size) should be set using $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \bullet x$).

- * **BLKLENx[7:0]**: Ch.x Block Length Bits in the HSDMA Ch.x Transfer Counter Register ($D[7:0]/0x301120 + 0x10 \bullet x$)

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, $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \bullet x$) is used as bits 7–0 of the transfer counter.

Transfer counter

Block transfer mode

In block transfer mode, up to 16 bits of transfer count can be specified using $TCx_L[7:0]$ ($D[15:8]/0x301120 + 0x10 \bullet x$) and $TCx_H[7:0]$ ($D[7:0]/0x301122 + 0x10 \bullet x$).

- * **TCx_L[7:0]**: Ch.x Transfer Counter [7:0] Bits in the HSDMA Ch.x Transfer Counter Register ($D[15:8]/0x301120 + 0x10 \bullet x$)
- * **TCx_H[7:0]**: Ch.x Transfer Counter [15:8] Bits in the HSDMA Ch.x Control Register ($D[7:0]/0x301122 + 0x10 \bullet x$)

Single transfer and successive transfer modes

In single transfer and successive transfer modes, up to 24 bits of transfer count can be specified using $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \bullet x$), $TCx_L[7:0]$ ($D[15:8]/0x301120 + 0x10 \bullet x$) and $TCx_H[7:0]$ ($D[7:0]/0x301122 + 0x10 \bullet x$).

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

Standard mode (HSDMAADV (D0/0x30119C) = 0, default)

In standard mode, a 28-bit source address and a 28-bit destination address for DMA transfer can be specified using $SxADR_L[15:0]$ ($D[15:0]/0x301124 + 0x10 \bullet x$), $SxADR_H[11:0]$ ($D[11:0]/0x301126 + 0x10 \bullet x$), $DxADR_L[15:0]$ ($D[15:0]/0x301128 + 0x10 \bullet x$) and $DxADR_H[11:0]$ ($D[11:0]/0x30112A + 0x10 \bullet x$).

- * **SxADR_L[15:0]**: Ch.x Source Address[15:0] in the HSDMA Ch.x Low-Order Source Address Setup Register ($D[15:0]/0x301124 + 0x10 \bullet x$)
- * **SxADR_H[11:0]**: Ch.x Source Address[27:16] in the HSDMA Ch.x High-Order Source Address Setup Register ($D[11:0]/0x301126 + 0x10 \bullet x$)
- * **DxADR_L[15:0]**: Ch.x Destination Address[15:0] in the HSDMA Ch.x Low-Order Destination Address Setup Register ($D[15:0]/0x301128 + 0x10 \bullet x$)
- * **DxADR_H[11:0]**: Ch.x Destination Address[27:16] in the HSDMA Ch.x High-Order Destination Address Setup Register ($D[11:0]/0x30112A + 0x10 \bullet x$)

Advanced mode (HSDMAADV (D0/0x30119C) = 1)

In advanced mode, a 32-bit source address and a 32-bit destination address for DMA transfer can be specified using $SxADR_L[15:0]$ ($D[15:0]/0x301164 + 0x10 \bullet x$), $SxADR_H[15:0]$ ($D[15:0]/0x301166 + 0x10 \bullet x$), $DxADR_L[15:0]$ ($D[15:0]/0x301168 + 0x10 \bullet x$) and $DxADR_H[15:0]$ ($D[15:0]/0x30116A + 0x10 \bullet x$).

- * **SxADR_L[15:0]**: Ch.x Source Address[15:0] in the HSDMA Ch.x Low-Order Source Address Setup Register for ADV mode ($D[15:0]/0x301164 + 0x10 \bullet x$)
- * **SxADR_H[15:0]**: Ch.x Source Address[31:16] in the HSDMA Ch.x High-Order Source Address Setup Register for ADV mode ($D[15:0]/0x301166 + 0x10 \bullet x$)
- * **DxADR_L[15:0]**: Ch.x Destination Address[15:0] in the HSDMA Ch.x Low-Order Destination Address Setup Register for ADV mode ($D[15:0]/0x301168 + 0x10 \bullet x$)
- * **DxADR_H[15:0]**: Ch.x Destination Address[31:16] in the HSDMA Ch.x High-Order Destination Address Setup Register for ADV mode ($D[15:0]/0x30116A + 0x10 \bullet x$)

Note: In advanced mode, be sure to use the control registers for advanced mode to set source/destination addresses.

Address increment/decrement control

Standard mode (HSDMAADV (D0/0x30119C) = 0, default)

The source and/or destination addresses can be incremented or decremented when one data transfer is completed. $SxIN[1:0]$ (D[13:12]/0x301126 + 0x10•x) (for source address) and $DxIN[1:0]$ (D[13:12]/0x30112A + 0x10•x) (for destination address) are used to set this function.

- * **$SxIN[1:0]$** : Ch.x Source Address Control Bits in the HSDMA Ch.x High-Order Source Address Setup Register (D[13:12]/0x301126 + 0x10•x)
- * **$DxIN[1:0]$** : Ch.x Destination Address Control Bits in the HSDMA Ch.x High-Order Destination Address Setup Register (D[13:12]/0x30112A + 0x10•x)

$SxIN[1:0]/DxIN[1:0] = 00$: 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.

$SxIN[1:0]/DxIN[1:0] = 01$: address decremented without initialization

The address is decremented by an amount equal to the specified data size when one data transfer is completed. The address that has been decremented during transfer does not return to the initial value.

$SxIN[1:0]/DxIN[1:0] = 10$: address incremented with initialization

The address is incremented by an amount equal to the specified data size when one data transfer is completed. In single transfer mode, the address that has been incremented during transfer does not return to the initial value. In successive transfer modes, the incremented address returns to the initial value when the specified number of transfers is completed. In block transfer mode, the incremented address returns to the initial value when the block transfer is completed.

$SxIN[1:0]/DxIN[1:0] = 11$: address incremented without initialization

The address is incremented by an amount equal to the specified data size when one data transfer is completed. The address that has been incremented during transfer does not return to the initial value.

Advanced mode (HSDMAADV (D0/0x30119C) = 1)

The address control conditions set using $SxIN[1:0]$ (D[13:12]/0x301126 + 0x10•x) and $DxIN[1:0]$ (D[13:12]/0x30112A + 0x10•x) are effective in advanced mode. Furthermore, advanced mode allows selection of “Address decremented with initialization.” This condition can be selected using the $SxID$ (D4/0x301162 + 0x10•x) and $DxID$ (D5/0x301162 + 0x10•x).

- * **$SxID$** : Ch.x Source Address Control Bit in the HSDMA Ch.x Control Register for ADV mode (D4/0x301162 + 0x10•x)
- * **$DxID$** : Ch.x Destination Address Control Bit in the HSDMA Ch.x Control Register for ADV mode (D5/0x301162 + 0x10•x)

When $SxID$ (D4/0x301162 + 0x10•x) and/or $DxID$ (D5/0x301162 + 0x10•x) are set to 0 (default), the conditions selected using $SxIN[1:0]$ (D[13:12]/0x301126 + 0x10•x) and/or $DxIN[1:0]$ (D[13:12]/0x30112A + 0x10•x) are effective. When $SxID$ (D4/0x301162 + 0x10•x) and/or $DxID$ (D5/0x301162 + 0x10•x) are set to 1, “Address decremented with initialization” is selected.

$SxID/DxID = 1$: address decremented with initialization

The address is decremented by an amount equal to the specified data size when one data transfer is completed. In single transfer mode, the address that has been decremented during transfer does not return to the initial value. In successive transfer modes, the decremented address returns to the initial value when the specified number of transfers is completed. In block transfer mode, the decremented address returns to the initial value when the block transfer is completed.

II.1.3.4 Setting the Registers in Single-Address Mode

Make sure that the HSDMA channel is disabled (HS_x_EN (D0/0x30112C + 0x10•x) = 0) before setting the control information.

* **HS_x_EN**: Ch.x Enable Bit in the HSDMA Ch.x Enable Register (D0/0x30112C + 0x10•x)

Address mode

The address mode select bit $DUALM_x$ (D15/0x301122 + 0x10•x) should be set to 0 (single-address mode). This bit is set to 0 at initial reset.

* **DUALM_x**: Ch.x Address Mode Select Bit in the HSDMA Ch.x Control Register (D15/0x301122 + 0x10•x)

Transfer mode

A transfer mode should be set using $DxMOD[1:0]$ (D[15:14]/0x30112A + 0x10•x).

* **DxMOD[1:0]**: Ch.x Transfer Mode Select Bits in the HSDMA Ch.x High-Order Destination Address Setup Register (D[15:14]/0x30112A + 0x10•x)

Table II.1.3.4.1 Transfer Mode

DxMOD1	DxMOD0	Mode
1	1	Invalid
1	0	Block transfer mode
0	1	Successive transfer mode
0	0	Single transfer mode

Refer to the explanation in Section II.1.3.3, “Setting the Registers in Dual-Address Mode.”

Direction of transfer

The direction of data transfer should be set using $DxDIR$ (D14/0x301122 + 0x10•x).

* **DxDIR**: Ch.x Transfer Direction Control Bit in the HSDMA Ch.x Control Register (D14/0x301122 + 0x10•x)

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

Standard mode (HSDMAADV (D0/0x30119C) = 0, default)

$DATSIZE_x$ (D14/0x301126 + 0x10•x) 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).

* **DATSIZE_x**: Ch.x Transfer Data Size Select Bit in the HSDMA Ch.x High-Order Source Address Setup Register (D14/0x301126 + 0x10•x)

Advanced mode (HSDMAADV (D0/0x30119C) = 1)

In advanced mode, $WORDSIZE_x$ (D0/0x301162 + 0x10•x) is provided to select a word size (32 bits) in addition to a half-word size and byte size that can be selected using $DATSIZE_x$ (D14/0x301126 + 0x10•x).

* **WORDSIZE_x**: Ch.x Transfer Data Size Select Bit in the HSDMA Ch.x Control Register for ADV mode (D0/0x301162 + 0x10•x)

Table II.1.3.4.2 Transfer Data Size Selectable in Advanced Mode

WORDSIZE _x	DATSIZE _x	Transfer data size
1	X	Word (32 bits)
0	1	Half-word (16 bits)
0	0	Byte (8 bits)

$DATSIZE_x$ (D14/0x301126 + 0x10•x) and $WORDSIZE_x$ (D0/0x301162 + 0x10•x) are used to set the unit size of data to be transferred.

Block length

When using block transfer mode ($DxMOD[1:0]$ ($D[15:14]/0x30112A + 0x10\bullet x$) = 10), the data block length (in units of the selected transfer data size) should be set using $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10\bullet x$).

- * **BLKLENx[7:0]**: Ch.x Block Length Bits in the HSDMA Ch.x Transfer Counter Register ($D[7:0]/0x301120 + 0x10\bullet x$)

In single transfer and successive transfer modes, $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10\bullet x$) are used as bits 7–0 of the transfer counter.

Note: When performing data transfer in block transfer mode, the block size must not be set to 0.

Transfer counter

Block transfer mode

In block transfer mode, up to 16 bits of transfer count can be specified using $TCx_L[7:0]$ ($D[15:8]/0x301120 + 0x10\bullet x$) and $TCx_H[7:0]$ ($D[7:0]/0x301122 + 0x10\bullet x$).

- * **TCx_L[7:0]**: Ch.x Transfer Counter [7:0] Bits in the HSDMA Ch.x Transfer Counter Register ($D[15:8]/0x301120 + 0x10\bullet x$)
- * **TCx_H[7:0]**: Ch.x Transfer Counter [15:8] Bits in the HSDMA Ch.x Control Register ($D[7:0]/0x301122 + 0x10\bullet x$)

Single transfer and successive transfer modes

In single transfer and successive transfer modes, up to 24 bits of transfer count can be specified using $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10\bullet x$), $TCx_L[7:0]$ ($D[15:8]/0x301120 + 0x10\bullet x$) and $TCx_H[7:0]$ ($D[7:0]/0x301122 + 0x10\bullet x$).

Memory address

Standard mode (HSDMAADV (D0/0x30119C) = 0, default)

In standard mode, $SxADR_L[15:0]$ ($D[15:0]/0x301124 + 0x10\bullet x$) and $SxADR_H[11:0]$ ($D[11:0]/0x301126 + 0x10\bullet x$) are used to specify a 28-bit memory address.

- * **SxADR_L[15:0]**: Ch.x Source Address[15:0] in the HSDMA Ch.x Low-Order Source Address Setup Register ($D[15:0]/0x301124 + 0x10\bullet x$)
- * **SxADR_H[11:0]**: Ch.x Source Address[27:16] in the HSDMA Ch.x High-Order Source Address Setup Register ($D[11:0]/0x301126 + 0x10\bullet x$)

Advanced mode (HSDMAADV (D0/0x30119C) = 1)

In advanced mode, $SxADR_L[15:0]$ ($D[15:0]/0x301164 + 0x10\bullet x$) and $SxADR_H[15:0]$ ($D[15:0]/0x301166 + 0x10\bullet x$) are used to specify a 32-bit memory address.

- * **SxADR_L[15:0]**: Ch.x Source Address[15:0] in the HSDMA Ch.x Low-Order Source Address Setup Register for ADV mode ($D[15:0]/0x301164 + 0x10\bullet x$)
- * **SxADR_H[15:0]**: Ch.x Source Address[31:16] in the HSDMA Ch.x High-Order Source Address Setup Register for ADV mode ($D[15:0]/0x301166 + 0x10\bullet x$)

Note: In advanced mode, be sure to use the control registers for advanced mode to set a memory address.

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. $DxADR_L[15:0]$ ($D[15:0]/0x301168 + 0x10\bullet x$) and $DxADR_H[15:0]$ ($D[11:0]/0x30116A + 0x10\bullet x$) are not used in single-address mode.

Address increment/decrement control

Standard mode (HSDMAADV (D0/0x30119C) = 0, default)

The memory addresses can be incremented or decremented when one data transfer is completed. $SxIN[1:0]$ ($D[13:12]/0x301126 + 0x10\bullet x$) is used to set this function.

- * **SxIN[1:0]**: Ch.x Source Address Control Bits in the HSDMA Ch.x High-Order Source Address Setup Register ($D[13:12]/0x301126 + 0x10\bullet x$)

Table II.1.3.4.3 Address Control

SxIN1	SxIN0	Address control
1	1	Increment without initialization
1	0	Increment with initialization
0	1	Decrement without initialization
0	0	Fixed

Advanced mode (HSDMAADV (D0/0x30119C) = 1)

The address control condition set using SxIN[1:0] (D[13:12]/0x301126 + 0x10•x) is effective in advanced mode. Furthermore, advanced mode allows selection of “Address decremented with initialization.” This condition can be selected using the SxID (D4/0x301162 + 0x10•x).

* **SxID**: Ch.x Source Address Control Bit in the HSDMA Ch.x Control Register for ADV mode (D4/0x301162 + 0x10•x)

When SxID (D4/0x301162 + 0x10•x) is set to 0 (default), the condition selected using SxIN[1:0] (D[13:12]/0x301126 + 0x10•x) is effective. When SxID (D4/0x301162 + 0x10•x) is set to 1, “Address decremented with initialization” is selected.

Refer to the explanation in Section II.1.3.3, “Setting the Registers in Dual-Address Mode.”

DxIN[1:0] (D[13:12]/0x30112A + 0x10•x) and DxID (D5/0x301162 + 0x10•x) are not used in single-address mode.

II.1.4 Enabling/Disabling DMA Transfer

The HSDMA transfer is enabled by writing 1 to HS x _EN (D0/0x30112C + 0x10•x).

* **HS x _EN**: Ch.x Enable Bit in the HSDMA Ch.x Enable Register (D0/0x30112C + 0x10•x)

However, the control information must always be set correctly before enabling a DMA transfer.

Note that the control information cannot be set when HS x _EN (D0/0x30112C + 0x10•x) = 1.

When HS x _EN (D0/0x30112C + 0x10•x) is set to 0, HSDMA requests are no longer accepted.

When a DMA transfer is completed (transfer counter = 0), HS x _EN (D0/0x30112C + 0x10•x) is reset to 0 to disable the following trigger inputs.

II.1.5 Trigger Source

A HSDMA trigger source for each channel can be selected from among 15 types using HSDxS[3:0] (D[7:0]/0x300298, D[7:0]/0x300299). This function is supported by the interrupt controller.

- * **HSD0S[3:0]**: Ch.0 Trigger Set-Up Bits in the HSDMA Ch.0–1 Trigger Set-Up Register (D[3:0]/0x300298)
- * **HSD1S[3:0]**: Ch.1 Trigger Set-Up Bits in the HSDMA Ch.0–1 Trigger Set-Up Register (D[7:4]/0x300298)
- * **HSD2S[3:0]**: Ch.2 Trigger Set-Up Bits in the HSDMA Ch.2–3 Trigger Set-Up Register (D[3:0]/0x300299)
- * **HSD3S[3:0]**: Ch.3 Trigger Set-Up Bits in the HSDMA Ch.2–3 Trigger Set-Up Register (D[7:4]/0x300299)

Table II.1.5.1 shows the setting value and the corresponding trigger source.

Table II.1.5.1 HSDMA Trigger Source

Value	Ch.0 trigger source	Ch.1 trigger source	Ch.2 trigger source	Ch.3 trigger source
0000	Software trigger	Software trigger	Software trigger	Software trigger
0001	#DMAREQ0 input (falling edge)	#DMAREQ1 input (falling edge)	#DMAREQ2 input (falling edge)	#DMAREQ3 input (falling edge)
0010	#DMAREQ0 input (rising edge)	#DMAREQ1 input (rising edge)	#DMAREQ2 input (rising edge)	#DMAREQ3 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	(reserved)	(reserved)	(reserved)	(reserved)
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	(reserved)	(reserved)
1001	I ² S left	I ² S right	SPI transmit DMA request	SPI receive DMA request
1010	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full	Serial I/F Ch.2 Rx buffer full	(reserved)
1011	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty	Serial I/F Ch.2 Tx buffer empty	(reserved)
1100	A/D conversion completion	A/D conversion completion	A/D conversion completion	A/D conversion completion
1101	Port 8 input (SPI interrupt)	Port 9 input (USB PDREQ)	Port 10 input (USB interrupt)	Port 11 input (DCSIO interrupt)
1110	Port 12 input	Port 13 input	Port 14 input	Port 15 input

By selecting a cause of interrupt with the HSDMA trigger set-up register, the HSDMA channel is invoked when the selected cause of interrupt occurs. The interrupt control bits (cause-of-interrupt flag, interrupt enable register, IDMA request register, interrupt priority register) do not affect this invocation. The cause of interrupt that invokes HSDMA sets the cause-of-interrupt 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 cause of interrupt 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 HSTx (Dx/0x30029A).

- * **HSTx**: Ch.x Software Trigger Bit in the HSDMA Software Trigger Register (Dx/0x30029A)

When the selected trigger 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 HSx_TF (D0/0x30112E + 0x10•x).

- * **HSx_TF**: Ch.x Trigger Flag Status/Clear Bit in the HSDMA Ch.x Trigger Flag Register (D0/0x30112E + 0x10•x)

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.

Note: The following shows the priority order of channels when DMA triggers with the same interrupt level occur in two or more HSDMA and IDMA channels.

Priority	High ←	→ Low
Channel	HSDMA Ch.0 > Ch.1 > Ch.2 > Ch.3 > IDMA software trigger > IDMA hardware trigger	

II.1.6 Operation of HSDMA

An HSDMA channel starts data transfer by the selected trigger source.

Make sure that transfer conditions and a trigger source are set and the HSDMA channel is enabled before starting a DMA transfer.

II.1.6.1 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 SRAMC and SDRAMC.

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 (dual-address mode)

The channel for which $DxMOD[1:0]$ ($D[15:14]/0x30112A + 0x10 \bullet x$) 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 $DATSIZEx$ ($D14/0x301126 + 0x10 \bullet x$) or $WORDSIZEx$ ($D0/0x301162 + 0x10 \bullet x$). 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.

- * **$DxMOD[1:0]$** : Ch.x Transfer Mode Select Bits in the HSDMA Ch.x High-Order Destination Address Setup Register ($D[15:14]/0x30112A + 0x10 \bullet x$)
- * **$DATSIZEx$** : Ch.x Transfer Data Size Select Bit in the HSDMA Ch.x High-Order Source Address Setup Register ($D14/0x301126 + 0x10 \bullet x$)
- * **$WORDSIZEx$** : Ch.x Transfer Data Size Select Bit in the HSDMA Ch.x Control Register for ADV mode ($D0/0x301162 + 0x10 \bullet x$)

The operation of HSDMA in single transfer mode is shown by the flow chart in Figure II.1.6.1.1.

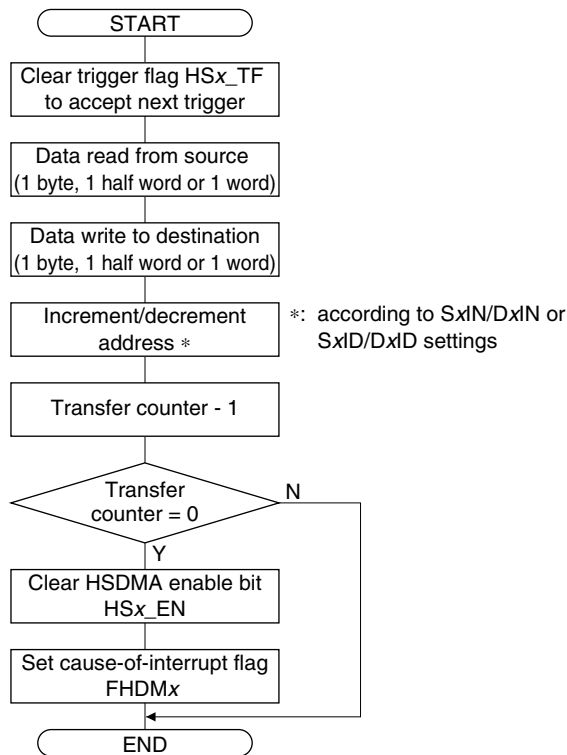


Figure II.1.6.1.1 Operation Flow in Single Transfer Mode

(1) When a trigger is accepted, the trigger flag HSx_TF ($D0/0x30112E + 0x10\bullet x$) is cleared and then data of the size set in the control information is read from the source address.

* **HSx_TF**: Ch.x Trigger Flag Status/Clear Bit in the HSDMA Ch.x Trigger Flag Register ($D0/0x30112E + 0x10\bullet x$)

(2) The read data is written to the destination address.

(3) The addresses are incremented or decremented according to the $SxIN[1:0]$ ($D[13:12]/0x301126 + 0x10\bullet x$)/ $DxIN[1:0]$ ($D[13:12]/0x30112A + 0x10\bullet x$) or $SxID$ ($D4/0x301162 + 0x10\bullet x$)/ $DxID$ ($D5/0x301162 + 0x10\bullet x$) settings. *1

* **SxIN[1:0]**: Ch.x Source Address Control Bits in the HSDMA Ch.x High-Order Source Address Setup Register ($D[13:12]/0x301126 + 0x10\bullet x$)

* **DxIN[1:0]**: Ch.x Destination Address Control Bits in the HSDMA Ch.x High-Order Destination Address Setup Register ($D[13:12]/0x30112A + 0x10\bullet x$)

* **SxID**: Ch.x Source Address Control Bit in the HSDMA Ch.x Control Register for ADV mode ($D4/0x301162 + 0x10\bullet x$)

* **DxID**: Ch.x Destination Address Control Bit in the HSDMA Ch.x Control Register for ADV mode ($D5/0x301162 + 0x10\bullet x$)

(4) The transfer counter is decremented.

(5) The HSDMA enable bit HSx_EN ($D0/0x30112C + 0x10\bullet x$) is cleared and HSDMA cause-of-interrupt flag in ITC is set when the transfer counter reaches 0.

* **HSx_EN**: Ch.x Enable Bit in the HSDMA Ch.x Enable Register ($D0/0x30112C + 0x10\bullet x$)

*1: In standard mode, $SxID$ ($D4/0x301162 + 0x10\bullet x$) and $DxID$ ($D5/0x301162 + 0x10\bullet x$) are both fixed at 0.

Successive transfer mode (dual-address mode)

The channel for which $DxMOD[1:0]$ ($D[15:14]/0x30112A + 0x10\bullet x$) 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 II.1.6.1.2.

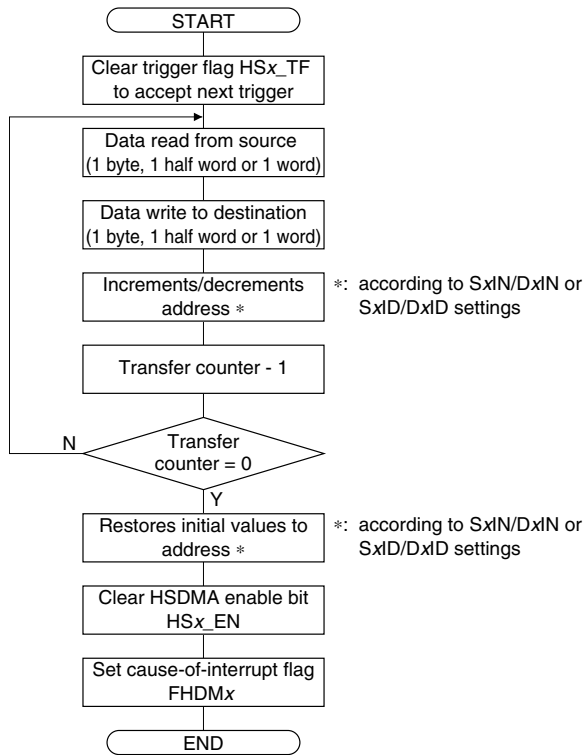


Figure II.1.6.1.2 Operation Flow in Successive Transfer Mode

- (1) When a trigger is accepted, the trigger flag HSx_TF ($D0/0x30112E + 0x10\bullet x$) 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 $SxIN[1:0]$ ($D[13:12]/0x301126 + 0x10\bullet x$)/ $DxIN[1:0]$ ($D[13:12]/0x30112A + 0x10\bullet x$) or $SxID$ ($D4/0x301162 + 0x10\bullet x$)/ $DxID$ ($D5/0x301162 + 0x10\bullet x$) settings. *1
- (4) The transfer counter is decremented.
- (5) Steps (1) to (4) are repeated until the transfer counter reaches 0.
- (6) The address returns to the initial value if $SxIN[1:0]$ ($D[13:12]/0x301126 + 0x10\bullet x$)/ $DxIN[1:0]$ ($D[13:12]/0x30112A + 0x10\bullet x$) is 10 or $SxID$ ($D4/0x301162 + 0x10\bullet x$)/ $DxID$ ($D5/0x301162 + 0x10\bullet x$) is 1. *1
- (7) The HSDMA enable bit HSx_EN ($D0/0x30112C + 0x10\bullet x$) is cleared and HSDMA cause-of-interrupt flag in ITC is set when the transfer counter reaches 0.

*1: In standard mode, $SxID$ ($D4/0x301162 + 0x10\bullet x$) and $DxID$ ($D5/0x301162 + 0x10\bullet x$) are both fixed at 0.

Block transfer mode (dual-address mode)

The channel for which $DxMOD[1:0]$ ($D[15:14]/0x30112A + 0x10 \cdot x$) 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 $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \cdot x$). 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 II.1.6.1.3.

* **BLKLENx[7:0]**: Ch.x Block Length Bits in the HSDMA Ch.x Transfer Counter Register ($D[7:0]/0x301120 + 0x10 \cdot x$)

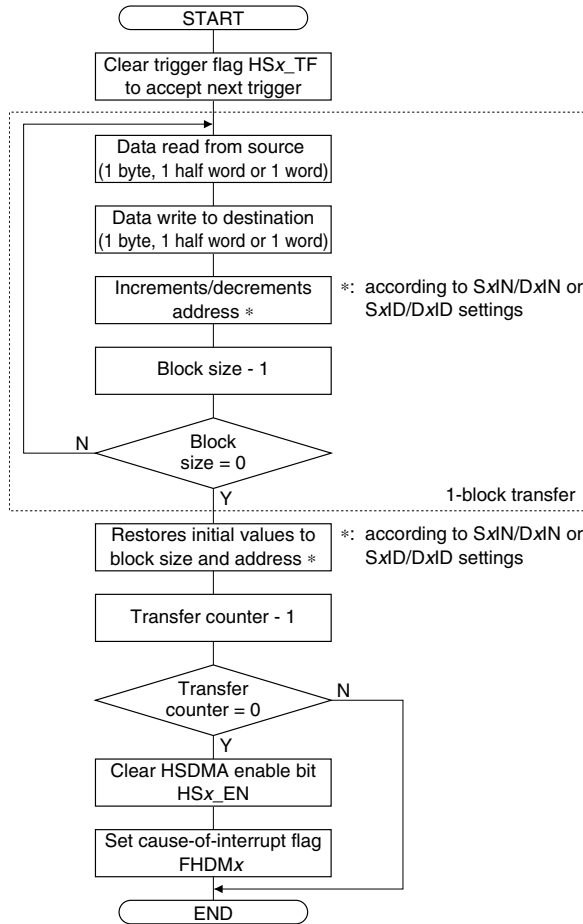


Figure II.1.6.1.3 Operation Flow in Block Transfer Mode

- (1) When a trigger is accepted, the trigger flag HSx_TF ($D0/0x30112E + 0x10 \cdot x$) 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 $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \cdot x$) is decremented.
- (4) Steps (1) to (3) are repeated until $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \cdot x$) reaches 0.
- (5) The address returns to the initial value if $SxIN[1:0]$ ($D[13:12]/0x301126 + 0x10 \cdot x$)/ $DxIN[1:0]$ ($D[13:12]/0x30112A + 0x10 \cdot x$) is 10 or $SxID$ ($D4/0x301162 + 0x10 \cdot x$)/ $DxID$ ($D5/0x301162 + 0x10 \cdot x$) is 1. *1
- (6) The transfer counter is decremented.
- (7) Steps (1) to (6) are repeated until the transfer counter reaches 0.
- (8) The HSDMA enable bit HSx_EN ($D0/0x30112C + 0x10 \cdot x$) is cleared and HSDMA cause-of-interrupt flag in ITC is set when the transfer counter reaches 0.

*1: In standard mode, $SxID$ ($D4/0x301162 + 0x10 \cdot x$) and $DxID$ ($D5/0x301162 + 0x10 \cdot x$) are both fixed at 0.

II.1.6.2 Operation in Single-Address Mode

In single-address mode, data read/write operations are performed simultaneously. The data transfer direction (read from I/O device → write to memory or read from memory → write to I/O device) is set using $DxDIR$ ($D14/0x301122 + 0x10 \bullet x$).

* $DxDIR$: Ch.x Transfer Direction Control Bit in the HSDMA Ch.x Control Register ($D14/0x301122 + 0x10 \bullet x$)

Single-address mode has three transfer size modes, in each of which data transfer operates differently. The following describes the operation of HSDMA in single-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 ($DxDIR$ ($D14/0x301122 + 0x10 \bullet x$) = 1)**

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 ($DxDIR$ ($D14/0x301122 + 0x10 \bullet x$) = 0, default)**

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.

The number of bus operations for a DMA transfer is decided according to the transfer data size and I/O device size as shown in the table below.

Table II.1.6.2.1 Number of Bus Operations Per DMA Transfer

Transfer data size	I/O device size	Number of bus operations
32 bits	8 bits	4
32 bits	16 bits	2
16 bits	8 bits	2
Other		1

- Notes:**
- A0RAM (area 0), Specific ROM (area 1), area 2, IVRAM (area 0 or area 3), DST RAM (area 3) and the internal peripheral I/O registers (area 6) cannot be used for single-address transfer.
 - Single-address mode does not allow data transfer between memory devices. An external logic circuit is required to perform single-address transfer between memory devices.
 - Single-address mode does not support the external memory area that is configured for SDRAM.

#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 cause of interrupt (completion of HSDMA) is generated.

Single transfer mode (single-address mode)

The channel for which $DxMOD[1:0]$ ($D[15:14]/0x30112A + 0x10 \bullet x$) 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 $DATSIZEx$ ($D14/0x301126 + 0x10 \bullet x$) or $WORDSIZEx$ ($D0/0x301162 + 0x10 \bullet x$). 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.

- * **$DxMOD[1:0]$** : Ch.x Transfer Mode Select Bits in the HSDMA Ch.x High-Order Destination Address Setup Register ($D[15:14]/0x30112A + 0x10 \bullet x$)
- * **$DATSIZEx$** : Ch.x Transfer Data Size Select Bit in the HSDMA Ch.x High-Order Source Address Setup Register ($D14/0x301126 + 0x10 \bullet x$)
- * **$WORDSIZEx$** : Ch.x Transfer Data Size Select Bit in the HSDMA Ch.x Control Register for ADV mode ($D0/0x301162 + 0x10 \bullet x$)

The operation of HSDMA in single transfer mode is shown by the flow chart in Figure II.1.6.2.1.

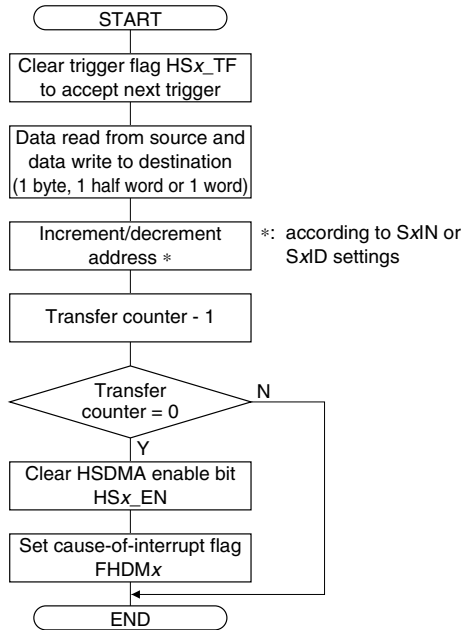


Figure II.1.6.2.1 Operation Flow in Single Transfer Mode

(1) When a trigger is accepted, the trigger flag HSx_TF ($D0/0x30112E + 0x10 \bullet x$) is cleared. Data of the size set in the control information is read from the external memory or I/O device according to the specified direction and is written to the I/O device or external memory. *1

* **HSx_TF** : Ch.x Trigger Flag Status/Clear Bit in the HSDMA Ch.x Trigger Flag Register ($D0/0x30112E + 0x10 \bullet x$)

(2) The addresses are incremented or decremented according to the $SxIN[1:0]$ ($D[13:12]/0x301126 + 0x10 \bullet x$) or $SxID$ ($D4/0x301162 + 0x10 \bullet x$) settings. *2

* **$SxIN[1:0]$** : Ch.x Source Address Control Bits in the HSDMA Ch.x High-Order Source Address Setup Register ($D[13:12]/0x301126 + 0x10 \bullet x$)

* **$SxID$** : Ch.x Source Address Control Bit in the HSDMA Ch.x Control Register for ADV mode ($D4/0x301162 + 0x10 \bullet x$)

(3) The transfer counter is decremented.

(4) The HSDMA enable bit HSx_EN ($D0/0x30112C + 0x10 \bullet x$) is cleared and HSDMA cause-of-interrupt flag in ITC is set when the transfer counter reaches 0.

* **HSx_EN** : Ch.x Enable Bit in the HSDMA Ch.x Enable Register ($D0/0x30112C + 0x10 \bullet x$)

II BUS MODULES: HIGH-SPEED DMA (HSDMA)

*1: The data bus is placed in high-impedance state during reading from the I/O device. Furthermore, the external memory read/write address is delivered from the memory address registers in the control information S_xADRL and S_xADRH .

* S_xADRL : Ch.x Source Address[15:0] in the HSDMA Ch.x Low-Order Source Address Setup Register (STD mode: $D[15:0]/0x301124 + 0x10 \cdot x$, ADV mode: $D[15:0]/0x301164 + 0x10 \cdot x$)

* S_xADRH : Ch.x Source Address (high-order bits) in the HSDMA Ch.x High-Order Source Address Setup Register (STD mode: $D[11:0]/0x301126 + 0x10 \cdot x$, ADV mode: $D[15:0]/0x301166 + 0x10 \cdot x$)

*2: In standard mode, S_xID ($D4/0x301162 + 0x10 \cdot x$) is fixed at 0.

Successive transfer mode (single-address mode)

The channel for which $D_xMOD[1:0]$ ($D[15:14]/0x30112A + 0x10 \cdot x$) 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 II.1.6.2.2.

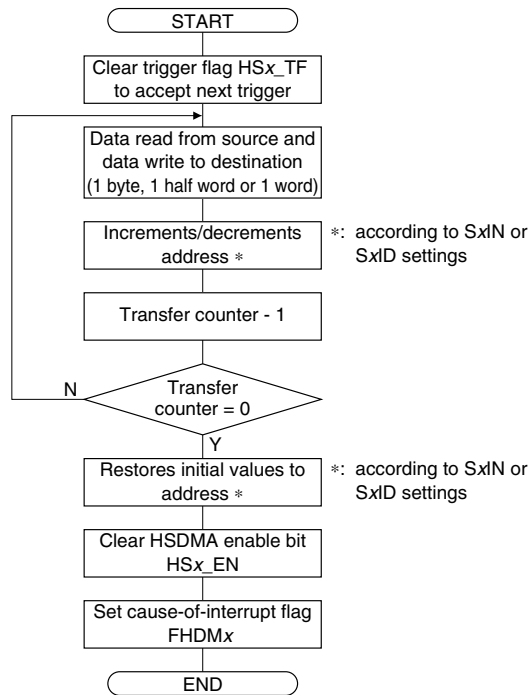


Figure II.1.6.2.2 Operation Flow in Successive Transfer Mode

(1) When a trigger is accepted, the trigger flag HS_xTF ($D0/0x30112E + 0x10 \cdot x$) is cleared. Data of the size set in the control information is read from the external memory or I/O device according to the specified direction and is written to the I/O device or external memory. *1

(2) The addresses are incremented or decremented according to the $S_xIN[1:0]$ ($D[13:12]/0x301126 + 0x10 \cdot x$) or S_xID ($D4/0x301162 + 0x10 \cdot x$) settings. *2

(3) The transfer counter is decremented.

(4) Steps (1) to (3) are repeated until the transfer counter reaches 0.

(5) The address returns to the initial value if $S_xIN[1:0]$ ($D[13:12]/0x301126 + 0x10 \cdot x$) is 10 or S_xID ($D4/0x301162 + 0x10 \cdot x$) is 1. *2

(6) The HSDMA enable bit HS_xEN ($D0/0x30112C + 0x10 \cdot x$) is cleared and HSDMA cause-of-interrupt flag in ITC is set when the transfer counter reaches 0.

*1: The data bus is placed in high-impedance state during reading from the I/O device. Furthermore, the external memory read/write address is delivered from the memory address registers in the control information S_xADRL and S_xADRH .

*2: In standard mode, S_xID ($D4/0x301162 + 0x10 \cdot x$) is fixed at 0.

Block transfer mode (single-address mode)

The channel for which $DxMOD[1:0]$ ($D[15:14]/0x30112A + 0x10 \bullet x$) 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 $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \bullet x$). 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 II.1.6.2.3.

* **BLKLENx[7:0]**: Ch.x Block Length Bits in the HSDMA Ch.x Transfer Counter Register ($D[7:0]/0x301120 + 0x10 \bullet x$)

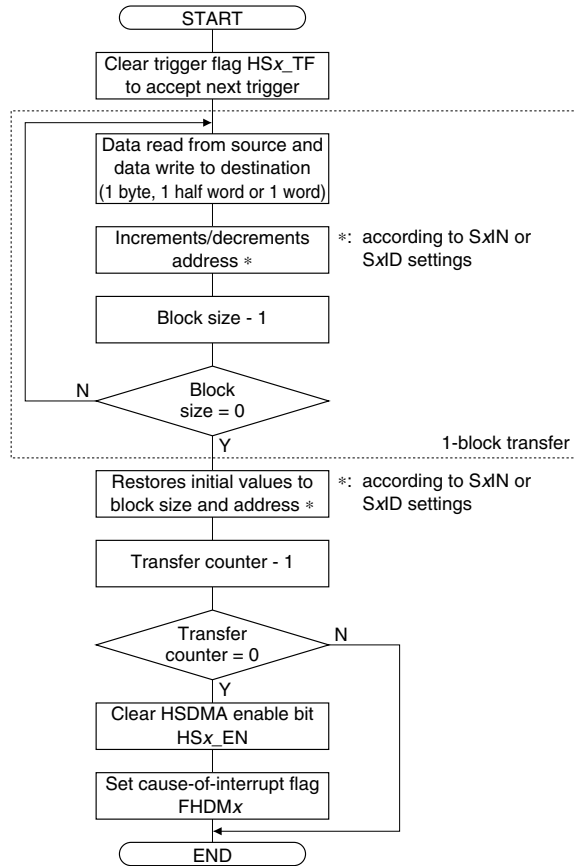


Figure II.1.6.2.3 Operation Flow in Block Transfer Mode

- (1) When a trigger is accepted, the trigger flag HSx_TF ($D0/0x30112E + 0x10 \bullet x$) is cleared. Data of the size set in the control information is read from the external memory or I/O device according to the specified direction and is written to the I/O device or external memory. *1
- (2) The address is incremented or decremented and $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \bullet x$) is decremented.
- (3) Steps (1) to (2) are repeated until $BLKLENx[7:0]$ ($D[7:0]/0x301120 + 0x10 \bullet x$) reaches 0.
- (4) The address returns to the initial value if $SxIN[1:0]$ ($D[13:12]/0x301126 + 0x10 \bullet x$) is 10 or $SxID$ ($D4/0x301162 + 0x10 \bullet x$) is 1. *2
- (5) The transfer counter is decremented.
- (6) Steps (1) to (5) are repeated until the transfer counter reaches 0.
- (7) The HSDMA enable bit HSx_EN ($D0/0x30112C + 0x10 \bullet x$) is cleared and HSDMA cause-of-interrupt flag in ITC is set when the transfer counter reaches 0.

*1: The data bus is placed in high-impedance state during reading from the I/O device. Furthermore, the external memory read/write address is delivered from the memory address registers in the control information $SxADRL$ and $SxADRH$.

*2: In standard mode, $SxID$ ($D4/0x301162 + 0x10 \bullet x$) is fixed at 0.

II.1.7 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 cause of interrupt.

Control registers of the interrupt controller

Table II.1.7.1 shows the control registers of the interrupt controller that are provided for each channel.

Table II.1.7.1 Control Registers of Interrupt Controller

Channel	Cause-of-interrupt flag	Interrupt enable register	Interrupt priority register
Ch. 0	FHDM0(D0/0x300281)	EHDM0(D0/0x300271)	PHSD0L[2:0](D[2:0]/0x300263)
Ch. 1	FHDM1(D1/0x300281)	EHDM1(D1/0x300271)	PHSD1L[2:0](D[6:4]/0x300263)
Ch. 2	FHDM2(D2/0x300281)	EHDM2(D2/0x300271)	PHSD2L[2:0](D[2:0]/0x300264)
Ch. 3	FHDM3(D3/0x300281)	EHDM3(D3/0x300271)	PHSD3L[2:0](D[6:4]/0x300264)

The HSDMA controller sets the HSDMA cause-of-interrupt 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 cause-of-interrupt 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 Section III.2, "Interrupt Controller (ITC)."

Intelligent DMA

Intelligent DMA (IDMA) can be invoked by the end-of-transfer interrupt source 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 II.1.7.2 Control Bits for IDMA Transfer

Channel	IDMA request bit	IDMA enable bit
Ch. 0	RHDM0(D4/0x300290)	DEHDM0(D4/0x300294)
Ch. 1	RHDM1(D5/0x300290)	DEHDM1(D5/0x300294)

If the IDMA request and enable bits are set to 1, IDMA is invoked through generation of a cause of interrupt. 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 Section II.2, "Intelligent DMA (IDMA)."

Trap vector

The trap vector addresses for causes of interrupt in each channel are set by default as follows:

Channel 0 end-of-transfer interrupt: 0xC00058

Channel 1 end-of-transfer interrupt: 0xC0005C

Channel 2 end-of-transfer interrupt: 0xC00060

Channel 3 end-of-transfer interrupt: 0xC00064

Note that the trap table base address can be modified using the TTBR register.

II.1.8 HSDMA Operating Clock

The HSDMA circuit is clocked by the DMA_CLK clock (= MCLK) generated by the CMU.
For details on how to control the clock, see Section III.1, “Clock Management Unit (CMU).”

Controlling the supply of the HSDMA operating clock

DMA_CLK is supplied to the DMA controller with default settings. When DMA transfer is not performed, the clock supply can be turned off using DMA_CKE (D1/0x301B04) to reduce the amount of power consumed on the chip.

* **DMA_CKE**: DMAC Clock Control Bit in the Gated Clock Control Register 1 (D1/0x301B04)

Setting DMA_CKE (D1/0x301B04) to 0 (initially 1) turns off the corresponding clock supply to the DMA controller.

Clock state in standby mode

The supply of DMA_CLK stops depending on type of standby mode.

HALT mode: DMA_CLK is supplied the same way as in normal mode (when DMA_CKE = 1).

SLEEP mode: The supply of DMA_CLK stops.

Therefore, the DMA controller also stops operating in SLEEP mode.

II.1.9 Details of Control Registers

Table II.1.9.1 List of HSDMA Registers

Address	Register name	Size	Function
0x00301120	HSDMA Ch.0 Transfer Counter Register (pHS0_CNT)	16	Sets Ch.0 low-order transfer counter data and block length.
0x00301122	HSDMA Ch.0 Control Register	16	Sets Ch.0 address mode and high-order transfer counter data.
0x00301124	HSDMA Ch.0 Low-Order Source Address Setup Register (pHS0_SADR)	16	Sets Ch.0 low-order source address.
0x00301126	HSDMA Ch.0 High-Order Source Address Setup Register	16	Sets Ch.0 high-order source address, transfer data size, and source address inc/dec condition.
0x00301128	HSDMA Ch.0 Low-Order Destination Address Setup Register (pHS0_DADR)	16	Sets Ch.0 low-order destination address.
0x0030112A	HSDMA Ch.0 High-Order Destination Address Setup Register	16	Sets Ch.0 high-order destination address, transfer mode, and destination address inc/dec condition.
0x0030112C	HSDMA Ch.0 Enable Register (pHS0_EN)	16	Enables Ch.0 DMA transfer.
0x0030112E	HSDMA Ch.0 Trigger Flag Register (pHS0_TF)	16	Ch.0 trigger status
0x00301130	HSDMA Ch.1 Transfer Counter Register (pHS1_CNT)	16	Sets Ch.1 low-order transfer counter data and block length.
0x00301132	HSDMA Ch.1 Control Register	16	Sets Ch.1 address mode and high-order transfer counter data.
0x00301134	HSDMA Ch.1 Low-Order Source Address Setup Register (pHS1_SADR)	16	Sets Ch.1 low-order source address.
0x00301136	HSDMA Ch.1 High-Order Source Address Setup Register	16	Sets Ch.1 high-order source address, transfer data size, and source address inc/dec condition.
0x00301138	HSDMA Ch.1 Low-Order Destination Address Setup Register (pHS1_DADR)	16	Sets Ch.1 low-order destination address.
0x0030113A	HSDMA Ch.1 High-Order Destination Address Setup Register	16	Sets Ch.1 high-order destination address, transfer mode, and destination address inc/dec condition.
0x0030113C	HSDMA Ch.1 Enable Register (pHS1_EN)	16	Enables Ch.1 DMA transfer.
0x0030113E	HSDMA Ch.1 Trigger Flag Register (pHS1_TF)	16	Ch.1 trigger status
0x00301140	HSDMA Ch.2 Transfer Counter Register (pHS2_CNT)	16	Sets Ch.2 low-order transfer counter data and block length.
0x00301142	HSDMA Ch.2 Control Register	16	Sets Ch.2 address mode and high-order transfer counter data.
0x00301144	HSDMA Ch.2 Low-Order Source Address Setup Register (pHS2_SADR)	16	Sets Ch.2 low-order source address.
0x00301146	HSDMA Ch.2 High-Order Source Address Setup Register	16	Sets Ch.2 high-order source address, transfer data size, and source address inc/dec condition.
0x00301148	HSDMA Ch.2 Low-Order Destination Address Setup Register (pHS2_DADR)	16	Sets Ch.2 low-order destination address.
0x0030114A	HSDMA Ch.2 High-Order Destination Address Setup Register	16	Sets Ch.2 high-order destination address, transfer mode, and destination address inc/dec condition.
0x0030114C	HSDMA Ch.2 Enable Register (pHS2_EN)	16	Enables Ch.2 DMA transfer.
0x0030114E	HSDMA Ch.2 Trigger Flag Register (pHS2_TF)	16	Ch.2 trigger status
0x00301150	HSDMA Ch.3 Transfer Counter Register (pHS3_CNT)	16	Sets Ch.3 low-order transfer counter data and block length.
0x00301152	HSDMA Ch.3 Control Register	16	Sets Ch.3 address mode and high-order transfer counter data.
0x00301154	HSDMA Ch.3 Low-Order Source Address Setup Register (pHS3_SADR)	16	Sets Ch.3 low-order source address.
0x00301156	HSDMA Ch.3 High-Order Source Address Setup Register	16	Sets Ch.3 high-order source address, transfer data size, and source address inc/dec condition.
0x00301158	HSDMA Ch.3 Low-Order Destination Address Setup Register (pHS3_DADR)	16	Sets Ch.3 low-order destination address.
0x0030115A	HSDMA Ch.3 High-Order Destination Address Setup Register	16	Sets Ch.3 high-order destination address, transfer mode, and destination address inc/dec condition.
0x0030115C	HSDMA Ch.3 Enable Register (pHS3_EN)	16	Enables Ch.3 DMA transfer.
0x0030115E	HSDMA Ch.3 Trigger Flag Register (pHS3_TF)	16	Ch.3 trigger status

Address	Register name	Size	Function
0x00301162	HSDMA Ch.0 Control Register (pHS0_ADVMODE) for ADV mode	16	Selects Ch.0 ADV mode functions.
0x00301164	HSDMA Ch.0 Low-Order Source Address Setup Register (pHS0_AD_SADR) for ADV mode	16	Sets Ch.0 low-order source address for ADV mode.
0x00301166	HSDMA Ch.0 High-Order Source Address Setup Register for ADV mode	16	Sets Ch.0 high-order source address for ADV mode.
0x00301168	HSDMA Ch.0 Low-Order Destination Address Setup Register (pHS0_ADV_DADR) for ADV mode	16	Sets Ch.0 low-order destination address for ADV mode.
0x0030116A	HSDMA Ch.0 High-Order Destination Address Setup Register for ADV mode	16	Sets Ch.0 high-order destination address for ADV mode.
0x00301172	HSDMA Ch.1 Control Register (pHS1_ADVMODE) for ADV mode	16	Selects Ch.1 ADV mode functions.
0x00301174	HSDMA Ch.1 Low-Order Source Address Setup Register (pHS1_AD_SADR) for ADV mode	16	Sets Ch.1 low-order source address for ADV mode.
0x00301176	HSDMA Ch.1 High-Order Source Address Setup Register for ADV mode	16	Sets Ch.1 high-order source address for ADV mode.
0x00301178	HSDMA Ch.1 Low-Order Destination Address Setup Register (pHS1_ADV_DADR) for ADV mode	16	Sets Ch.1 low-order destination address for ADV mode.
0x0030117A	HSDMA Ch.1 High-Order Destination Address Setup Register for ADV mode	16	Sets Ch.1 high-order destination address for ADV mode.
0x00301182	HSDMA Ch.2 Control Register (pHS2_ADVMODE) for ADV mode	16	Selects Ch.2 ADV mode functions.
0x00301184	HSDMA Ch.2 Low-Order Source Address Setup Register (pHS2_AD_SADR) for ADV mode	16	Sets Ch.2 low-order source address for ADV mode.
0x00301186	HSDMA Ch.2 High-Order Source Address Setup Register for ADV mode	16	Sets Ch.2 high-order source address for ADV mode.
0x00301188	HSDMA Ch.2 Low-Order Destination Address Setup Register (pHS2_ADV_DADR) for ADV mode	16	Sets Ch.2 low-order destination address for ADV mode.
0x0030118A	HSDMA Ch.2 High-Order Destination Address Setup Register for ADV mode	16	Sets Ch.2 high-order destination address for ADV mode.
0x00301192	HSDMA Ch.3 Control Register (pHS3_ADVMODE) for ADV mode	16	Selects Ch.3 ADV mode functions.
0x00301194	HSDMA Ch.3 Low-Order Source Address Setup Register (pHS3_AD_SADR) for ADV mode	16	Sets Ch.3 low-order source address for ADV mode.
0x00301196	HSDMA Ch.3 High-Order Source Address Setup Register for ADV mode	16	Sets Ch.3 high-order source address for ADV mode.
0x00301198	HSDMA Ch.3 Low-Order Destination Address Setup Register (pHS3_ADV_DADR) for ADV mode	16	Sets Ch.3 low-order destination address for ADV mode.
0x0030119A	HSDMA Ch.3 High-Order Destination Address Setup Register for ADV mode	16	Sets Ch.3 high-order destination address for ADV mode.
0x0030119C	HSDMA STD/ADV Mode Select Register (pHS_CNTLMODE)	16	Selects standard or advanced mode.
0x0030119E	DMA Sequential Access Time Register (pHS_ACCTIME)	16	Sets sequential access time for IDMA and HSDMA.

The following describes each HSDMA control register.

The HSDMA control registers are mapped in the 16-bit device area from 0x301120 to 0x30119E, and can be accessed in units of half-words or bytes.

Note: When setting the HSDMA control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x301120–0x301150: HSDMA Ch.x Transfer Counter Registers (pHSx_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.x transfer counter register (pHSx_CNT)	00301120 00301150 (HW)	D15	TCx_L7	Ch.x transfer counter[7:0] (block transfer mode)		0	R/W	
		D14	TCx_L6					
		D13	TCx_L5	Ch.x transfer counter[15:8] (single/successive transfer mode)				
		D12	TCx_L4					
		D11	TCx_L3					
		D10	TCx_L2					
		D9	TCx_L1					
		D8	TCx_L0					
	D7	BLKLENx7	Ch.x block length (block transfer mode)		0	R/W		
	D6	BLKLENx6						
	D5	BLKLENx5	Ch.x transfer counter[7:0] (single/successive transfer mode)					
	D4	BLKLENx4						
	D3	BLKLENx3						
	D2	BLKLENx2						
	D1	BLKLENx1						
	D0	BLKLENx0						

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 3.

- 0x301120 HSDMA Ch.0 Transfer Counter Register (pHS0_CNT)
- 0x301130 HSDMA Ch.1 Transfer Counter Register (pHS1_CNT)
- 0x301140 HSDMA Ch.2 Transfer Counter Register (pHS2_CNT)
- 0x301150 HSDMA Ch.3 Transfer Counter Register (pHS3_CNT)

D[15:8] TCx_L[7:0]: Ch.x Transfer Counter Bits

Set the data transfer count. (Default: 0x00)

In block transfer mode, TCx_L[7:0] is bits[7:0] of the transfer counter. In single or successive transfer mode, TCx_L[7:0] is bits[15:8] of the transfer counter.

This counter is decremented each time a DMA transfer in the corresponding channel is performed. When the counter reaches 0, a cause of interrupt 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 0xFFFFF).

Be sure to disable DMA transfers (HSx_EN (D0/0x30112C + 0x10•x) = 0) before writing and reading to and from the counter.

D[7:0] BLKLENx[7:0]: Ch.x Block Length Bits

In block transfer mode, these bits are used to specify a transfer block size. (Default: 0x00)

A transfer operation invoked by one trigger is completed after transferring one block of data of the size set by BLKLENx[7:0].

In single or successive transfer mode, these bits are used to specify the 8 low-order bits of the transfer counter.

Note: When performing data transfer in block transfer mode, the block size must not be set to 0.

0x301122–0x301152: HSDMA Ch.x Control Registers

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
HSDMA Ch.x control register	00301122	D15	DUALM_x	Ch.x address mode selection	1	Dual addr	0	Single addr	0	R/W	
		D14	DxDIR	D) Invalid S) Ch.x transfer direction control							
Note: D) Dual address mode S) Single address mode	00301152 (HW)	D13–8	–	reserved							0 when being read.
		D7	TC_x_H7	Ch.x transfer counter[15:8] (block transfer mode)					0	R/W	
		D6	TC_x_H6						0		
		D5	TC_x_H5	Ch.x transfer counter[23:16] (single/successive transfer mode)					0		
		D4	TC_x_H4						0		
		D3	TC_x_H3						0		
		D2	TC_x_H2						0		
		D1	TC_x_H1						0		
		D0	TC_x_H0					0			

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 3.

- 0x301122 HSDMA Ch.0 Control Register
- 0x301132 HSDMA Ch.1 Control Register
- 0x301142 HSDMA Ch.2 Control Register
- 0x301152 HSDMA Ch.3 Control Register

D15 **DUALM_x: Ch.x Address Mode Select Bit**

Select an address mode.

1 (R/W): Dual-address mode

0 (R/W): Single-address mode (default)

When 1 is written to DUALM_x, 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.

D14 **DxDIR: Ch.x Transfer Direction Control Bit**

Control the direction of data transfer in single-address mode.

1 (R/W): Memory write

0 (R/W): Memory read (default)

Data transfer from an external I/O device to external memory (or an external/internal I/O) is performed by writing 1 to DxDIR. Data transfer from external memory (or an external/internal I/O) to an external I/O is performed by writing 0.

This bit is effective only in single-address mode.

D[13:8] **Reserved**

D[7:0] **TC_x_H[7:0]: Ch.x Transfer Counter Bits**

Set the data transfer count. (Default: 0x00)

In block transfer mode, TC_x_H[7:0] is bits[15:8] of the transfer counter. In single or successive transfer mode, TC_x_H[7:0] is bits[23:16] of the transfer counter.

This counter is decremented each time a DMA transfer in the corresponding channel is performed. When the counter reaches 0, a cause of interrupt is generated. In single-address mode, the end-of-transfer signal is output from the #DMAEND_x pin at the same time. Even when the counter is 0, a DMA request is accepted and the counter is decremented to 0xFFFF (or 0xFFFFF).

Be sure to disable DMA transfers (HS_x_EN (D0/0x30112C + 0x10•x) = 0) before writing and reading to and from the counter.

0x301124–0x301154: HSDMA Ch.x Low-Order Source Address Setup Registers (pHSx_SADR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.x low-order source address setup register (pHSx_SADR) Note: D) Dual address mode S) Single address mode	00301124	D15	SxADRL15	D) Ch.x source address[15:0]		0	R/W	
		D14	SxADRL14	S) Ch.x memory address[15:0]				
	00301154 (HW)	D13	SxADRL13					
		D12	SxADRL12					
		D11	SxADRL11					
		D10	SxADRL10					
		D9	SxADRL9					
		D8	SxADRL8					
		D7	SxADRL7					
		D6	SxADRL6					
		D5	SxADRL5					
		D4	SxADRL4					
		D3	SxADRL3					
		D2	SxADRL2					
		D1	SxADRL1					
		D0	SxADRL0					

Note: The letter ‘x’ in bit names, etc., denotes a channel number from 0 to 3.

- 0x301124 HSDMA Ch.0 Low-Order Source Address Setup Register (pHS0_SADR)
- 0x301134 HSDMA Ch.1 Low-Order Source Address Setup Register (pHS1_SADR)
- 0x301144 HSDMA Ch.2 Low-Order Source Address Setup Register (pHS2_SADR)
- 0x301154 HSDMA Ch.3 Low-Order Source Address Setup Register (pHS3_SADR)

D[15:0] SxADRL[15:0]: Ch.x Source Address[15:0] (for standard mode)

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[15:0] to set the 16 low-order bits of the address.

Be sure to disable DMA transfers (HSx_EN (D0/0x30112C + 0x10•x) = 0) before writing or reading to and from these registers.

The address is incremented or decremented (as set by SxIN[1:0] (D[13:12]/0x301126 + 0x10•x) or SxID (D4/0x301162 + 0x10•x)) according to the transfer data size each time a DMA transfer in the corresponding channel is performed.

Notes: • The following areas cannot be used for DMA transfer:

Dual-address mode: Area 0, Area 2

Single-address mode: Area 0, Area 1, Area 2, Area 3, Area 6

- Single-address mode does not allow data transfer between memory devices.
- Single-address mode does not support the external memory area that is configured for SDRAM.
- Use SxADRL[15:0] (D[15:0]/0x301164 + 0x10•x) and SxADRH[15:0] (D[15:0]/0x301166 + 0x10•x) for specifying an address in advanced mode.

0x301126–0x301156: HSDMA Ch.x High-Order Source Address Setup Registers

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks			
HSDMA Ch.x high-order source address setup register	00301126	D15	–	reserved	–		–	–	0 when being read.			
		D14	DATSIZE _x	Ch.x transfer data size	1	Half word	0	Byte	0	R/W		
Note: D) Dual address mode S) Single address mode	00301156 (HW)	D13	SxIN1	D) Ch.x source address control	SxIN[1:0]	Inc/dec	0	R/W				
		D12	SxIN0	S) Ch.x memory address control						11	Inc.(no init)	0
										10	Inc.(init)	
										01	Dec.(no init)	
					00	Fixed						
		D11	SxADDRH11	D) Ch.x source address[27:16]	S) Ch.x memory address[27:16]					R/W		
		D10	SxADDRH10									
		D9	SxADDRH9									
		D8	SxADDRH8									
		D7	SxADDRH7									
		D6	SxADDRH6									
		D5	SxADDRH5									
		D4	SxADDRH4									
		D3	SxADDRH3									
		D2	SxADDRH2									
		D1	SxADDRH1									
D0	SxADDRH0											

Note: The letter ‘x’ in bit names, etc., denotes a channel number from 0 to 3.

- 0x301126 HSDMA Ch.0 High-Order Source Address Setup Register
- 0x301136 HSDMA Ch.1 High-Order Source Address Setup Register
- 0x301146 HSDMA Ch.2 High-Order Source Address Setup Register
- 0x301156 HSDMA Ch.3 High-Order Source Address Setup Register

D15 Reserved

D14 DATSIZE_x: Ch.x Transfer Data Size Select Bit

Select the data size to be transferred.

1 (R/W): Half-word

0 (R/W): Byte (default)

The transfer data size is set to 16 bits by writing 1 to DATSIZE_x and set to 8 bits by writing 0.

Note: In advanced mode, this bit is effective when WORDSIZE_x (D0/0x301162 + 0x10•x) = 0. The setting of this bit is ignored when WORDSIZE_x (D0/0x301162 + 0x10•x) = 1 and the transfer data size is set to 32 bits.

In standard mode, this bit is always effective regardless of the WORDSIZE_x (D0/0x301162 + 0x10•x) setting.

D[13:12] SxIN[1:0]: Ch.x Source Address Control Bits

Control the incrementing or decrementing of the memory address.

Table II.1.9.2 Address Control

SxIN1	SxIN0	Address control
1	1	Increment without initialization
1	0	Increment with initialization
0	1	Decrement without initialization
0	0	Fixed

(Default: 0b00)

In dual-address mode, this setting applies to the source address. In single-address mode, this setting applies to the external memory address.

When “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 “Increment without initialization” (11) is selected, the source address is incremented by an amount equal to the data size set by DATSIZE_x (D14) or WORDSIZE_x (D0/0x301162 + 0x10•x) when one data transfer is completed.

II BUS MODULES: HIGH-SPEED DMA (HSDMA)

When “Decrement without initialization” (01) is selected, the source address is decremented in the same way.

When “Increment with initialization” (10) is selected, the source address is incremented by an amount equal to the data size set by DATSIZE_x (D14) or WORDSIZE_x (D0/0x301162 + 0x10•x) when one data transfer is completed. In single transfer mode, the address that has been incremented during transfer does not return to the initial value. In successive transfer modes, the incremented address returns to the initial value when the specified number of transfers is completed. In block transfer mode, the incremented address returns to the initial value when the block transfer is completed.

Note: In advanced mode, these bits are effective when SxID (D4/0x301162 + 0x10•x) = 0. The setting of these bits is ignored when SxID (D4/0x301162 + 0x10•x) = 1 and “Decrement with initialization” is selected.

In standard mode, this bit is always effective regardless of the SxID (D4/0x301162 + 0x10•x) setting.

D[11:0] SxADRH[11:0]: Ch.x Source Address[27:16] (for standard mode)

In dual-address mode, these bits are used to specify 12 high-order bits of source address. In single-address mode, 12 high-order bits of external memory address at the destination or source of transfer is specified.

See SxADRL[15:0] (D[15:0]/0x301124 + 0x10•x) for more information.

0x301128–0x301158: HSDMA Ch.x Low-Order Destination Address Setup Registers (pHSx_DADR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.x low-order destination address setup register (pHSx_DADR) Note: D) Dual address mode S) Single address mode	00301128 00301158 (HW)	D15	DxADRL15	D) Ch.x destination address[15:0]		0	R/W	
		D14	DxADRL14	S) Invalid		0		
		D13	DxADRL13			0		
		D12	DxADRL12			0		
		D11	DxADRL11			0		
		D10	DxADRL10			0		
		D9	DxADRL9			0		
		D8	DxADRL8			0		
		D7	DxADRL7			0		
		D6	DxADRL6			0		
		D5	DxADRL5			0		
		D4	DxADRL4			0		
		D3	DxADRL3			0		
		D2	DxADRL2			0		
		D1	DxADRL1			0		
		D0	DxADRL0			0		

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 3.

0x301128 HSDMA Ch.0 Low-Order Destination Address Setup Register (pHS0_DADR)

0x301138 HSDMA Ch.1 Low-Order Destination Address Setup Register (pHS1_DADR)

0x301148 HSDMA Ch.2 Low-Order Destination Address Setup Register (pHS2_DADR)

0x301158 HSDMA Ch.3 Low-Order Destination Address Setup Register (pHS3_DADR)

D[15:0] DxADRL[15:0]: Ch.x Destination Address[15:0] (for standard mode)

In dual-address mode, these bits are used to specify a destination address.

Use DxADRL[15:0] to set the 16 low-order bits of the address.

Be sure to disable DMA transfers ($HSx_EN(D0/0x30112C + 0x10 \bullet x) = 0$) before writing or reading to and from these registers.

The address is incremented or decremented (as set by DxIN[1:0] ($D[13:12]/0x30112A + 0x10 \bullet x$) or DxID ($D5/0x301162 + 0x10 \bullet x$)) according to the transfer data size each time a DMA transfer in the corresponding channel is performed.

Notes: • In single-address mode, these bits are not used.

• The following areas cannot be specified for destination addresses:
Area 0 (A0RAM), Area 2

• Use DxADRL[15:0] ($D[15:0]/0x301168 + 0x10 \bullet x$) and DxADRH[15:0] ($D[15:0]/0x30116A + 0x10 \bullet x$) for specifying an address in advanced mode.

0x30112A–0x30115A: HSDMA Ch.x High-Order Destination Address Setup Registers

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
HSDMA Ch.x high-order destination address setup register Note: D) Dual address mode S) Single address mode	0030112A 0030115A (HW)	D15	DxMOD1	Ch.x transfer mode	DxMOD[1:0]	Mode	0	R/W		
		D14	DxMOD0						11	Invalid
									10	Block
									01	Successive
					00	Single				
			D13	DxIN1	D) Ch.x destination address control S) Invalid	DxIN[1:0]	Inc/dec	0	R/W	
		D12	DxIN0	11						Inc.(no init)
				10						Inc.(init)
				01						Dec.(no init)
					00	Fixed				
			D11	DxADRH11	D) Ch.x destination address[27:16] S) Invalid			0	R/W	
			D10	DxADRH10						
		D9	DxADRH9							
		D8	DxADRH8							
		D7	DxADRH7							
		D6	DxADRH6							
		D5	DxADRH5							
		D4	DxADRH4							
		D3	DxADRH3							
		D2	DxADRH2							
		D1	DxADRH1							
		D0	DxADRH0							

Note: The letter ‘x’ in bit names, etc., denotes a channel number from 0 to 3.

- 0x30112A HSDMA Ch.0 High-Order Destination Address Setup Register
- 0x30113A HSDMA Ch.1 High-Order Destination Address Setup Register
- 0x30114A HSDMA Ch.2 High-Order Destination Address Setup Register
- 0x30115A HSDMA Ch.3 High-Order Destination Address Setup Register

D[15:14] DxMOD[1:0]: Ch.x Transfer Mode Select Bits

Select a transfer mode.

Table II.1.9.3 Transfer Mode

DxMOD1	DxMOD0	Mode
1	1	Invalid
1	0	Block transfer mode
0	1	Successive transfer mode
0	0	Single transfer mode

(Default: 0b00)

In single transfer mode, a transfer operation invoked by one trigger is completed after transferring one unit of data of the size set by DATSIZE_x (D14/0x301126 + 0x10•x) or WORDSIZE_x (D0/0x301162 + 0x10•x). 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 BLKLEN_x[7:0] (D[7:0]/0x301120 + 0x10•x).

D[13:12] DxIN[1:0]: Ch.x Destination Address Control Bits

Control the incrementing or decrementing of the memory address.

Table II.1.9.4 Address Control

DxIN1	DxIN0	Address control
1	1	Increment without initialization
1	0	Increment with initialization
0	1	Decrement without initialization
0	0	Fixed

(Default: 0b00)

In dual-address mode, this setting applies to the destination address.

When “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 “Increment without initialization” (11) is selected, the destination address is incremented by an amount equal to the data size set by DATSIZE_x (D14/0x301126 + 0x10•x) or WORDSIZE_x (D0/0x301162 + 0x10•x) when one data transfer is completed.

When “Decrement without initialization” (01) is selected, the destination address is decremented in the same way.

When “Increment with initialization” (10) is selected, the destination address is incremented by an amount equal to the data size set by DATSIZE_x (D14/0x301126 + 0x10•x) or WORDSIZE_x (D0/0x301162 + 0x10•x) when one data transfer is completed. In single transfer mode, the address that has been incremented during transfer does not return to the initial value. In successive transfer modes, the incremented address returns to the initial value when the specified number of transfers is completed. In block transfer mode, the incremented address returns to the initial value when the block transfer is completed.

In single-address mode, these bits are not used.

Note: In advanced mode, these bits are effective when DxID (D5/0x301162 + 0x10•x) = 0. The setting of these bits is ignored when DxID (D5/0x301162 + 0x10•x) = 1 and “Decrement with initialization” is selected.

In standard mode, this bit is always effective regardless of the DxID (D5/0x301162 + 0x10•x) setting.

D[11:0] DxADRH[11:0]: Ch.x Destination Address[27:16] (for standard mode)

In dual-address mode, these bits are used to specify 12 high-order bits of destination address.

See DxADRL[15:0] (D[15:0]/0x301128 + 0x10•x) for more information.

In single-address mode, these bits are not used.

0x30112C–0x30115C: HSDMA Ch.x Enable Registers (pHSx_EN)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
HSDMA Ch.x enable register (pHSx_EN)	0030112C	D15–1	–	reserved	–			–	–	0 when being read.
	0030115C (HW)	D0	HSx_EN	Ch.x enable	1	Enable	0	Disable	0	R/W

Note: The letter ‘x’ in bit names, etc., denotes a channel number from 0 to 3.

0x30112C HSDMA Ch.0 Enable Register (pHS0_EN)

0x30113C HSDMA Ch.1 Enable Register (pHS1_EN)

0x30114C HSDMA Ch.2 Enable Register (pHS2_EN)

0x30115C HSDMA Ch.3 Enable Register (pHS3_EN)

D[15:1] Reserved

D0 HSx_EN: Ch.x Enable Bit

Enable a DMA transfer.

1 (R/W): Enable

0 (R/W): Disable (default)

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 source. 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.

0x30112E–0x30115E: HSDMA Ch.x Trigger Flag Registers (pHSx_TF)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
HSDMA Ch.x trigger flag register (pHSx_TF)	0030112E 0030115E (HW)	D15–1	–	reserved	–			–	–	0 when being read.
		D0	HSx_TF	Ch.x trigger flag clear (writing) Ch.x trigger flag status (reading)	1 1	Clear Set	0 0	No operation Cleared	0	R/W

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 3.

0x30112E HSDMA Ch.0 Trigger Flag Register (pHS0_TF)

0x30113E HSDMA Ch.1 Trigger Flag Register (pHS1_TF)

0x30114E HSDMA Ch.2 Trigger Flag Register (pHS2_TF)

0x30115E HSDMA Ch.3 Trigger Flag Register (pHS3_TF)

D[15:1] Reserved**D0 HSx_TF: Ch.x Trigger Flag Clear/Status Bit**

These bits are used to check and clear the trigger flag status.

1 (R): Trigger flag has been set

0 (R): Trigger flag has been cleared (default)

1 (W): Clear trigger flag

0 (W): Has no effect

The trigger flag is set when a trigger 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.

0x301162–0x301192: HSDMA Ch.x Control Registers (pHSx_ADV MODE) for ADV mode

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.x control register (pHSx_ADV MODE) for ADV mode Note: D) Dual mode S) Single mode	00301162	D15–6	–	reserved	–	–	–	0 when being read.
		D5	DxID	D) Ch.x destination address control S) Invalid	1 Decrement (with init.) 0 DxIN[1:0] setting	0	R/W	
	00301192 (HW)	D4	SxID	D) Ch.x source address control S) Ch.x memory address control	1 Decrement (with init.) 0 SxIN[1:0] setting	0	R/W	
		D3–1	–	reserved	–	–	–	0 when being read.
	D0	WORDSIZE_x	Ch.x transfer data size	1 Word 0 DATSIZE _x setting	0	R/W		

Notes: • This register is effective only in advanced mode (HSDMAADV (D0/0x30119C) = 1).

- The letter ‘x’ in bit names, etc., denotes a channel number from 0 to 3.

0x301162 HSDMA Ch.0 Control Register (pHS0_ADV MODE)

0x301172 HSDMA Ch.1 Control Register (pHS1_ADV MODE)

0x301182 HSDMA Ch.2 Control Register (pHS2_ADV MODE)

0x301192 HSDMA Ch.3 Control Register (pHS3_ADV MODE)

D[15:6] Reserved

D5 **DxID: Ch.x Destination Address Control Bit**

Enable the address decrement function with initialization for destination address.

1 (R/W): Decrement with initialization

0 (R/W): DxIN[1:0] setting is effective (default)

When this bit is set to 1 in dual-address mode, the destination address decrement function with initialization is enabled. The destination address is decremented by an amount equal to the data size set by DATSIZE_x (D14/0x301126 + 0x10•x) or WORDSIZE_x (D0/0x301162 + 0x10•x) when one data transfer is completed. In single transfer mode, the address that has been decremented during transfer does not return to the initial value. In successive transfer modes, the decremented address returns to the initial value when the specified number of transfers is completed. In block transfer mode, the decremented address returns to the initial value when the block transfer is completed.

When this bit is set to 0, the condition set by DxIN[1:0] (D[13:12]/0x30112A + 0x10•x) is effective.

In single-address mode, this bit is not used.

D4 **SxID: Ch.x Source Address Control Bit**

Enable the address decrement function with initialization for source address.

1 (R/W): Decrement with initialization

0 (R/W): SxIN[1:0] setting (default)

In dual-address mode, this setting applies to the source address. In single-address mode, this setting applies to the external memory address.

When this bit is set to 1, the address decrement function with initialization is enabled. The source/external memory address is decremented by an amount equal to the data size set by DATSIZE_x (D14/0x301126 + 0x10•x) or WORDSIZE_x (D0/0x301162 + 0x10•x) when one data transfer is completed. In single transfer mode, the address that has been decremented during transfer does not return to the initial value. In successive transfer modes, the decremented address returns to the initial value when the specified number of transfers is completed. In block transfer mode, the decremented address returns to the initial value when the block transfer is completed.

When this bit is set to 0, the condition set by SxIN[1:0] (D[13:12]/0x301126 + 0x10•x) is effective.

D[3:1] Reserved

D0 WORDSIZE_x: Ch.x Transfer Data Size Select Bit

Select the data size to be transferred.

1 (R/W): Word

0 (R/W): DATSIZE_x setting is effective (default)

The transfer data size is set to 32 bits by writing 1 to this bit. When this bit is set to 0, the size set by DATSIZE_x (D14/0x301126 + 0x10•x) is effective.

0x301164–0x301196: HSDMA Ch.x Source Address Setup Registers (pHS_x_AD_SADR) for ADV mode

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.x low-order source address setup register (pHS _x _AD_SADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301164 00301194 (HW)	D15	SxADRL15	D) Ch.x source address[15:0]		0	R/W	
		D14	SxADRL14	S) Ch.x memory address[15:0]				
		D13	SxADRL13					
		D12	SxADRL12					
		D11	SxADRL11					
		D10	SxADRL10					
		D9	SxADRL9					
		D8	SxADRL8					
		D7	SxADRL7					
		D6	SxADRL6					
		D5	SxADRL5					
		D4	SxADRL4					
		D3	SxADRL3					
		D2	SxADRL2					
		D1	SxADRL1					
		D0	SxADRL0					
HSDMA Ch.x high-order source address setup register for ADV mode Note: D) Dual address mode S) Single address mode	00301166 00301196 (HW)	D15	SxADRH15	D) Ch.x source address[31:16]		0	R/W	
		D14	SxADRH14	S) Ch.x memory address[31:16]				
		D13	SxADRH13					
		D12	SxADRH12					
		D11	SxADRH11					
		D10	SxADRH10					
		D9	SxADRH9					
		D8	SxADRH8					
		D7	SxADRH7					
		D6	SxADRH6					
		D5	SxADRH5					
		D4	SxADRH4					
		D3	SxADRH3					
		D2	SxADRH2					
		D1	SxADRH1					
		D0	SxADRH0					

Notes: • This register is effective only in advanced mode (HSDMAADV (D0/0x30119C) = 1).

- The letter 'x' in bit names, etc., denotes a channel number from 0 to 3.

0x301164 HSDMA Ch.0 Low-Order Source Address Setup Register (pHS0_AD_SADR)

0x301166 HSDMA Ch.0 High-Order Source Address Setup Register for ADV mode

0x301174 HSDMA Ch.1 Low-Order Source Address Setup Register (pHS1_AD_SADR)

0x301176 HSDMA Ch.1 High-Order Source Address Setup Register for ADV mode

0x301184 HSDMA Ch.2 Low-Order Source Address Setup Register (pHS2_AD_SADR)

0x301186 HSDMA Ch.2 High-Order Source Address Setup Register for ADV mode

0x301194 HSDMA Ch.3 Low-Order Source Address Setup Register (pHS3_AD_SADR)

0x301196 HSDMA Ch.3 High-Order Source Address Setup Register for ADV mode

D[15:0]/0x301164–0x301194 SxADRL[15:0]: Ch.x Low-Order Source Address[15:0]

D[15:0]/0x301166–0x301196 SxADRH[15:0]: Ch.x High-Order Source Address[31:16]

In dual-address mode, these bits are used to specify a 32-bit source address. In single-address mode, a 32-bit external memory address at the destination or source of transfer is specified.

Be sure to disable DMA transfers (HS_x_EN (D0/0x30112C + 0x10•x) = 0) before writing or reading to and from these registers.

The address is incremented or decremented (as set by SxIN[1:0] (D[13:12]/0x301126 + 0x10•x) or SxID (D4/0x301162 + 0x10•x)) according to the transfer data size each time a DMA transfer in the corresponding channel is performed.

- Notes:**
- The following areas cannot be used for DMA transfer:
Dual-address mode: Area 0, Area 2
Single-address mode: Area 0, Area 1, Area 2, Area 3, Area 6
 - Single-address mode does not allow data transfer between memory devices.
 - Single-address mode does not support the external memory area that is configured for SDRAM.
 - Use $SxADRL[15:0]$ ($D[15:0]/0x301124 + 0x10 \bullet x$) and $SxADRH[11:0]$ ($D[11:0]/0x301126 + 0x10 \bullet x$) for specifying an address in standard mode.

0x301168–0x30119A: HSDMA Ch.x Destination Address Setup Registers (pHSx_ADV_DADR) for ADV mode

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.x low-order destination address setup register (pHSx_ADV_DADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301168 00301198 (HW)	D15	DxADRL15	D) Ch.x destination address[15:0] S) Invalid		0	R/W	
		D14	DxADRL14					
		D13	DxADRL13					
		D12	DxADRL12					
		D11	DxADRL11					
		D10	DxADRL10					
		D9	DxADRL9					
		D8	DxADRL8					
		D7	DxADRL7					
		D6	DxADRL6					
		D5	DxADRL5					
		D4	DxADRL4					
		D3	DxADRL3					
		D2	DxADRL2					
		D1	DxADRL1					
		D0	DxADRL0					
HSDMA Ch.x high-order destination address setup register for ADV mode Note: D) Dual address mode S) Single address mode	0030116A 0030119A (HW)	D15	DxADRH15	D) Ch.x destination address[31:16] S) Invalid		0	R/W	
		D14	DxADRH14					
		D13	DxADRH13					
		D12	DxADRH12					
		D11	DxADRH11					
		D10	DxADRH10					
		D9	DxADRH9					
		D8	DxADRH8					
		D7	DxADRH7					
		D6	DxADRH6					
		D5	DxADRH5					
		D4	DxADRH4					
		D3	DxADRH3					
		D2	DxADRH2					
		D1	DxADRH1					
		D0	DxADRH0					

Notes: • This register is effective only in advanced mode (HSDMAADV (D0/0x30119C) = 1).

- The letter 'x' in bit names, etc., denotes a channel number from 0 to 3.

0x301168 HSDMA Ch.0 Low-Order Destination Address Setup Register (pHS0_ADV_DADR)

0x30116A HSDMA Ch.0 High-Order Destination Address Setup Register for ADV mode

0x301178 HSDMA Ch.1 Low-Order Destination Address Setup Register (pHS1_ADV_DADR)

0x30117A HSDMA Ch.1 High-Order Destination Address Setup Register for ADV mode

0x301188 HSDMA Ch.2 Low-Order Destination Address Setup Register (pHS2_ADV_DADR)

0x30118A HSDMA Ch.2 High-Order Destination Address Setup Register for ADV mode

0x301198 HSDMA Ch.3 Low-Order Destination Address Setup Register (pHS3_ADV_DADR)

0x30119A HSDMA Ch.3 High-Order Destination Address Setup Register for ADV mode

D[15:0]/0x301168–0x301198 DxADRL[15:0]: Ch.x Destination Address[15:0]

D[15:0]/0x30116A–0x30119A DxADRH[15:0]: Ch.x Destination Address[31:16]

In dual-address mode, these bits are used to specify a 32-bit destination address.

Be sure to disable DMA transfers (HSx_EN (D0/0x30112C + 0x10•x) = 0) before writing or reading to and from these registers.

The address is incremented or decremented (as set by DxIN[1:0] (D[13:12]/0x30112A + 0x10•x) or DxID (D5/0x301162 + 0x10•x)) according to the transfer data size each time a DMA transfer in the corresponding channel is performed.

- Notes:**
- In single-address mode, these bits are not used.
 - The following areas cannot be specified for destination addresses:
Area 0 (A0RAM), Area 2
 - Use $DxADR[15:0]$ ($D[15:0]/0x301128 + 0x10 \bullet x$) and $DxADR[11:0]$ ($D[11:0]/0x30112A + 0x10 \bullet x$) for specifying an address in standard mode.

0x30119C: HSDMA STD/ADV Mode Select Register (pHS_CNTLMODE)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
HSDMA STD/ADV mode select register (pHS_CNTLMODE)	0030119C (HW)	D15-1	–	reserved	–			–	–	0 when being read.
		D0	HSDMAADV	Standard mode/advanced mode select	1	Advanced mode	0	Standard mode	0	R/W

D[15:1] Reserved

D0 HSDMAADV: Standard/Advanced Mode Select Bit

Select standard or advanced mode.

1 (R/W): Advanced mode

0 (R/W): Standard mode (default)

The HSDMA in the S1C33E08 is extended from that of the C33 STD models. The S1C33E08 HSDMA has two operating modes, standard (STD) mode of which functions are compatible with the existing C33 STD models and an advanced (ADV) mode allowing use of the extended functions. Table II.1.9.5 shows differences between standard mode and advanced mode.

Table II.1.9.5 Differences between Standard Mode and Advanced Mode

Function	Standard mode	Advanced mode
Source/destination address bit width	28 bits	32 bits
Word (32-bit) data transfer	Unavailable	Available
Address decrement function with initialization	Unavailable	Available

To configure the HSDMA in advanced mode, set this bit to 1. The control registers (0x301162–0x30119A) for the extended functions are enabled to write after this setting.

- Notes:**
- Be sure to use the control registers for advanced mode when the HSDMA is set to advanced mode.
 - Standard or advanced mode currently set is applied to all the HSDMA channels. It cannot be selected for each channel individually.

0x30119E: DMA Sequential Access Time Register (pHS_ACCTIME)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
DMA sequential access time register (pHS_ACCTIME)	0030119E (HW)	D15-4	--	reserved				--	--	0 when being read.	
		D3	DMAACCTIME3	IDMA and HSDMA sequential access time setup	0	Unlimited	8	512 cycles	0	R/W	
		D2	DMAACCTIME2		1	64 cycles	9	576 cycles	0		
		D1	DMAACCTIME1		2	128 cycles	A	640 cycles	0		
		D0	DMAACCTIME0		3	192 cycles	B	704 cycles	0		
					4	256 cycles	C	768 cycles			
					5	320 cycles	D	832 cycles			
					6	384 cycles	E	896 cycles			
7	448 cycles				F	960 cycles					

D[15:4] Reserved**D[3:0] DMAACCTIME: IDMA and HSDMA Sequential Access Time Setup Bits**

Sets the sequential access time for IDMA and HSDMA.

Table II.1.9.6 Setting the Sequential Access Time

DMAACCTIME3	DMAACCTIME2	DMAACCTIME1	DMAACCTIME0	DMA sequential access time
1	1	1	1	960 cycles
1	1	1	0	896 cycles
1	1	0	1	832 cycles
1	1	0	0	768 cycles
1	0	1	1	704 cycles
1	0	1	0	640 cycles
1	0	0	1	576 cycles
1	0	0	0	512 cycles
0	1	1	1	448 cycles
0	1	1	0	384 cycles
0	1	0	1	320 cycles
0	1	0	0	256 cycles
0	0	1	1	192 cycles
0	0	1	0	128 cycles
0	0	0	1	64 cycles
0	0	0	0	Unlimited

(Default: 0b0000 = Unlimited)

When “Unlimited” is selected, the AHB bus will not be released until a DMA transfer has been completed after it starts. Specifying a number of cycles allows a DMA transfer to be temporarily suspended when the specified cycles of data transfer have been executed to release the bus. The CPU or LCDC can perform a bus access during the suspended status. After that, the DMAC resumes the data transfer that was being suspended.

II.1.10 Precautions

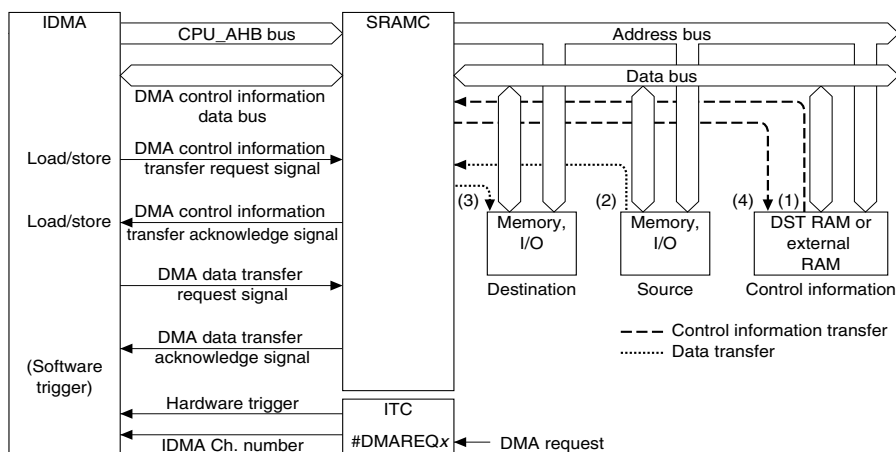
- When setting the transfer conditions, always make sure the DMA controller is inactive (HS_x_EN (D0/0x30112C + 0x10•x) = 0).
 - * **HS_x_EN**: Ch.x Enable Bit in the HSDMA Ch.x Enable Register (D0/0x30112C + 0x10•x)
- After an initial reset, the cause-of-interrupt flag (FHDM_x (Dx/0x300281)) becomes indeterminate. Always be sure to reset the flag to prevent interrupts or IDMA requests from being generated inadvertently.
 - * **FHDM_x**: HSDMA Ch.x Cause-of-Interrupt Flag in the DMA Interrupt Cause Flag Register (Dx/0x300281)
- To prevent an interrupt from being generated repeatedly for the same source, be sure to reset the cause-of-interrupt flag before setting up the PSR again or executing the reti instruction.
- 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.
- In dual-address mode, A0RAM (area 0), Specific ROM (area 1), and IVRAM (area 0) cannot be specified as the source or destination for DMA transfer. While IVRAM (area 3), DST RAM (area 3) and the internal peripheral I/O registers (area 6) can be used for dual-address transfer.
- In single-address mode, A0RAM (area 0), Specific ROM (area 1), area 2, IVRAM (area 0 or area 3), DST RAM (area 3) and the internal peripheral I/O registers (area 6) cannot be used for DMA transfer.
- Single-address mode does not allow data transfer between memory devices. An external logic circuit is required to perform single-address transfer between memory devices.
- Single-address mode does not support the external memory area that is configured for SDRAM.
- Be sure to disable the HSDMA before setting the chip in SLEEP mode (executing the slp instruction). HALT mode can be set even if the HSDMA is enabled.

II.2 Intelligent DMA (IDMA)

II.2.1 Functional Outline of IDMA

The S1C33E08 contains an intelligent DMA (IDMA), a function that allows control information to be programmed in RAM. Up to 128 channels can be programmed, including 41 channels that are invoked by a cause of interrupt 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 a cause of interrupt 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.

Intelligent DMA transfer



- (1) The control information stored in the memory is loaded into the IDMA temporary register.
- (2) Transfer data is read from the source memory or I/O device.
- (3) Transfer data is written to the destination memory or I/O device.
- (4) The updated control information in the IDMA temporary register is written back to the memory.

Figure II.2.1.1 Data and Control Information Flow in Intelligent DMA Transfer

The features of IDMA are outlined below.

- Controller Equivalent to the HSDMA dual-address transfer controller
- Number of channels 128 channels
- Control information Programmable in the RAM
The information table can be stored in DST RAM (area 3) or in the external RAM. (AORAM cannot be used.)
- Source External memory and internal memory except Areas 0 and 1
- Destination External memory and internal memory except Areas 0 and 1
- Transfer data size 8, 16, or 32 bits
- Trigger
 1. Software trigger (register control)
 2. Hardware trigger (causes of interrupts)
- Transfer mode
 1. Single transfer (one unit of data is transferred by one trigger)
 2. Successive transfer (specified number of data are transferred by one trigger)
 3. Block transfer (data block of the specified size is transferred by one trigger)
- Transfer address control The source and/or destination addresses can be incremented or decremented in units of the transfer data size upon completion of transfer. In successive or block transfers, the address can be reset to the initial value upon completion of transfer.
- Programmable link function Any channel can be linked with another to perform data transfer by multiple channels sequentially.

S1C33E08 extended functions

In the S1C33E08 DMA controller, some IDMA functions have been extended from those of the C33 STD. Table II.2.1.1 shows differences between C33 STD IDMA and S1C33E08 IDMA.

Table II.2.1.1 Differences between C33 STD IDMA and S1C33E08 IDMA

Function	C33 STD IDMA	S1C33E08 IDMA
Source/destination address bit width	28 bits	32 bits
Transfer counter (for single/successive transfer)	24 bits	32 bits
Transfer counter (for block transfer)	16 bits	20 bits
Block size setup bit width	8 bits	12 bits
Transfer data size	16 bits, 8 bits	32 bits, 16 bits, 8 bits
Address decrement function with initialization	Unavailable	Available
Control information size per channel	3 words (96 bits)	4 words (128 bits)
Control information base address	28 bits (word alignment)	32 bits (4-word alignment)

Note that the item layout in the control information has been changed along with this functional extension. Furthermore, the control information is no longer placed in the area 0 built-in RAM (AORAM).

II.2.2 Programming Control Information

The intelligent DMA operates according to the control information prepared in RAM. Note that the control information must be placed in DST RAM (area 3) or an external RAM. A0RAM (area 0) cannot be used to store control information.

The control information is 4 words (16 bytes) per channel in size, and must be located at continuous addresses beginning with the base address that is set in the software application as the starting address of channel 0. Consequently, an area of 512 words (2,048 bytes) in RAM is required in order for all of 128 channels to be used. Note that the last 132 bytes in DST RAM (area 3) are reserved for the debug circuits. Therefore, up to 119 channels are available when using the on-chip debug functions.

The following explains how to set the base address and the contents of control information. Before using IDMA, make each the settings described below.

II.2.2.1 Setting the Base Address

Set the starting address of control information (starting address of channel 0) to DBASEL[15:0] (D[15:0]/0x301100) for 16 low-order bits and DBASEH[15:0] (D[15:0]/0x301102) for 16 high-order address bits.

- * **DBASEL[15:0]**: IDMA Low-order Base Address Bits in the IDMA Base Address Register 0 (D[15:0]/0x301100)
- * **DBASEH[15:0]**: IDMA High-order Base Address Bits in the IDMA Base Address Register 1 (D[15:0]/0x301102)

When initially reset, the base address is set to 0x200003A0.

- Notes:**
- The control information must be placed in DST RAM (area 3) or an external RAM. A0RAM (area 0) cannot be used to store control information.
 - The address you set in the IDMA base address register must always be 4-word units boundary address.
 - Be sure to disable DMA transfers (IDMAEN (D0/0x301105) = 0) before setting the base address. Writing to the IDMA base address register is ignored when the DMA transfer is enabled (IDMAEN (D0/0x301105) = 1). When the register is read, the read data is indeterminate.

* **IDMAEN**: IDMA Enable Bit in the IDMA Enable Register (D0/0x301105)

II.2.2.2 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 × 16 [bytes])

Note: The control information must be written only when the channel to be set does not start a DMA transfer. If a DMA transfer starts when the control information is being written to the RAM, proper transfer cannot be performed. Reading the control information can always be done.

II BUS MODULES: INTELLIGENT DMA (IDMA)

The contents of control information (4 words) in each channel are shown in the table below.

Table II.2.2.2.1 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			
	D23–18	reserved	–			
	D17–16	DATSIZ[1:0]	Data size control (Do not set to 11.)			
			DATSIZ1	DATSIZ0	Setting contents	
			1	0	Word (32 bits)	
			0	1	Half-word (16 bits)	
			0	0	Byte (8 bits)	
	D15	reserved	–			
	D14–12	SRINC[2:0]	Source address control (Do not set to others.)			
			SRINC2	SRINC1	SRINC0	Setting contents
			1	0	0	Address decrement with initialization (address is reset in successive or block transfer mode)
			0	1	1	Address increment without initialization (address is not reset)
			0	1	0	Address increment with initialization (address is reset in successive or block transfer mode)
0			0	1	Address decrement without initialization (address is not reset)	
		0	0	Address fixed		
D11	reserved	–				
D10–8	DSINC[2:0]	Destination address control (Do not set to others.)				
		DSINC2	DSINC1	DSINC0	Setting contents	
		1	0	0	Address decrement with initialization (address is reset in successive or block transfer mode)	
		0	1	1	Address increment without initialization (address is not reset)	
		0	1	0	Address increment with initialization (address is reset in successive or block transfer mode)	
		0	0	1	Address decrement without initialization (address is not reset)	
		0	0	Address fixed		
D7–6	reserved	–				
D5–4	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		
D3–1	reserved	–				
D0	DINTEN	End-of-transfer interrupt enable 1 = Enabled, 0 = Disabled				
2nd	D31–12	TC[19:0]	Transfer counter (block transfer mode) Transfer counter - 20 high-order bits (single or successive transfer mode)			
	D11–0	BLKLEN[11:0]	Block size (block transfer mode) Transfer counter - 12 low-order bits (single or successive transfer mode)			
3rd	D31–0	SRADR[31:0]	Source address			
4th	D31–0	DSADR[31:0]	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.

DATSIZ[1:0]: Data size control (D[17:16]/1st word)

Set the unit size of data to be transferred.

Table II.2.2.2.2 Transfer Data Size

DATSIZ1	DATSIZ0	Transfer data size
1	1	Invalid
1	0	Word (32 bits)
0	1	Half-word (16 bits)
0	0	Byte (8 bits)

SRINC[2:0]: Source address control (D[14:12]/1st word)

Set the source address control condition.

- **SRINC[2:0] = 000: Address fixed**

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.

- **SRINC[2:0] = 011: Address increment without initialization** (address is not reset)

The source address is incremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. The address that has been incremented during transfer does not return to the initial value.

- **SRINC[2:0] = 001: Address decrement without initialization** (address is not reset)

The source address is decremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. The address that has been decremented during transfer does not return to the initial value.

- **SRINC[2:0] = 010: Address increment with initialization**

(address is reset in successive or block transfer mode)

The source address is incremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. In single transfer mode, the address that has been incremented during transfer does not return to the initial value. In successive transfer modes, the incremented address returns to the initial value when the specified number of transfers is completed (CNT = 0). In block transfer mode, the incremented address returns to the initial value when the block transfer is completed.

- **SRINC[2:0] = 100: Address decrement with initialization**

(address is reset in successive or block transfer mode)

The source address is decremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. In single transfer mode, the address that has been decremented during transfer does not return to the initial value. In successive transfer modes, the decremented address returns to the initial value when the specified number of transfers is completed (CNT = 0). In block transfer mode, the decremented address returns to the initial value when the block transfer is completed.

- **SRINC[2:0] = Other than above: settings are prohibited**

Note: In single transfer mode, the address does not return to the initial value even if a condition with address initialization is specified.

DSINC[2:0]: Destination address control (D[10:8]/1st word)

Set the destination address control condition.

- **DSINC[2:0] = 000: Address fixed**

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.

- **DSINC[2:0] = 011: Address increment without initialization** (address is not reset)

The destination address is incremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. The address that has been incremented during transfer does not return to the initial value.

II BUS MODULES: INTELLIGENT DMA (IDMA)

- **DSINC[2:0] = 001: Address decrement without initialization** (address is not reset)

The destination address is decremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. The address that has been decremented during transfer does not return to the initial value.

- **DSINC[2:0] = 010: Address increment with initialization**
(address is reset in successive or block transfer mode)

The destination address is incremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. In single transfer mode, the address that has been incremented during transfer does not return to the initial value. In successive transfer modes, the incremented address returns to the initial value when the specified number of transfers is completed (CNT = 0). In block transfer mode, the incremented address returns to the initial value when the block transfer is completed.

- **DSINC[2:0] = 100: Address decrement with initialization**
(address is reset in successive or block transfer mode)

The destination address is decremented by an amount equal to the data size set by DATSIZ when one data transfer is completed. In single transfer mode, the address that has been decremented during transfer does not return to the initial value. In successive transfer modes, the decremented address returns to the initial value when the specified number of transfers is completed (CNT = 0). In block transfer mode, the decremented address returns to the initial value when the block transfer is completed.

- **DSINC[2:0] = Other than above: settings are prohibited**

Note: In single transfer mode, the address does not return to the initial value even if a condition with address initialization is specified.

DMOD[1:0]: Transfer mode (D[5:4]/1st word)

Use these bits to set the desired transfer mode.

The transfer modes are outlined below (to be detailed later):

- **DMOD[1:0] = 00: Single transfer mode**

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.

- **DMOD[1:0] = 01: Successive transfer mode**

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.

- **DMOD[1:0] = 10: 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 need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

- **DMOD[1:0] = 11: Settings are prohibited**

DINTEN: End-of-transfer interrupt enable (D0/1st 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 cause-of-interrupt 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.

TC[19:0]: Transfer counter (D[31:12]/2nd word)

In block transfer mode, a transfer count can be specified using up to 20 bits. Set this value here.

In single transfer and successive transfer modes, a transfer count can be specified using up to 32 bits. Set a 20-bit high-order value here.

BLKLEN[11:0]: Block size/transfer counter (D[11:0]/2nd 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 12-bit low-order value for the transfer count here.

Note: The transfer count and block size 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.

SRADR[31:0]: Source address (D[31:0]/3rd 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.

DSADR[31:0]: Destination address (D[31:0]/4th 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.

Notes:

- Area 0 (A0RAM) and area 2 cannot be used for IDMA transfer and storing control information.
- 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.

II.2.3 IDMA Invocation

The triggers by which IDMA is invoked have the following three causes:

1. Cause of interrupt in internal peripheral circuits (hardware trigger)
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 (D0/0x301105), 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 (D0/0x301105) is set to 0, no IDMA invocation request is accepted.

* **IDMAEN**: IDMA Enable Bit in the IDMA Enable Register (D0/0x301105)

IDMA invocation by a cause of interrupt in internal peripheral circuits

Some internal peripheral circuits that have an interrupt generating function can invoke IDMA by a cause of interrupt in that circuit. The IDMA channel numbers corresponding to such IDMA invocation are predetermined. The relationship between the causes of interrupt that have this function and the IDMA channels is shown in Table II.2.3.1.

Table II.2.3.1 Interrupt Causes Used to Invoke IDMA

Peripheral circuit	Cause of interrupt	IDMA Ch.	IDMA request bit	IDMA enable bit
I/O ports	Port input 0	1	RP0 (D0/0x300290)	DEP0 (D0/0x300294)
	Port input 1	2	RP1 (D1/0x300290)	DEP1 (D1/0x300294)
	Port input 2	3	RP2 (D2/0x300290)	DEP2 (D2/0x300294)
	Port input 3	4	RP3 (D3/0x300290)	DEP3 (D3/0x300294)
High-speed DMA	Ch.0, end of transfer	5	RHDM0 (D4/0x300290)	DEHDM0 (D4/0x300294)
	Ch.1, end of transfer	6	RHDM1 (D5/0x300290)	DEHDM1 (D5/0x300294)
16-bit timers 0–5	Timer 0 comparison B	7	R16TU0 (D6/0x300290)	DE16TU0 (D6/0x300294)
	Timer 0 comparison A	8	R16TC0 (D7/0x300290)	DE16TC0 (D7/0x300294)
	Timer 1 comparison B	9	R16TU1 (D0/0x300291)	DE16TU1 (D0/0x300295)
	Timer 1 comparison A	10	R16TC1 (D1/0x300291)	DE16TC1 (D1/0x300295)
	Timer 2 comparison B	11	R16TU2 (D2/0x300291)	DE16TU2 (D2/0x300295)
	Timer 2 comparison A	12	R16TC2 (D3/0x300291)	DE16TC2 (D3/0x300295)
	Timer 3 comparison B	13	R16TU3 (D4/0x300291)	DE16TU3 (D4/0x300295)
	Timer 3 comparison A	14	R16TC3 (D5/0x300291)	DE16TC3 (D5/0x300295)
	Timer 4 comparison B	15	R16TU4 (D6/0x300291)	DE16TU4 (D6/0x300295)
	Timer 4 comparison A	16	R16TC4 (D7/0x300291)	DE16TC4 (D7/0x300295)
Serial interface Ch.0–Ch.1	Ch.0 receive buffer full	23	RSRX0 (D6/0x300292)	DESRX0 (D6/0x300296)
	Ch.0 transmit buffer empty	24	RSTX0 (D7/0x300292)	DESTX0 (D7/0x300296)
	Ch.1 receive buffer full	25	RSRX1 (D0/0x300293)	DESRX1 (D0/0x300297)
	Ch.1 transmit buffer empty	26	RSTX1 (D1/0x300293)	DESTX1 (D1/0x300297)
	A/D converter	End of A/D conversion	27	RADE (D2/0x300293)
I/O ports	Port input 4	28	RP4 (D4/0x300293)	DEP4 (D4/0x300297)
	Port input 5	29	RP5 (D5/0x300293)	DEP5 (D5/0x300297)
	Port input 6	30	RP6 (D6/0x300293)	DEP6 (D6/0x300297)
	Port input 7	31	RP7 (D7/0x300293)	DEP7 (D7/0x300297)
LCDC	End of frame	33	RLCDC (D1/0x30029B)	DELCDC (D1/0x30029C)
Serial interface Ch.2	Ch.2 receive buffer full	34	RSRX2 (D2/0x30029B)	DESRX2 (D2/0x30029C)
	Ch.2 transmit buffer empty	35	RSTX2 (D3/0x30029B)	DESTX2 (D3/0x30029C)
SPI	Receive DMA request	36	RSPIRX (D4/0x30029B)	DESPIRX (D4/0x30029C)
	Transmit DMA request	37	RSPLITX (D5/0x30029B)	DESPITX (D5/0x30029C)
I/O ports or port MUX interrupt	Port input 8 / SPI	38	RP8 (D0/0x3002AC)	DEP8 (D0/0x3002AE)
	Port input 9 / USB PDREQ	39	RP9 (D1/0x3002AC)	DEP9 (D1/0x3002AE)
	Port input 10 / USB INT	40	RP10 (D2/0x3002AC)	DEP10 (D2/0x3002AE)
	Port input 11 / DCSIO	41	RP11 (D3/0x3002AC)	DEP11 (D3/0x3002AE)
I/O ports	Port input 12	42	RP12 (D4/0x3002AC)	DEP12 (D4/0x3002AE)
	Port input 13	43	RP13 (D5/0x3002AC)	DEP13 (D5/0x3002AE)
	Port input 14	44	RP14 (D6/0x3002AC)	DEP14 (D6/0x3002AE)
	Port input 15	45	RP15 (D7/0x3002AC)	DEP15 (D7/0x3002AE)
I ² S	I ² S	46	RI2S (D0/0x3002AD)	DEI2S (D0/0x3002AF)

These causes of interrupt are used in common for interrupt requests and IDMA invocation requests.

To invoke IDMA upon the occurrence of a cause of interrupt, set the corresponding bits of the IDMA request and IDMA enable registers shown in the table by writing 1. Then when a cause of interrupt occurs, an interrupt request to the CPU is kept pending and the corresponding IDMA channel is invoked.

The cause-of-interrupt 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 cause-of-interrupt flag.

- The transfer counter has reached 0.
- DINTEN in control information is set to 1 (interrupt enabled).

In this case, the IDMA request bit is cleared to 0. Therefore, if IDMA needs to be invoked when a cause of interrupt 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 cause-of-interrupt flag. The IDMA enable bit is not cleared and remains set to 1.

If the transfer counter is not 0, the cause-of-interrupt 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 cause-of-interrupt 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 cause of interrupt generates an interrupt request and not an IDMA request.

The control registers (interrupt enable register and interrupt priority register) corresponding to the cause of interrupt 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 causes of interrupt described above, can be invoked by a trigger in the software application.

When the IDMA channel number to be invoked (0 to 127) is written to DCHN[6:0] (D[6:0]/0x301104) and DSTART (D7/0x301104) is set to 1 after setting IDMAEN (D0/0x301105) to 1, the specified IDMA channel starts a DMA transfer.

- * **DCHN[6:0]**: IDMA Channel Number Set-up Bits in the IDMA Start Register (D[6:0]/0x301104)
- * **DSTART**: IDMA Start Control Bit in the IDMA Start Register (D7/0x301104)

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), a cause of interrupt 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 cause of interrupt 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 (D0/0x301105) is 0 (DMA transfer disabled) is kept pending until IDMAEN (D0/0x301105) 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 executed after the DMA transfer by the software trigger is completed.

II.2.4 Operation of IDMA

IDMA has three transfer modes, in each of which data transfer operates differently. Furthermore, a cause of interrupt is processed differently depending on the type of trigger. IDMA supports only dual-address transfers. It does not support single-address transfers. The following describes the operation of IDMA in each transfer mode and how a cause of interrupt is processed for each type of trigger.

II.2.4.1 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 II.2.4.1.1.

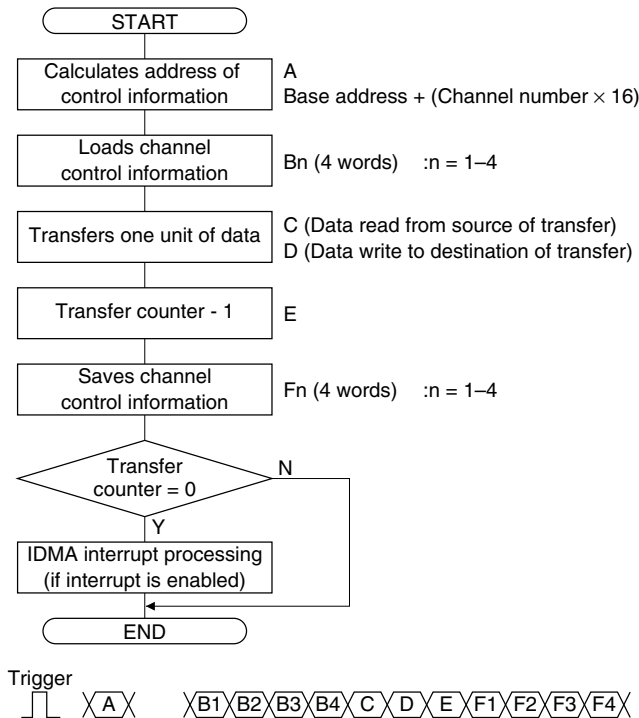


Figure II.2.4.1.1 Operation Flow in Single Transfer Mode

- (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	Cause-of-interrupt flag	IDMA request bit	IDMA enable bit
Transfer counter \neq 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)

II.2.4.2 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 II.2.4.2.1.

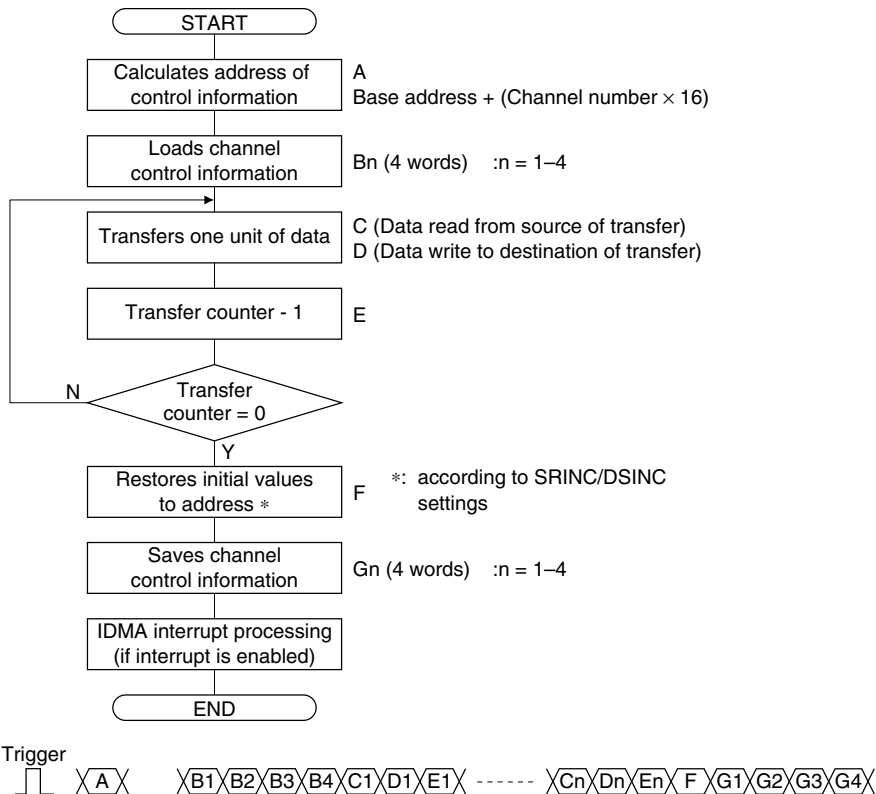


Figure II.2.4.2.1 Operation Flow in Successive Transfer Mode

- (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) If SRINC and/or DSINC are 010 or 100, the address is recycled to the initial value.
- (8) The modified control information is written to RAM.
- (9) In the case of a hardware trigger, the interrupt control bits are processed before completing IDMA.

Condition	Cause-of-interrupt 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)

II.2.4.3 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 II.2.4.3.1.

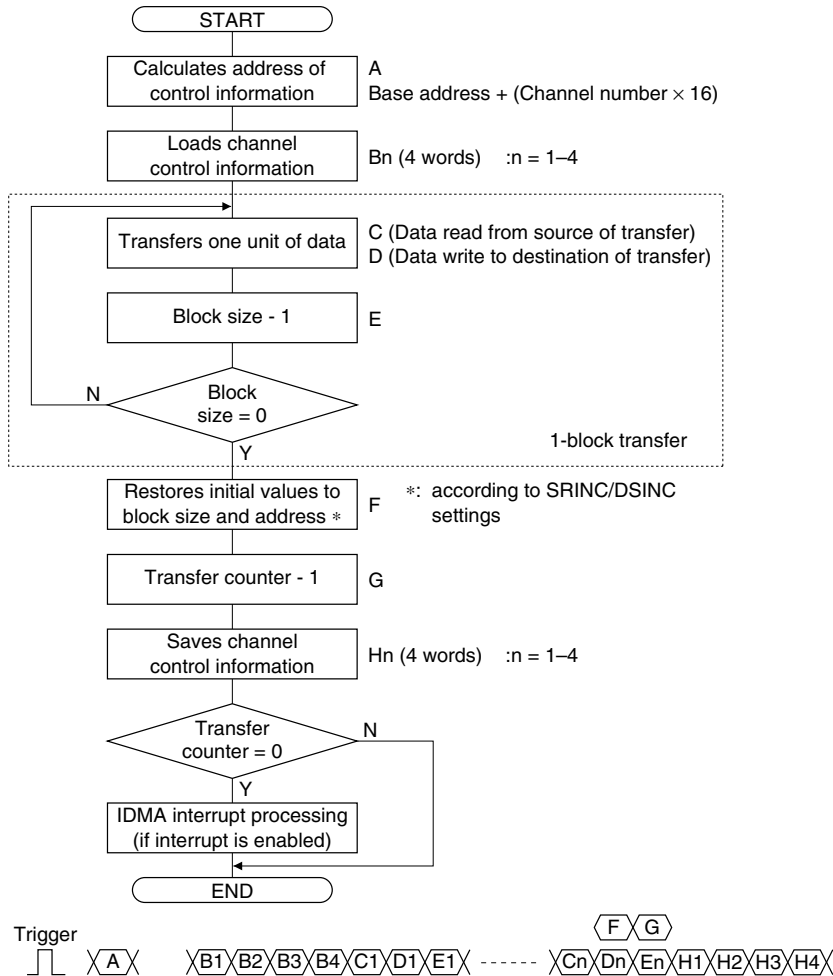


Figure II.2.4.3.1 Operation Flow in Block Transfer Mode

- (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/or DSINC are 010 or 100, 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	Cause-of-interrupt 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)

II.2.4.4 Cause-of-Interrupt Processing by Trigger Type

When invoked by a cause of interrupt

The cause-of-interrupt 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 cause of interrupt that has invoked IDMA is not reset and an interrupt request is generated. At the same time, the IDMA request bit is cleared to 0. The IDMA enable bit is not cleared and remains set to 1.

If the transfer counter is not 0, the cause-of-interrupt 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 cause-of-interrupt 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.

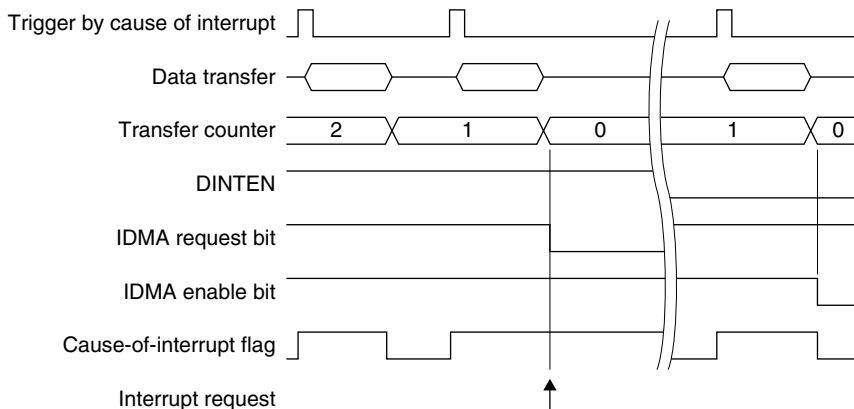


Figure II.2.4.4.1 Operation when Invoked by Cause of Interrupt

When IDMA is invoked by a cause of interrupt, the IDMA cause-of-interrupt flag FIDMA (D4/0x300281) will not be set.

* **FIDMA**: IDMA Cause-of-Interrupt Flag in the DMA Interrupt Cause Flag Register (D4/0x300281)

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, FIDMA (D4/0x300281) is set, thereby generating an interrupt request.

If the transfer counter is not 0 or DINTEN = 0 (interrupt disabled), FIDMA (D4/0x300281) is not set.

If the cause-of-interrupt flag for the same channel is set during a software-triggered transfer, the IDMA invocation request by that cause-of-interrupt flag is kept pending. However, the cause-of-interrupt flag will be reset when the current execution is completed, so there will be no DMA transfer by the cause-of-interrupt flag.

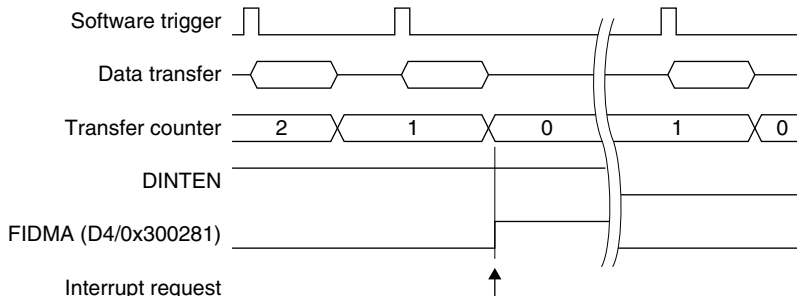


Figure II.2.4.4.2 Operation when Invoked by Software Trigger

II.2.5 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 II.2.5.1.

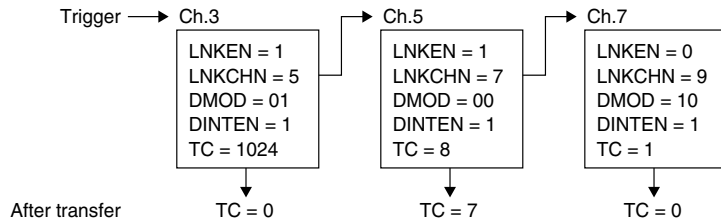


Figure II.2.5.1 Example of Link Setting

For the above example, IDMA operates as described below.

For trigger in hardware

- (1) The IDMA channel 3 is invoked by a cause of interrupt 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 cause-of-interrupt 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 cause-of-interrupt 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 cause-of-interrupt 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 software trigger DSTART (D7/0x301104) 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 cause-of-interrupt flag FIDMA (D4/0x300281) is set when the transfer is completed.

- * **DSTART**: IDMA Start Control Bit in the IDMA Start Register (D7/0x301104)
- * **FIDMA**: IDMA Cause-of-Interrupt Flag in the DMA Interrupt Cause Flag Register (D4/0x300281)

- (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 FIDMA (D4/0x300281) to be set to 1. However, FIDMA (D4/0x300281) has already been set when the transfer is completed in (1) above.
- (4) Since FIDMA (D4/0x300281) 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.

II.2.6 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 a cause of interrupt

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 cause of interrupt has occurred and the IDMA channel assigned to that cause of interrupt is invoked.

If the transfer counter is decremented to 0 and DINTEN = 1 (interrupt enabled) when one DMA transfer is completed, the cause of interrupt that has invoked IDMA is not reset and an interrupt request is generated. At the same time, the IDMA request bit is cleared to 0. The IDMA enable bit is not cleared and remains set to 1.

If the transfer counter is not 0, the cause-of-interrupt 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 cause-of-interrupt 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 a cause of interrupt, the IDMA cause-of-interrupt flag FIDMA (D4/0x300281) will not be set.

For details about the causes of interrupt 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 causes of interrupt are set by the interrupt priority register. Refer to Section III.2, "Interrupt Controller (ITC)." However, when compared between IDMA and interrupt requests, IDMA is given higher priority over the other. Consequently, even when a cause of interrupt occurring during an IDMA transfer has higher priority than the cause of interrupt 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, FIDMA (D4/0x300281) is set, thereby generating an interrupt request. If the transfer counter is not 0 or DINTEN = 0 (interrupt disabled), FIDMA (D4/0x300281) is not set.

IDMA interrupt control register in the interrupt controller

The following control bits are used to control an interrupt for the completion of IDMA transfer:

- * **FIDMA**: IDMA Cause-of-Interrupt Flag in the DMA Interrupt Cause Flag Register (D4/0x300281)
- * **EIDMA**: IDMA Interrupt Enable Bit in the DMA Interrupt Enable Register (D4/0x300271)
- * **PDM[2:0]**: IDMA Interrupt Level Bits in the IDMA Interrupt Priority Register (D[2:0]/0x300265)

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 cause-of-interrupt 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 Section III.2, "Interrupt Controller (ITC)."

Trap vector

The trap vector address for an interrupt upon completion of IDMA transfer by default is set to 0xC00068.

The trap table base address can be changed using the TTBR registers.

II.2.7 Details of Control Registers

Table II.2.7.1 List of IDMA Registers

Address	Register name	Size	Function
0x00301100	IDMA Base Address Register 0 (pIDMABASE)	16	Sets 16 low-order bits of IDMA base address.
0x00301102	IDMA Base Address Register 1	16	Sets 16 high-order bits of IDMA base address.
0x00301104	IDMA Start Register (pIDMA_START)	8	Invokes an IDMA channel.
0x00301105	IDMA Enable Register (pIDMA_EN)	8	Enables IDMA.

The following describes each IDMA control register.

The IDMA control registers are mapped in the 16-bit device area from 0x301100 to 0x301105, and can be accessed in units of half-words or bytes.

Note: When setting the IDMA control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x301100: IDMA Base Address Register 0 (pIDMABASE)
0x301102: IDMA Base Address Register 1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IDMA base address register 0 (pIDMABASE)	00301100 (HW)	D15	DBASEL15	IDMA base address		0	R/W	
		D14	DBASEL14	low-order 16 bits				
		D13	DBASEL13	(Initial value: 0x200003A0)				
		D12	DBASEL12					
		D11	DBASEL11					
		D10	DBASEL10					
		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				
								Fix at 0.
IDMA base address register 1	00301102 (HW)	D15	DBASEH15	IDMA base address		0	R/W	
		D14	DBASEH14	high-order 16 bits				
		D13	DBASEH13	(Initial value: 0x200003A0)				
		D12	DBASEH12					
		D11	DBASEH11					
		D10	DBASEH10					
		D9	DBASEH9	0				
		D8	DBASEH8	0				
		D7	DBASEH7	0				
		D6	DBASEH6	0				
		D5	DBASEH5	0				
		D4	DBASEH4	0				
		D3	DBASEH3	0				
		D2	DBASEH2	0				
		D1	DBASEH1	0				
		D0	DBASEH0	0				

Specify the starting address of the control information to be placed in RAM.

At initial reset, the base address is set to 0x200003A0.

D[15:0]/0x301100 DBASEL[15:0]: IDMA Low-order Base Address Bits

Use DBASEL to set the 16 low-order bits of the base address.

D[15:0]/0x301102 DBASEH[15:0]: IDMA High-order Base Address Bits

Use DBASEH to set the 16 high-order bits of the base address.

In the S1C33E08 IDMA, the DBASEH[15:12] bits have been added to extend the base address into 32 bits.

- Notes:**
- The control information must be placed in DST RAM (area 3) or an external RAM. A0RAM (area 0) cannot be used to store control information.
 - The address you set in the IDMA base address registers must always be 4-word units boundary address.
 - These registers cannot be read or written in bytes. The registers must be accessed in words for read/write operations to address 0x301100, or in half-words for read/write operations to addresses 0x301100 and 0x301102. Write operations in half-words must be performed in order of 0x301100 and 0x301102. Read operations in half-words may be performed in any order.
 - Be sure to disable DMA transfers (IDMAEN (D0/0x301105) = 0) before setting the base address. Writing to the IDMA base address register is ignored when the DMA transfer is enabled (IDMAEN (D0/0x301105) = 1). When the register is read, the read data is indeterminate.

0x301104: IDMA Start Register (pIDMA_START)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IDMA start register (pIDMA_START)	00301104 (B)	D7	DSTART	IDMA start	1 IDMA start 0 Stop	0	R/W	
		D6	DCHN6	IDMA channel number	0 to 127	0	R/W	
		D5	DCHN5					
		D4	DCHN4					
		D3	DCHN3					
		D2	DCHN2					
		D1	DCHN1					
		D0	DCHN0					

D7 DSTART: IDMA Start Control Bit

Use this bit for software trigger and for monitoring the operation of IDMA.

1 (W): Start IDMA

0 (W): Has no effect

1 (R): Operating (only when invoked by software trigger)

0 (R): Idle (default)

When DSTART is set to 1, it functions as a software trigger, invoking the IDMA channel that is set in the DCHN register.

D[6:0] DCHN[6:0]: IDMA Channel Number Setting Bits

Set the channel numbers (0 to 127) to be invoked by software trigger. (Default: 0)

Note: Do not start an IDMA transfer and change the IDMA channel number simultaneously. When setting DCHN[6:0], write 0 to DSTART.

0x301105: IDMA Enable Register (pIDMA_EN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
IDMA enable register (pIDMA_EN)	00301105 (B)	D7-1	–	reserved	–			–	–	0 when being read.
		D0	IDMAEN	IDMA enable (for software trigger)	1	Enabled	0	Disabled	0	R/W

D[7:1] Reserved**D0 IDMAEN: IDMA Enable Bit**

Enable a IDMA transfer.

1 (R/W): Enable

0 (R/W): Disable (default)

IDMA transfer is enabled by writing 1 to this bit and is disabled by writing 0.

II.2.8 Precautions

- The control information must be placed in DST RAM (area 3) or an external RAM. Area 0 (AORAM) and area 2 cannot be used for IDMA transfer and storing control information.
- The address you set in the IDMA base address registers must always be 4-word units boundary address.
- Be sure to disable DMA transfers (IDMAEN (D0/0x301105) = 0) before setting the base address. Writing to the IDMA base address register is ignored when the DMA transfer is enabled (IDMAEN (D0/0x301105) = 1). When the register is read, the read data is indeterminate.
 - * **IDMAEN**: IDMA Enable Bit in the IDMA Enable Register (D0/0x301105)
- Do not start an IDMA transfer and change the IDMA channel number simultaneously. When setting DCHN[6:0] (D[6:0]/0x301104), write 0 to DSTART (D7/0x301104).
 - * **DCHN[6:0]**: IDMA Channel Number Set-up Bits in the IDMA Start Register (D[6:0]/0x301104)
 - * **DSTART**: IDMA Start Control Bit in the IDMA Start Register (D7/0x301104)
- 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.
- Since the C33 PE Core performs look-ahead operations, do not specify another channel immediately after a software trigger has invoked a channel.
- Be sure to disable the IDMA before setting the chip in SLEEP mode (executing the slp instruction). HALT mode can be set even if the IDMA is enabled.

THIS PAGE IS BLANK.

II.3 SRAM Controller (SRAMC)

II.3.1 Overview of the SRAMC

The SRAM Controller (SRAMC) is a bus module connected to the CPU_AHB bus. The SRAMC manages the external memory space by dividing it into 19 areas. This module controls external bus signals according to bus conditions set for each area as it accesses the connected memory or I/O device.

The SRAMC functions and features are outlined below.

- Supports a 32-bit address bus and data bus.
- Controls external memory space as 19 divided areas (Areas 4 to 22).
- Allows various conditions (e.g., device type, device size, number of wait cycles) to be set for each area.
- Outputs 8 chip-enable signals (#CE4 to #CE11) corresponding to each external area.
- Supports two interface modes: A0 and BSL (with BSL mode for external memory only).
- Allows SRAM, ROM, or flash memory to be connected directly to the external bus.
- Allows wait states to be inserted from the external #WAIT pin (for SRAM type only).
- Little endian

II.3.2 SRAMC Pins

Table II.3.2.1 lists the pins used by the SRAMC.

Table II.3.2.1 SRAMC Pin List

Pin name	I/O	Function
A0/ #BSL	O	Address signal output pin / Low-order byte bus strobe signal output pin
A[24:1]	O	Address signal output pins (external address bus)
D[15:0]	I/O	Data signal input/output pins (external data bus)
#CE11	O	Area 11/12 chip enable signal output pin
#CE10	I/O	Area 10/13/20 chip enable signal output pin / Boot mode select pin
#CE9	O	Area 9/22 chip enable signal output pin
#CE8	O	Area 8/21 chip enable signal output pin
#CE7	O	Area 7/19 chip enable signal output pin
#CE6	O	Area 6/17/18 chip enable signal output pin
#CE5	O	Area 5/15/16 chip enable signal output pin
#CE4	O	Area 4/14 chip enable signal output pin
#RD	O	Read signal output pin
#WRL/ #WR	O	Low-order byte write signal output pin (when accessing A0 interfaced area) / Write signal output pin (when accessing BSL interfaced area)
#WRH/ #BSH	O	High-order byte write signal output pin (when accessing A0 interfaced area) / High-order byte bus strobe signal output pin (when accessing BSL interfaced area)
BCLK	O	Bus clock output pin
#WAIT	I	External wait request input pin
BOOT[1:0]	I	Boot mode select pins (BOOT0 is not available in the QFP24-144pin package model)

- Notes:**
- Some control pins above are shared with general-purpose input/output ports or other peripheral circuit input/output pins, so that functionality in the initial state is set to other than the SRAMC. Before the SRAMC signals assigned to these pins can be used, the functions of these pins must be switched for the SRAMC by setting each corresponding Port Function Select Register.
For details on how to switch over the pin functions, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”
 - The bus control signals can be pulled high or forcibly driven low in software. For details on how to control, see Section III.4, “Misc Registers.”

II.3.3 External Memory Area (Areas 4, 5, 7 to 22)

The SRAMC supports an external memory space, which is divided into 19 areas as shown in Figure II.3.3.1.

			#CE4	#CE5	#CE6	#CE7	#CE8	#CE9	#CE10	#CE11
Area 22	0x83FF FFFF	External Memory								
(*1, *2)	0x8000 0000	64M bytes				(*4)		●		
Area 21	0x43FF FFFF	External Memory								
	0x4000 0000	64M bytes					●			
Area 20	0x23FF FFFF	External Memory								
	0x2000 0000	64M bytes							●	
Area 19	0x13FF FFFF	External Memory								
(*1, *2)	0x1000 0000	64M bytes				●				
Area 18	0x0FFF FFFF	External Memory								
	0x0C00 0000	64M bytes			●					
Area 17	0x0BFF FFFF	External Memory								
	0x0800 0000	64M bytes			●					
Area 16	0x07FF FFFF	External Memory								
	0x0600 0000	32M bytes		●						
Area 15	0x05FF FFFF	External Memory								
	0x0400 0000	32M bytes		●						
Area 14	0x03FF FFFF	External Memory								
(*1)	0x0300 0000	16M bytes	●							
Area 13	0x02FF FFFF	External Memory								
	0x0200 0000	16M bytes							●	
Area 12	0x01FF FFFF	External Memory								
(*1)	0x0180 0000	8M bytes								●
Area 11	0x017F FFFF	External Memory								
(*1)	0x0100 0000	8M bytes								●
Area 10	0x00FF FFFF	External Memory								
	0x00C0 0000	4M bytes							●	
Area 9	0x00BF FFFF	External Memory								
(*1)	0x0080 0000	4M bytes						●		
Area 8	0x007F FFFF	External Memory								
	0x0060 0000	2M bytes					●			
Area 7	0x005F FFFF	External Memory								
(*1, *2)	0x0040 0000	2M bytes				●				
Area 6	0x003F FFFF	(Reserved for internal peripherals)								
	0x0030 0000				(*3)					
Area 5	0x002F FFFF	External Memory								
	0x0020 0000	1M bytes		●						
Area 4	0x001F FFFF	External Memory								
(*1)	0x0010 0000	1M bytes	●							

*1 Usable as memory space for SmartMedia (NAND flash), CompactFlash, or PC Card.

*2 Usable as the SDRAM area.

*3 External memory cannot be accessed.

*4 Area 22 is assigned to #CE9 in default settings. Note that Area 22 will be reassigned to #CE7 when the SDRAMC is enabled.

Figure II.3.3.1 External Memory Space of the S1C33E08

Areas 4, 5, 7 to 22 comprise an external memory area accessible from the SRAMC, to which external memory devices may be connected. The device type and size, and number of wait cycles may be set for each of these areas to be accessed.

II.3.3.1 Chip Enable Signals

The S1C33E08 provides 25 bits of an external address bus, 16 bits of an external data bus, and eight chip-enable pins (#CE4 to #CE11), allowing access to the 512MB address space.

Two or more areas are assigned to each chip-enable signal. Table II.3.3.1.1 shows the relationship between the chip-enable pins and corresponding areas.

Table II.3.3.1.1 Relationship between Chip-Enable Pins and Corresponding Areas

#CE pin	Corresponding area	Usable size of area in continuous address range					
		Area	Size	Area	Size	Area	Size
#CE4	Areas 4, 14	Area 4	1MB	Area 14	16MB	–	–
#CE5	Areas 5, 15, 16	Area 5	1MB	Area 15+16	64MB	–	–
#CE6	Areas 17, 18	Area 17+18	128MB	–	–	–	–
#CE7	Areas 7, 19	Area 7	2MB	Area 19	64MB	–	–
#CE8	Areas 8, 21	Area 8	2MB	Area 21	64MB	–	–
#CE9	Areas 9, 22	Area 9	4MB	Area 22	64MB	–	–
#CE10	Areas 10, 13, 20	Area 10	4MB	Area 13	16MB	Area 20	64MB
#CE11	Areas 11, 12	Area 11+12	16MB	–	–	–	–

The #CE_x signal also becomes active when an address in any corresponding area is accessed.

Area 6 is allocated to the I/O area for S1C33E08 IP and peripheral circuits. Although area 6 is one of external memory areas, external memory cannot be accessed.

II.3.3.2 Area Condition Settings

Bus access conditions can be set by area for each #CE_x signal. Therefore, the same conditions for two or more areas accommodated by the respective #CE_x signals will be set.

This section describes the parameters to be set individually for each area and the relevant control bits.

The SRAMC control registers are initialized by an initial reset. These registers should be set up back again in software to suit the external device configuration or specification as required.

For details of bus cycle operation, see Section II.3.6, “Bus Access Timing Chart.”

Note: The control register and control bit configurations are the same for all #CE4 to #CE11 areas. The control bit names begin with CE4 to CE11 to indicate the relevant areas, which in the description below are commonly represented by CE_x for all areas.

Table II.3.3.2.1 Area Parameter Settings

Setup item	Content	Control bit settings
Device type (#CE4–#CE11)	BSL	CE _x TYPE = 1
	A0	CE _x TYPE = 0 (default)
Device size (#CE4–#CE9, #CE11)	16 bits	CE _x SIZE[1:0] = 01 (default)
	8 bits	CE _x SIZE[1:0] = 10
Static wait cycle (#CE4–#CE11)	Insert 7 wait cycles	CE _x WAIT[2:0] = 111 (default)
	⋮	⋮
	Insert 0 wait cycles	CE _x WAIT[2:0] = 000
#CE setup time (#CE4, #CE11)	No setup time	CE _x STUP = 1
	+1 BCLK	CE _x STUP = 0 (default)
Output disable time (#CE9)	7 cycles	CE9HOLD[2:0] = 111
	⋮	⋮
	0 cycles	CE9HOLD[2:0] = 000 (default)

Endian mode

The S1C33E08 supports little endian mode only.

Device type

The SRAMC incorporates an SRAM-type bus interface, allowing A0 (default) or BSL to be selected as the device type. To use a BSL-type device in the #CE_x area, set CE_xTYPE (D_x - 4/0x30150C) to 1.

* **CE_xTYPE:** #CE_x Device Type Select Bit in the Device Type Setup Register (D_x - 4/0x30150C)

Table II.3.3.2.2 lists the bus control signal pins used in each device type.

Table II.3.3.2.2 Bus Control Signal Pins Used in A0 and BSL Modes

Pin name	A0 (default)	BSL
#CE _x	#CE _x	#CE _x
#RD	#RD	#RD
A0/#BSL	Unused	#BSL
#WRL/#WR	#WRL	#WR
#WRH/#BSH	#WRH	#BSH

Device size

Use CExSIZE[1:0] (0x301508) to select a device size.

- * **CE4SIZE[1:0]:** #CE4 Device Size Select Bits in the Device Size Setup Register (D[1:0]/0x301508)
- * **CE5SIZE[1:0]:** #CE5 Device Size Select Bits in the Device Size Setup Register (D[3:2]/0x301508)
- * **CE6SIZE[1:0]:** #CE6 Device Size Select Bits in the Device Size Setup Register (D[5:4]/0x301508)
- * **CE7SIZE[1:0]:** #CE7 Device Size Select Bits in the Device Size Setup Register (D[7:6]/0x301508)
- * **CE8SIZE[1:0]:** #CE8 Device Size Select Bits in the Device Size Setup Register (D[9:8]/0x301508)
- * **CE9SIZE[1:0]:** #CE9 Device Size Select Bits in the Device Size Setup Register (D[11:10]/0x301508)
- * **CE11SIZE[1:0]:** #CE11 Device Size Select Bits in the Device Size Setup Register (D[15:14]/0x301508)

Table II.3.3.2.3 Selection of Device Sizes

CExSIZE1	CExSIZE0	Device size	Connected data bus
1	1	Reserved	–
1	0	8 bits	D[7:0]
0	1	16 bits	D[15:0]
0	0	Reserved	–

At an initial reset, the device size is initialized to 16 bits.

Note: The device size of the #CE10 area is determined by the contents in address 0xC00000 at system boot. The device size is set to 16 bits when the LSB of the 0xC00000 contents is 0 or 8 bits when it is 1.

Static wait cycle

If the number of static wait cycles is specified, the chip enable and read/write signals are always prolonged for the number of specified cycles when the area is accessed. Set up the wait cycle according to the specifications of the device connected to the area using CExWAIT[2:0] (0x301504).

- * **CE4WAIT[2:0]:** Number of #CE4 Static Wait Cycles Setup Bits in the Wait Control Register (D[2:0]/0x301504)
- * **CE5WAIT[2:0]:** Number of #CE5 Static Wait Cycles Setup Bits in the Wait Control Register (D[6:4]/0x301504)
- * **CE6WAIT[2:0]:** Number of #CE6 Static Wait Cycles Setup Bits in the Wait Control Register (D[10:8]/0x301504)
- * **CE7WAIT[2:0]:** Number of #CE7 Static Wait Cycles Setup Bits in the Wait Control Register (D[14:12]/0x301504)
- * **CE8WAIT[2:0]:** Number of #CE8 Static Wait Cycles Setup Bits in the Wait Control Register (D[18:16]/0x301504)
- * **CE9WAIT[2:0]:** Number of #CE9 Static Wait Cycles Setup Bits in the Wait Control Register (D[22:20]/0x301504)
- * **CE10WAIT[2:0]:** Number of #CE10 Static Wait Cycles Setup Bits in the Wait Control Register (D[26:24]/0x301504)
- * **CE11WAIT[2:0]:** Number of #CE11 Static Wait Cycles Setup Bits in the Wait Control Register (D[30:28]/0x301504)

Table II.3.3.2.4 Setting the Static Wait Cycle

CExWAIT2	CExWAIT1	CExWAIT0	Number of wait cycles
1	1	1	7 cycles
1	1	0	6 cycles
1	0	1	5 cycles
1	0	0	4 cycles
0	1	1	3 cycles
0	1	0	2 cycles
0	0	1	1 cycle
0	0	0	No wait cycle

At initial reset, the static wait conditions for all external areas are set to 7 cycles.

The area to which an SRAM device is connected allows dynamic wait control using the #WAIT pin in addition to the static wait control.

For details of bus cycle operation including wait cycles, see Section II.3.6, “Bus Access Timing Chart.”

#CE4/#CE11 setup time

Normally a #CE signal is asserted one BCLK clock cycle before the read/write signal becomes active. For the #CE4 and #CE11 signals, this setup time can be removed to assert the #CE and read/write signals simultaneously. Set CE_xSTUP (D1, D2/0x301500) to 1 to remove the #CE setup time.

- * **CE4STUP**: #CE4 Setup Time Select Bit in the BCLK and Setup Time Control Register (D1/0x301500)
- * **CE11STUP**: #CE11 Setup Time Select Bit in the BCLK and Setup Time Control Register (D2/0x301500)

At initial reset, #CE4 and #CE11 signals are configured with one BCLK setup time.

For the bus cycle operations with or without a setup time, see Section II.3.6, “Bus Access Timing Chart.”

#CE9 output disable time

In cases when a device having a long output disable time is connected, if a read cycle for that device is followed in 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 time is provided to prevent such a data bus contention. This is accomplished by inserting a specified number of output disable cycles between a read cycle and the next bus operation. However, this setting is effective only for the #CE9 area. The output disable time affects bus control signals such as #RD and #WRL/#WRH.

Check the specifications of the device to be connected before setting the output disable time.

Use CE9HOLD[2:0] (D[6:4]/0x301500) to set the #CE9 output disable time.

- * **CE9HOLD[2:0]**: #CE9 Output Disable Time Setup Bits in the BCLK and Setup Time Control Register (D[6:4]/0x301500)

Table II.3.3.2.5 Setting the #CE9 Output Disable Time

CE9HOLD2	CE9HOLD1	CE9HOLD0	Output disable cycles
1	1	1	7 cycles
1	1	0	6 cycles
1	0	1	5 cycles
1	0	0	4 cycles
0	1	1	3 cycles
0	1	0	2 cycles
0	0	1	1 cycle
0	0	0	None

At initial reset, the disable delay time is initialized to “None” (0 cycles).

The following shows the conditions under which the output disable cycle is inserted.

- The output disable cycle is always inserted during read access.
- For read access where data size > device size, the output disable cycle is only inserted during the last access.
- No output disable cycle is inserted during write access.
- No output disable cycle is inserted during consecutive accesses to the same area.

II.3.4 Connection of External Devices and Bus Operation

II.3.4.1 Connecting External Devices

The following shows an example of connecting the S1C33E08 and SRAM.

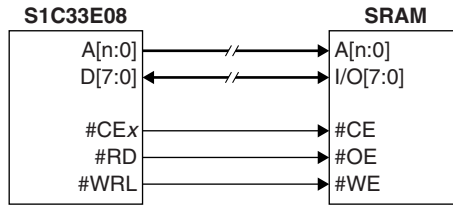


Figure II.3.4.1.1 Example of 8-bit SRAM Connection with 8-bit Device Size

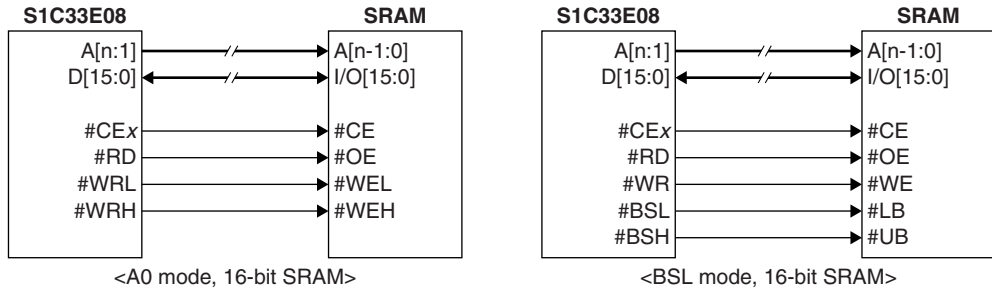


Figure II.3.4.1.2 Example of 16-bit SRAM Connection with 16-bit Device Size

II.3.4.2 Data Configuration in Memory

The S1C33E08 SRAMC handles byte (8-bit), halfword (16-bit), and word (internal 32-bit) data. To access data in memory, addresses aligned to the boundary of the data size must be specified. Specifying other addresses generates address misaligned exceptions.

Instructions (e.g., stack manipulating and branch instructions) that rewrite the content of the Stack Pointer (SP) or Program Counter (PC) forcibly alter the address specified to a boundary address to prevent address misaligned exceptions. For details of address misaligned exceptions, refer to the C33 PE Core Manual.

Table II.3.4.2.1 shows where each type of data is located in memory.

Table II.3.4.2.1 Data Locations in Memory

Data type	Location
Byte	Byte boundary (all addresses)
Halfword	Halfword boundary (A[0] = 0)
Word	Word boundary (A[1:0] = 0b00)

All halfword and word data in memory are accessed in little endian mode. To increase memory efficiency, try locating the same type of data at contiguous addresses to reduce blank areas created by positioning at boundary addresses as much as possible.

II.3.4.3 External Bus Operation

The internal data bus size in the S1C33E08 is 32 bits. Note, however, that it has 16 external bus pins D[15:0]. Depending on the device size and data size of the instruction executed, two or more bus operations may occur. Table II.3.4.3.1 shows bus operation in A0 and BSL modes.

For details on how to connect memory, see Section II.3.4.1, “Connecting External Devices.”

Table II.3.4.3.1 Bus Operation

Device size	Data size	R/W	A1	A0	A0 mode			BSL mode			Access count
					Valid signal	D[15:8] pins	D[7:0] pins	Valid signal	D[15:8] pins	D[7:0] pins	
8 bits	Byte	W	*	*	#WRL	–	D[7:0]	–	–	–	1
		R	*	*	#RD	–	D[7:0]	–	–	–	1
	Half word	W	*	0	#WRL	–	D[7:0]	–	–	–	1st
			*	1		–	D[15:8]	–	–	–	2nd
		R	*	0	#RD	–	D[7:0]	–	–	–	1st
			*	1		–	D[15:8]	–	–	–	2nd
	Word	W	0	0	#WRL	–	D[7:0]	–	–	–	1st
			0	1		–	D[15:8]	–	–	–	2nd
			1	0		–	D[23:16]	–	–	–	3rd
			1	1		–	D[31:24]	–	–	–	4th
		R	0	0	#RD	–	D[7:0]	–	–	–	1st
			0	1		–	D[15:8]	–	–	–	2nd
			1	0		–	D[23:16]	–	–	–	3rd
			1	1		–	D[31:24]	–	–	–	4th
16 bits	Byte	W	*	0	#WRL	–	D[7:0]	#WR #BSL	–	D[7:0]	1
			*	1	#WRH	D[7:0]	–	#WR #BSH	D[7:0]	–	1
		R	*	0	#RD	–	D[7:0]	#RD #BSL	–	D[7:0]	1
			*	1		D[7:0]	–	#RD #BSH	D[7:0]	–	1
	Half word	W	*	0	#WRH #WRL	D[15:0]		#WR #BSH #BSL	D[15:0]		1
		R	*	0	#RD	D[15:0]		#RD #BSH #BSL	D[15:0]		1
	Word	W	0	0	#WRH #WRL	D[15:0]		#WR #BSH #BSL	D[15:0]		1st
			1	0		D[31:16]			D[31:16]		2nd
		R	0	0	#RD	D[15:0]		#RD #BSH #BSL	D[15:0]		1st
			1	0		D[31:16]			D[31:16]		2nd

II.3.5 SRAMC Operating Clock and Bus Clock

II.3.5.1 Operating Clock of the SRAMC

The SRAMC is clocked by the SRAMC_CLK and SRAMC_SAPB_CLK clocks (= MCLK) generated by the CMU.

The bus control signals are generated synchronously with SRAMC_CLK.

The SRAMC_SAPB_CLK is used for the SRAMC control registers.

For details on how to set and control the SRAMC operating clocks, see Section III.1, “Clock Management Unit (CMU).”

Controlling supply of the SRAMC operating clock

The SRAMC operating clocks are supplied to the SRAMC with default settings. Each clock supply can be controlled in the CMU. Use the respective control bits to turn off any unnecessary clock supplies to reduce the amount of power consumed on the chip.

1. SRAMC_SAPB_CLK

The SRAMC_SAPB_CLK is used to operate the SRAMC control registers. To setup the registers, this clock is required. After the registers are set up, the clock supply can be stopped to reduce power consumption by setting SRAMSAPB_CKE (D7/0x301B04) to 0.

- * **SRAMSAPB_CKE**: SRAMC SAPB I/F Clock Control Bit in the Gated Clock Control Register 1 (D7/0x301B04)

2. SRAMC_CLK

The SRAMC_CLK is used for the SRAM interface. To access the external memories/devices and the peripheral control registers in area 6, this clock is required. So this clock cannot be stopped in normal operation mode. However, the clock supply can be stopped in HALT mode. By setting SRAMC_HCKE (D26/0x301B04) to 0, the SRAMC_CLK stops when the CPU enters HALT mode and it resumes when the CPU exits HALT mode.

- * **SRAMC_HCKE**: SRAMC Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D26/0x301B04)

Clock state in standby mode

The supply of the SRAMC operating clock stops depending on the type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

It can be stopped by setting the CMU register.

SLEEP mode: The clock supply stops.

Therefore, the SRAMC also stops operating in SLEEP mode.

II.3.5.2 Generation of the Bus Clock

The SRAMC divides SRAMC_CLK by a specified number to generate the bus clock (BCLK). This divide-by ratio is set using the control bits shown below.

Setting the bus clock for areas other than #CE9

- * **BCLK**: BCLK Divide Control Bit in the BCLK and Setup Time Control Register (D0/0x301500)

Setting the bus clock for the #CE9 area

- * **CE9BCLK**: #CE9 Area BCLK Divide Control Bit in the BCLK and Setup Time Control Register (D7/0x301500)

Table II.3.5.2.1 BCLK (SRAMC_CLK Divide-by Ratio) Settings

BCLK/CE9BCLK	BCLK frequency
1	SRAMC_CLK × 1/2
0	SRAMC_CLK × 1

When initially reset, the BCLK clock is set to SRAMC_CLK × 1/2.

II.3.5.3 External Output of the Bus Clock

The BCLK output is an extended port function. Therefore, before BCLK can be output to external devices, the pin function must be switched for BCLK output by using the Function Select Register for the corresponding port. For details of the pins assigned to the BCLK output function and how to switch the pin functions, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

II.3.6 Bus Access Timing Chart

II.3.6.1 SRAM Read/Write Timings with No External #WAIT

1. SRAM read/write timings with no static wait cycles

[Example settings]

Device size: 16 bits

Number of static wait cycles: 0 cycles

#CE4/#CE11 setup time: no setup time

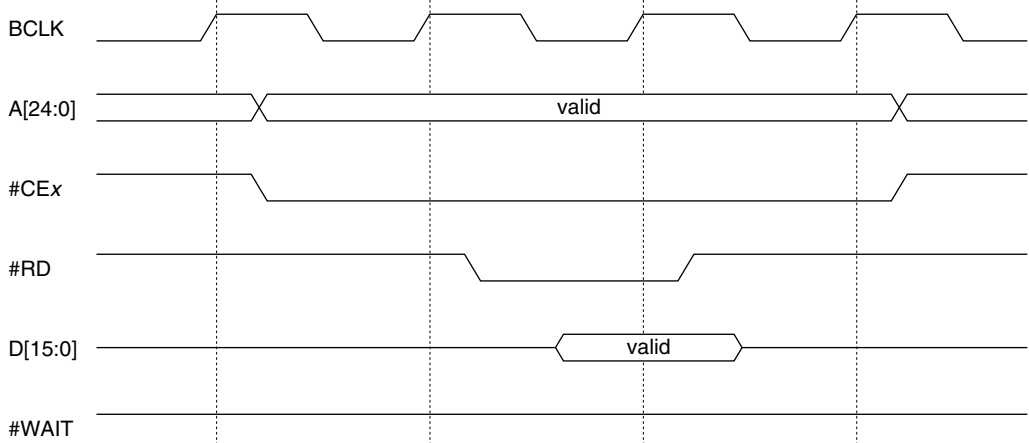


Figure II.3.6.1.1 SRAM Read Timing with No Static Wait Cycle

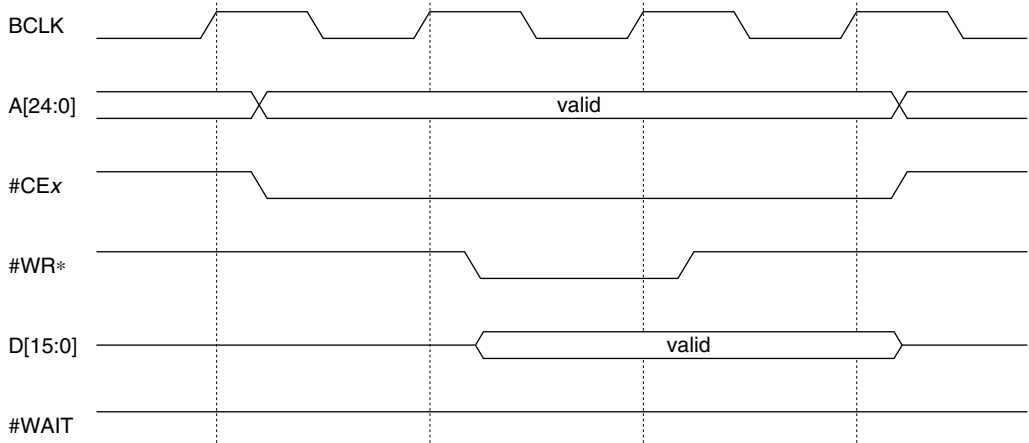


Figure II.3.6.1.2 SRAM Write Timing with No Static Wait Cycle

2. SRAM read/write timings with static wait cycles

[Example settings]

Device size: 16 bits

Number of static wait cycles: 2 cycles

#CE4/#CE11 setup time: no setup time

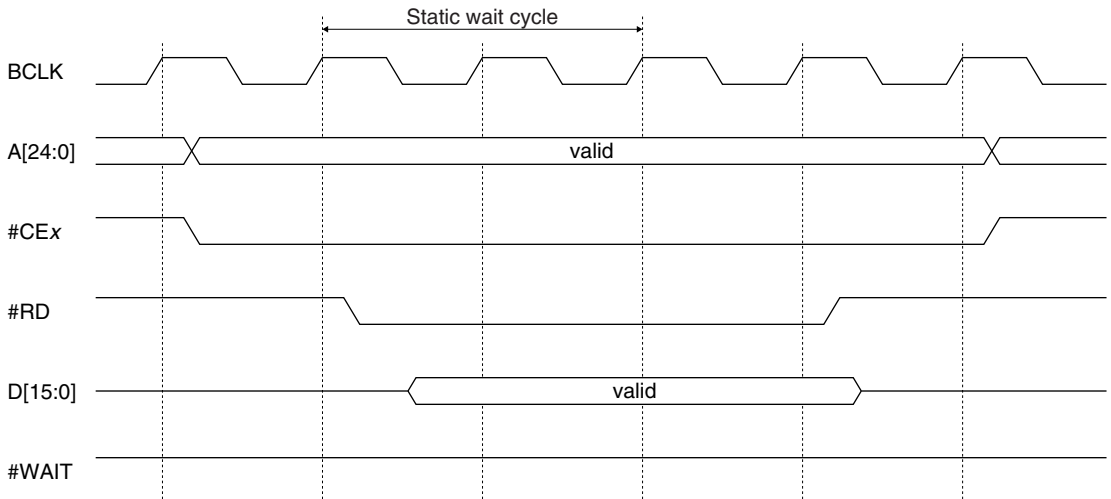


Figure II.3.6.1.3 SRAM Read Timing with Static Wait Cycle

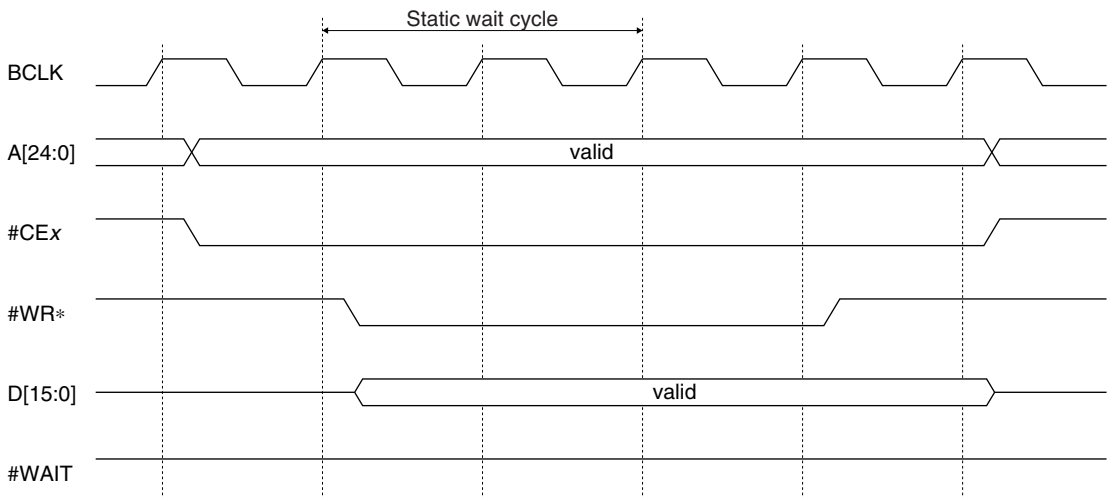


Figure II.3.6.1.4 SRAM Write Timing with Static Wait Cycle

II.3.6.2 SRAM Read/Write Timings with External #WAIT

A wait cycle can be inserted from the #WAIT pin only for SRAM-type devices.

The external #WAIT signal is sampled on the rising edges of BCLK at one clock before the read or write signal goes high. A wait state is entered while the #WAIT signal is sampled active (low), and subsequent operation resumes when the #WAIT signal is sampled inactive (high).

[Example settings]

- Device size: 16 bits
- Number of static wait cycles: 0 cycles
- #CE4/#CE11 setup time: no setup time

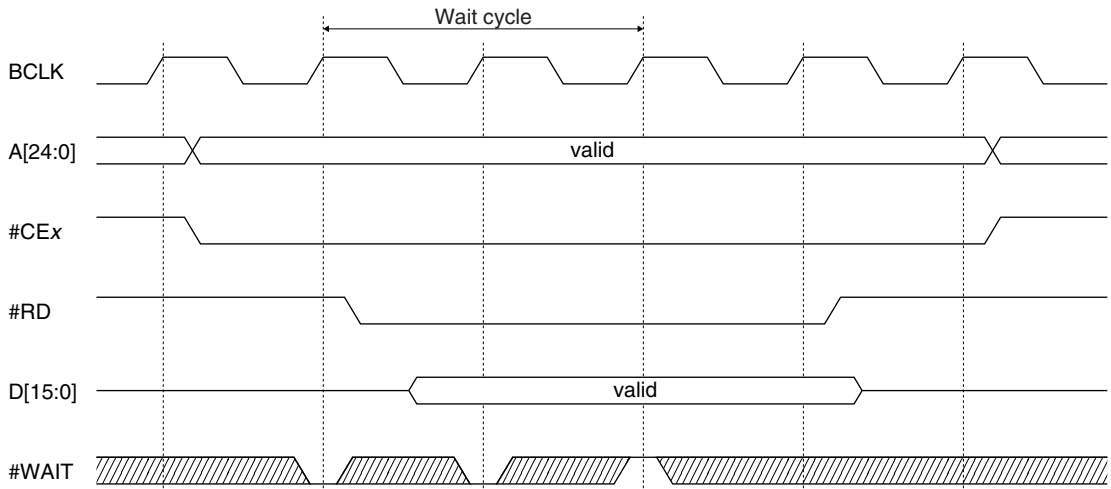


Figure II.3.6.2.1 SRAM Read Timing with External #WAIT

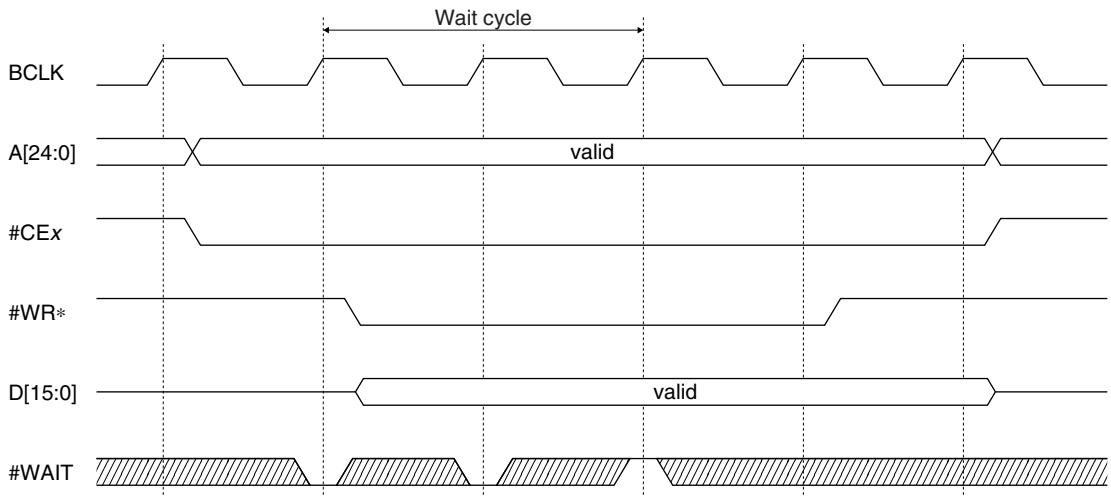


Figure II.3.6.2.2 SRAM Write Timing with External #WAIT

II.3.6.3 SRAM Read/Write Timings with #CE4/#CE11 Setup Time

[Example settings]

Device size: 16 bits

Number of static wait cycles: 0 cycles

#CE4/#CE11 setup time: +1 BCLK

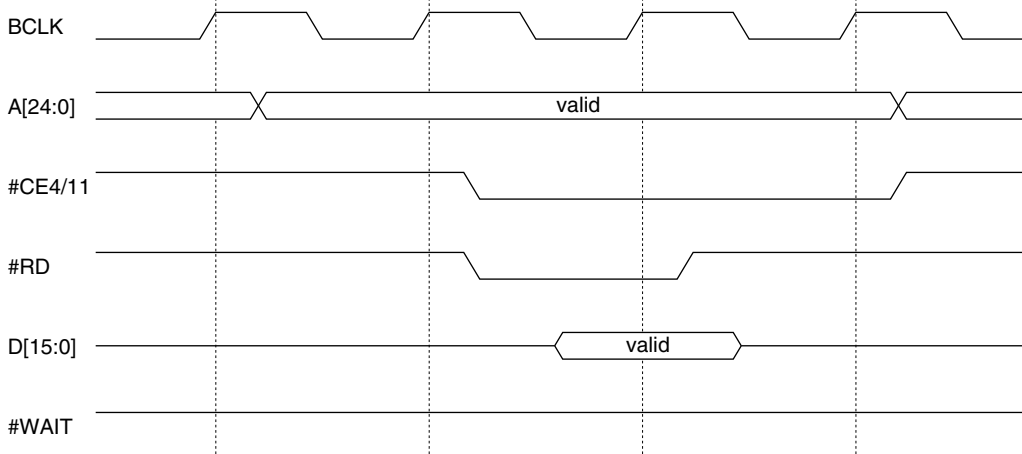


Figure II.3.6.3.1 SRAM Read Timing with #CE4/#CE11 Setup Time

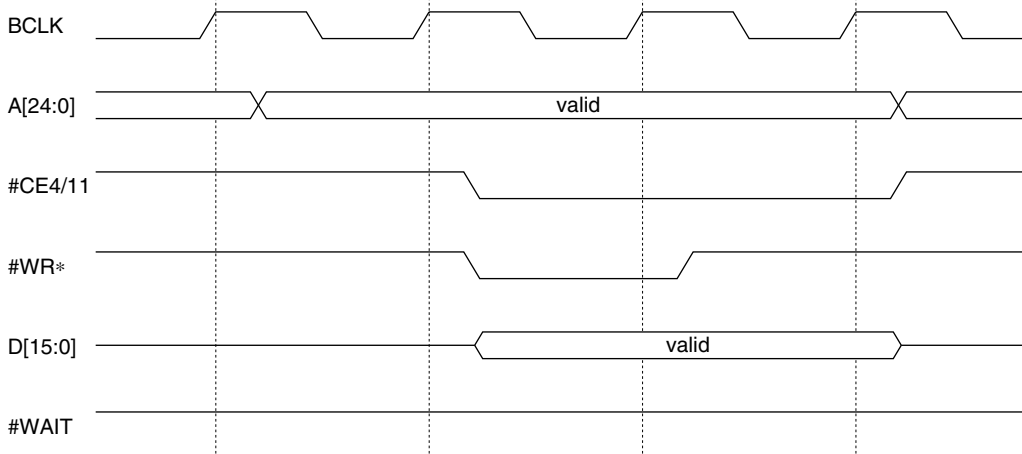


Figure II.3.6.3.2 SRAM Write Timing with #CE4/#CE11 Setup Time

II.3.6.4 SRAM Read Timings with #CE9 Output Disable Time

[Example settings]

Device size: 16 bits
 Number of static wait cycles: 0 cycles
 #CE9 output disable time: None

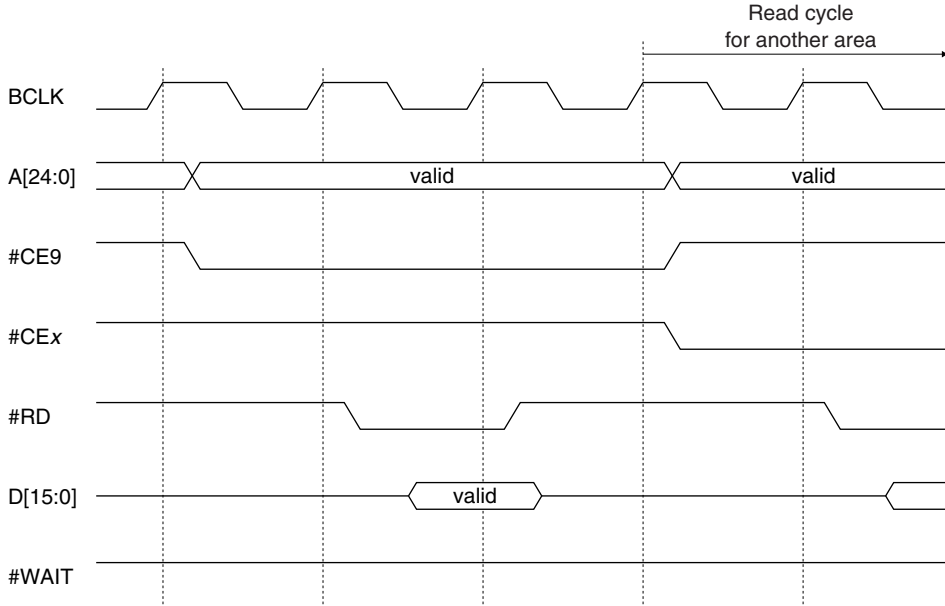


Figure II.3.6.4.1 SRAM Read Timing with No #CE9 Output Disable Time

[Example settings]

Device size: 16 bits
 Number of static wait cycles: 0 cycles
 #CE9 output disable time: 1 cycle

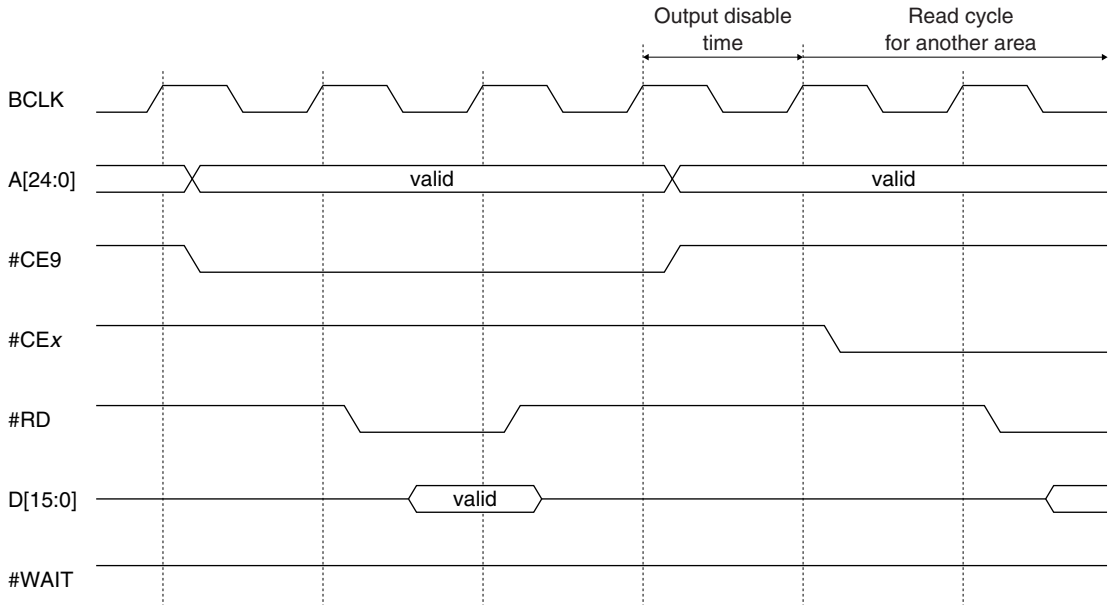


Figure II.3.6.4.2 SRAM Read Timing with #CE9 Output Disable Time

II.3.7 Control Register Details

Table II.3.7.1 SRAMC Register List

Address	Register name	Size	Function
0x00301500	BCLK and Setup Time Control Register (pSRAMC_BCLK_SETUP)	32	Sets BCLK and #CE4/#CE11 setup time.
0x00301504	Wait Control Register (pSRAMC_SWAIT)	32	Sets static wait cycle for each area.
0x00301508	Device Size Setup Register (pSRAMC_SLV_SIZE)	32	Sets device size for each area.
0x0030150C	Device Type Setup Register (pSRAMC_A0_BSL)	32	Sets device type for each area.
0x00301510	Area Location Setup Register (pSRAMC_ALS)	32	Sets area 6 location.

Each SRAMC control register is described below.

The SRAMC control registers are mapped to the 32-bit device area at addresses 0x301500 to 0x301510, and can be accessed in units of words, halfwords, or bytes.

Note: When setting the SRAMC control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x301500: BCLK and Setup Time Control Register (pSRAMC_BCLK_SETUP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
BCLK and setup time control register (pSRAMC_BCLK_SETUP)	00301500 (W)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	CE9BCLK	#CE9 area BCLK divide control	1 SRAMC_CLK × 1/2 0 SRAMC_CLK × 1	1	R/W		
		D6	CE9HOLD2	#CE9 area output disable time	0 to 7	0	0	R/W	
		D5	CE9HOLD1					0	
		D4	CE9HOLD0					0	
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	CE11STUP	#CE11 setup time	1 No setup time 0 +1 BCLK	0	R/W		
		D1	CE4STUP	#CE4 setup time	1 No setup time 0 +1 BCLK	0	R/W		
		D0	BCLK	BCLK divide control	1 SRAMC_CLK × 1/2 0 SRAMC_CLK × 1	1	R/W		

D[31:8] Reserved**D7 CE9BCLK: #CE9 Area BCLK Divide Control Bit**

The BCLK clock for the #CE9 area is independent of other areas and is generated from the SRAMC_CLK clock by being divided by 1 or 2. CE9BCLK is used to select this divide-by ratio.

1 (R/W): SRAMC_CLK × 1/2 (default)

0 (R/W): SRAMC_CLK × 1

D[6:4] CE9HOLD[2:0]: #CE9 Area Output Disable Time Setup Bits

These bits select the output disable time for accessing the #CE9 area.

Table II.3.7.2 Setting the #CE9 Output Disable Time

CE9HOLD2	CE9HOLD1	CE9HOLD0	Output disable cycles
1	1	1	7 cycles
1	1	0	6 cycles
1	0	1	5 cycles
1	0	0	4 cycles
0	1	1	3 cycles
0	1	0	2 cycles
0	0	1	1 cycle
0	0	0	None

(Default: 0b000 = None)

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.

D3 Reserved**D2 CE11STUP: #CE11 Setup Time Select Bit**

This bit selects the setup time (#CE active to #RD/#WR* active) for the #CE11 signal.

1 (R/W): No setup time

0 (R/W): +1 BCLK (default)

D1 CE4STUP: #CE4 Setup Time Select Bit

This bit selects the setup time (#CE active to #RD/#WR* active) for the #CE4 signal.

1 (R/W): No setup time

0 (R/W): +1 BCLK (default)

D0 BCLK: BCLK Divide Control Bit

The BCLK clock is used for SRAM areas except the #CE9 area and is generated from the SRAMC_CLK clock by being divided by 1 or 2. BCLK is used to select this divide-by ratio.

Note that the BCLK pin output clock will not be divided; it is always the same as the SRAMC_CLK clock.

1 (R/W): SRAMC_CLK × 1/2 (default)

0 (R/W): SRAMC_CLK × 1

Use CE9BCLK (D7) to set BCLK for the #CE9 area.

0x301504: Wait Control Register (pSRAMC_SWAIT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Wait control register (pSRAMC_SWAIT)	00301504 (W)	D31	–	reserved		–	–	0 when being read.
		D30	CE11WAIT2	Number of #CE11 static wait cycles	0 to 7	1	R/W	
		D29	CE11WAIT1					
		D28	CE11WAIT0					
		D27	CE10WAIT3	reserved		–	–	0 when being read.
		D26	CE10WAIT2	Number of #CE10 static wait cycles	0 to 7	1	R/W	
		D25	CE10WAIT1					
		D24	CE10WAIT0					
		D23	CE9WAIT3	reserved		–	–	0 when being read.
		D22	CE9WAIT2	Number of #CE9 static wait cycles	0 to 7	1	R/W	
		D21	CE9WAIT1					
		D20	CE9WAIT0					
		D19	CE8WAIT3	reserved		–	–	0 when being read.
		D18	CE8WAIT2	Number of #CE8 static wait cycles	0 to 7	1	R/W	
		D17	CE8WAIT1					
		D16	CE8WAIT0					
		D15	CE7WAIT3	reserved		–	–	0 when being read.
		D14	CE7WAIT2	Number of #CE7 static wait cycles	0 to 7	1	R/W	
		D13	CE7WAIT1					
		D12	CE7WAIT0					
		D11	CE6WAIT3	reserved		–	–	0 when being read.
		D10	CE6WAIT2	Number of #CE6 static wait cycles	0 to 7	1	R/W	
		D9	CE6WAIT1					
		D8	CE6WAIT0					
		D7	CE5WAIT3	reserved		–	–	0 when being read.
		D6	CE5WAIT2	Number of #CE5 static wait cycles	0 to 7	1	R/W	
		D5	CE5WAIT1					
		D4	CE5WAIT0					
D3	CE4WAIT3	reserved		–	–	0 when being read.		
D2	CE4WAIT2	Number of #CE4 static wait cycles	0 to 7	1	R/W			
D1	CE4WAIT1							
D0	CE4WAIT0							

D31 Reserved

D[30:28] CE11WAIT[2:0]: Number of #CE11 Static Wait Cycles Setup Bits

These bits set the static wait cycle for accessing the #CE11 area.

Table II.3.7.3 Setting the Static Wait Cycle

CExWAIT2	CExWAIT1	CExWAIT0	Number of wait cycles
1	1	1	7 cycles
1	1	0	6 cycles
1	0	1	5 cycles
1	0	0	4 cycles
0	1	1	3 cycles
0	1	0	2 cycles
0	0	1	1 cycle
0	0	0	No wait cycle

(Default: 0b111 = 7 cycles)

D27 Reserved

D[26:24] CE10WAIT[2:0]: Number of #CE10 Static Wait Cycles Setup Bits

These bits set the static wait cycle for accessing the #CE10 area.

D23 Reserved

D[22:20] CE9WAIT[2:0]: Number of #CE9 Static Wait Cycles Setup Bits

These bits set the static wait cycle for accessing the #CE9 area.

D19 Reserved

D[18:16] CE8WAIT[2:0]: Number of #CE8 Static Wait Cycles Setup Bits

These bits set the static wait cycle for accessing the #CE8 area.

D15 **Reserved**

D[14:12] **CE7WAIT[2:0]: Number of #CE7 Static Wait Cycles Setup Bits**

These bits set the static wait cycle for accessing the #CE7 area.

D11 **Reserved**

D[10:8] **CE6WAIT[2:0]: Number of #CE6 Static Wait Cycles Setup Bits**

These bits set the static wait cycle for accessing the #CE6 area.

D7 **Reserved**

D[6:4] **CE5WAIT[2:0]: Number of #CE5 Static Wait Cycles Setup Bits**

These bits set the static wait cycle for accessing the #CE5 area.

D3 **Reserved**

D[2:0] **CE4WAIT[2:0]: Number of #CE4 Static Wait Cycles Setup Bits**

These bits set the static wait cycle for accessing the #CE4 area.

0x301508: Device Size Setup Register (pSRAMC_SLV_SIZE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Device size setup register (pSRAMC_SLV_SIZE)	00301508 (W)	D31-16	–	reserved	–	–	–	0 when being read.	
		D15	CE11SIZE1	#CE11 device size	(See below)	0	R/W		
		D14	CE11SIZE0				1		
		D13-12	–	reserved	–	–	–	–	0 when being read.
		D11	CE9SIZE1	#CE9 device size	CExSIZE[1:0]	Size	0	R/W	
		D10	CE9SIZE0				11	reserved	
		D9	CE8SIZE1	#CE8 device size	10	8 bits	0	R/W	
		D8	CE8SIZE0				1		
		D7	CE7SIZE1	#CE7 device size	01	16 bits	0	R/W	
		D6	CE7SIZE0				1		
		D5	CE6SIZE1	#CE6 device size	00	reserved	0	R/W	
		D4	CE6SIZE0				1		
		D3	CE5SIZE1	#CE5 device size	00	reserved	0	R/W	
		D2	CE5SIZE0				1		
D1	CE4SIZE1	#CE4 device size	00	reserved	0	R/W			
D0	CE4SIZE0				1				

D[31:16] Reserved

D[15:14] CE11SIZE[1:0]: Device Size Select Bits

These bits select the device size for the #CE11 area.

Table II.3.7.4 Selection of Device Size

CExSIZE1	CExSIZE0	Device size	Connected data bus
1	1	Reserved	–
1	0	8 bits	D[7:0]
0	1	16 bits	D[15:0]
0	0	Reserved	–

(Default: 0b01 = 16 bits)

D[13:12] Reserved

Note: The device size of the #CE10 area is determined by the contents in address 0xC00000 at system boot. The device size is set to 16 bits when the LSB of the 0xC00000 contents is 0 or 8 bits when it is 1.

D[11:10] CE9SIZE[1:0]: Device Size Select Bits

These bits select the device size for the #CE9 area.

D[9:8] CE8SIZE[1:0]: Device Size Select Bits

These bits select the device size for the #CE8 area.

D[7:6] CE7SIZE[1:0]: Device Size Select Bits

These bits select the device size for the #CE7 area.

D[5:4] CE6SIZE[1:0]: Device Size Select Bits

These bits select the device size for the #CE6 area.

D[3:2] CE5SIZE[1:0]: Device Size Select Bits

These bits select the device size for the #CE5 area.

D[1:0] CE4SIZE[1:0]: Device Size Select Bits

These bits select the device size for the #CE4 area.

0x30150C: Device Type Setup Register (pSRAMC_A0_BSL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Device type setup register (pSRAMC_A0_BSL)	0030150C (W)	D31-8	-	reserved			-	-	0 when being read.	
		D7	CE11TYPE	#CE11 device type	1	BSL	0	A0	0	R/W
		D6	CE10TYPE	#CE10 device type					0	R/W
		D5	CE9TYPE	#CE9 device type					0	R/W
		D4	CE8TYPE	#CE8 device type					0	R/W
		D3	CE7TYPE	#CE7 device type					0	R/W
		D2	CE6TYPE	#CE6 device type					0	R/W
		D1	CE5TYPE	#CE5 device type					0	R/W
		D0	CE4TYPE	#CE4 device type					0	R/W

D[31:8] Reserved

D7 CE11TYPE: Device Type Select Bit

This bit selects a device type (A0 or BSL) for the #CE11 area.

1 (R/W): BSL

0 (R/W): A0 (default)

Table II.3.7.5 Bus Control Signal Pin Functions in A0/BSL Mode

Pin name	A0 (default)	BSL
#CE _x	#CE _x	#CE _x
#RD	#RD	#RD
A0/#BSL	Unused	#BSL
#WRL/#WR	#WRL	#WR
#WRH/#BSH	#WRH	#BSH

D6 CE10TYPE: Device Type Select Bit

This bit selects a device type (A0 or BSL) for the #CE10 area.

1 (R/W): BSL

0 (R/W): A0 (default)

D5 CE9TYPE: Device Type Select Bit

This bit selects a device type (A0 or BSL) for the #CE9 area.

1 (R/W): BSL

0 (R/W): A0 (default)

D4 CE8TYPE: Device Type Select Bit

This bit selects a device type (A0 or BSL) for the #CE8 area.

1 (R/W): BSL

0 (R/W): A0 (default)

D3 CE7TYPE: Device Type Select Bit

This bit selects a device type (A0 or BSL) for the #CE7 area.

1 (R/W): BSL

0 (R/W): A0 (default)

D2 CE6TYPE: Device Type Select Bit

This bit selects a device type (A0 or BSL) for the #CE6 area.

1 (R/W): BSL

0 (R/W): A0 (default)

D1 CE5TYPE: Device Type Select Bit

This bit selects a device type (A0 or BSL) for the #CE5 area.

1 (R/W): BSL

0 (R/W): A0 (default)

D0 CE4TYPE: Device Type Select Bit

This bit selects a device type (A0 or BSL) for the #CE4 area.

1 (R/W): BSL

0 (R/W): A0 (default)

0x301510: Area Location Setup Register (pSRAMC_ALS)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
Area location setup register (pSRAMC_ALS)	00301510 (W)	D31-1	–	reserved	–			–	–	0 when being read.
		D0	A6LOC	Area 6 location setup	1	External	0	Internal	0	R/W

D[31:1] Reserved**D0 A6LOC: Area 6 Location Setup Bit**

This bit selects Area 6 location from between external area or internal area.

1 (R/W): External

0 (R/W): Internal (default)

Note: The S1C33E08 does not support an external device to be used for Area 6. Do not set A6LOC to 1.

II.3.8 Precautions

The BCLK pin output clock will not be divided regardless of how the BCLK divide-by ratio is set using BCLK (D0/0x301500); it is always the same as the SRAMC_CLK clock.

* **BCLK**: BCLK Divide Control Bit in the BCLK and Setup Time Control Register (D0/0x301500)

THIS PAGE IS BLANK.

II.4 SDRAM Controller (SDRAMC)

II.4.1 SDRAM Interface

The SDRAM controller allows up to 64MB of SDRAM to be connected directly to Areas 7, 19, and 22. 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 CPU and LCD, refer to Section II.4.2, “Instruction/Data Queue Buffers,” and Section II.4.3, “Bus Arbiter,” respectively.

II.4.1.1 Overview of the SDRAM Interface

The following shows the main features and specifications of the SDRAM interface.

- Up to 64MB SDRAM can be connected.
- Three SDRAM areas (Areas 7, 19, and 22) are reserved.
SDRAM configuration examples
 - 32M × 16-bit × 1 chip (64MB)
 - 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)
- Data bus width: 16 bits
- CAS latency: 1, 2, or 3
- Burst length: 2
- 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 byte writes with the DQML and DQMH pins.
- 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
- Supports Extended Mode Register Set to program drive strength, temperature compensated self refresh, and partial array self refresh.

II.4.1.2 SDRAMC Pins

Table II.4.1.2.1 lists the pins used by the SDRAMC.

Table II.4.1.2.1 SDRAMC Pin List

Pin name	I/O	Function
A[13:12]	O	Address signal output pins (SDA[12:11])
A[10:1]	O	Address signal output pins (SDA[9:0])
A[15:14]	O	Bank select signal output pins (SDBA[1:0])
D[15:0]	I/O	Data signal input/output pins (external data bus)
SDA10	O	Address signal output pin (SDA10)
SDCKE	O	SDRAM clock-enable signal output pin
SDCLK	O	SDRAM clock output pin
#SDCS	O	SDRAM chip select signal output pin
#SDRAS	O	SDRAM row address strobe signal output pin
#SDCAS	O	SDRAM column address strobe signal output pin
#SDWE	O	SDRAM write signal output pin
DQML	O	SDRAM data (to select low-order byte) input/output mask signal output pin
DQMH	O	SDRAM data (to select high-order byte) input/output mask signal output pin

Note: Some control pins above are shared with general-purpose input/output ports or other peripheral circuit input/output pins, so that functionality in the initial state is set to other than the SDRAMC. Before the SDRAMC signals assigned to these pins can be used, the functions of these pins must be switched for the SDRAMC by setting each corresponding Port Function Select Register. For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

II.4.1.3 Configuration of SDRAM

SDRAM area

The #CE7 area (area 7, area 19, or area 22) is reserved for the SDRAMC. However, the #CE7 area is configured for an SRAM area controlled with the SRAMC and the SDRAMC is disabled at initial reset. Therefore, to use an SDRAM, the #CE7 area must be configured as the SDRAM area by setting SDON (D4/0x301600) and APPON (D1/0x301610) to 1.

- * **SDON**: SDRAM Controller Enable Bit in the SDRAM Initial Register (D4/0x301600)
- * **APPON**: SDAPP Control Bit in the SDRAM Application Configuration Register (D1/0x301610)

Note: When SDON (D4/0x301600) and APPON (D1/0x301610) are set to 1, the #CE7 area external SRAM access conditions set in the SRAMC are disabled.

Setting SDRAM size and access conditions

The table below lists the conditions related to SDRAM size and timing parameters that the SDRAMC can accommodate.

Table II.4.1.3.1 SDRAM Setup Items

Setup item	Content	Control bit settings
SDRAM address configuration	32M × 16 bits × 1	ADDRC[2:0] (D[2:0]/0x301604) = 111
	16M × 16 bits × 1	ADDRC[2:0] (D[2:0]/0x301604) = 011
	8M × 16 bits × 1	ADDRC[2:0] (D[2:0]/0x301604) = 010
	4M × 16 bits × 1	ADDRC[2:0] (D[2:0]/0x301604) = 001
	1M × 16 bits × 1 (default)	ADDRC[2:0] (D[2:0]/0x301604) = 000 (default)
	16M × 8 bits × 2	ADDRC[2:0] (D[2:0]/0x301604) = 110
	8M × 8 bits × 2	ADDRC[2:0] (D[2:0]/0x301604) = 101
	2M × 8 bits × 2	ADDRC[2:0] (D[2:0]/0x301604) = 100
CAS latency	3, 2 (default) or 1	CAS[1:0] (D[3:2]/0x301610) = 11, 10 (default) or 01
Burst length	2 (fixed)	-
tRP, tRCD	1 (default) to 4 cycles	T24NS[1:0] (D[13:12]/0x301604) = 00 (default) to 11
tRAS	1 (default) to 8 cycles	T60NS[2:0] (D[10:8]/0x301604) = 000 (default) to 111
tRC, tRFC, tXSR	1 to 16 cycles (default: 15)	T80NS[3:0] (D[7:4]/0x301604) = 0000 to 1110 (default) and 1111

SDRAM address configuration

Use ADDR[2:0] (D[2:0]/0x301604) to select SDRAM size and chip configuration. This selection also sets up the bank size, column address size (page size), and row address size.

- * **ADDR[2:0]**: SDRAM Address Configuration Bits in the SDRAM Configuration Register (D[2:0]/0x301604)

Table II.4.1.3.2 Selecting SDRAM Size

ADDR2	ADDR1	ADDR0	Bank	Row	Column	SDRAM configuration	Memory size
1	1	1	4	8K	1K	32M × 16-bit × 1	64M bytes
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 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 II.4.1.3.1 SDRAM Address

- m: Column address size (number of bits) 8 bits (256), 9 bits (512), or 10 bits (1K)
- n: Row address size (number of bits) 11 bits (2K), 12 bits (4K), or 13 bits (8K)
- p: Bank address size (number of bits) 1 bit (2 banks) or 2 bits (4 banks)

When reading/writing byte data, the SDRAM controller decodes A0/BSL and WRH/BSH into DQML and DQMH.

Upper address bits that are not used (depending on memory size) are all set to 0s.

II BUS MODULES: SDRAM CONTROLLER (SDRAMC)

Figures II.4.1.3.2 and II.4.1.3.3 show examples of how to connect SDRAMs and Figure II.4.1.3.4 shows the area configuration and address ranges according to the SDRAM to be used.

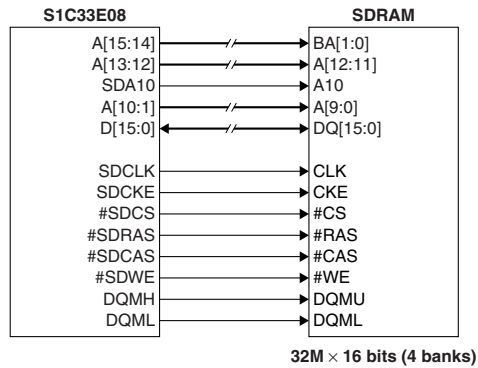


Figure II.4.1.3.2 Example of Connecting 64-MB SDRAM

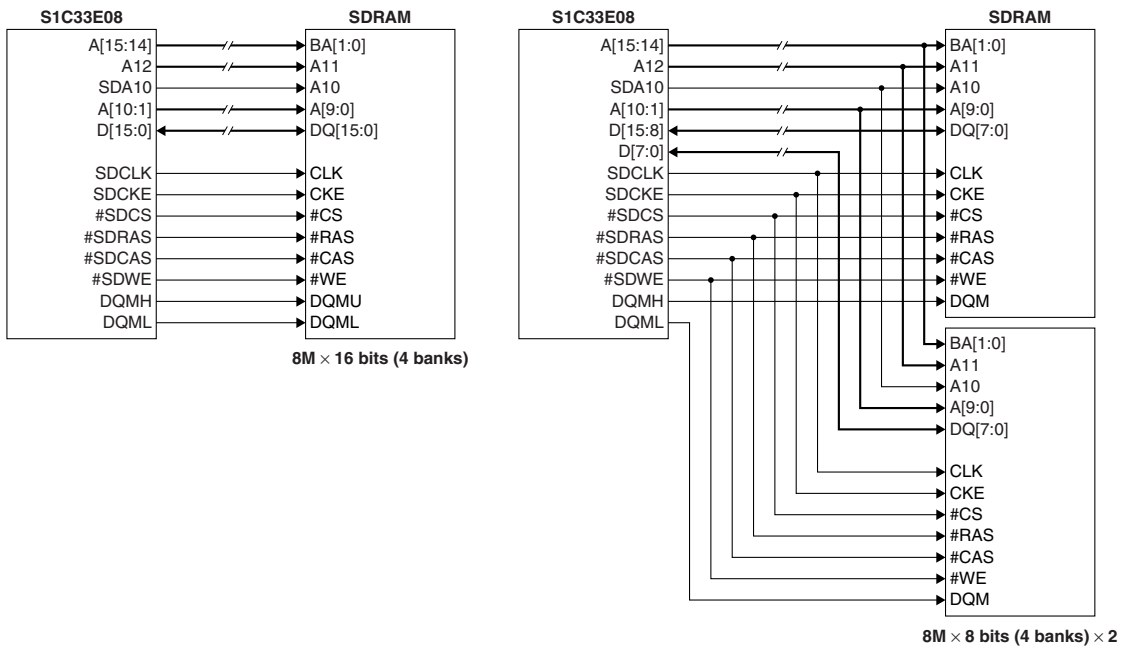


Figure II.4.1.3.3 Example of Connecting 16-MB SDRAM

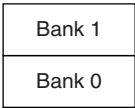
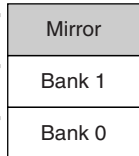
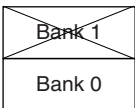
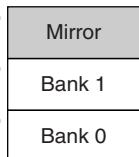
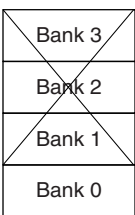
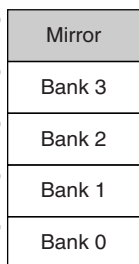
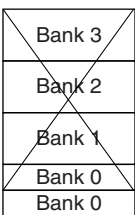
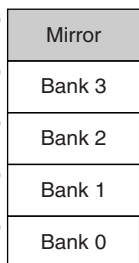
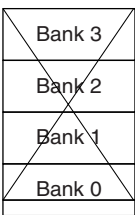
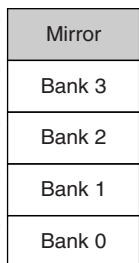
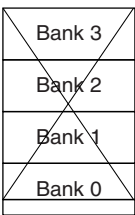
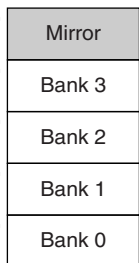
Area 7 (2MB, 0x00400000–0x005FFFFFFF)	Area 19 (256MB, 0x10000000–0x1FFFFFFF)
<p>1M × 16-bit × 1 = 2MB (ADDR[2:0] = "000")</p> <p>0x005FFFFFFF 0x00500000 0x004FFFFFFF 0x00400000</p>  <p>Area 7</p>	<p>1M × 16-bit × 1 = 2MB (ADDR[2:0] = "000")</p> <p>0x1FFFFFFF 0x10200000 0x101FFFFFFF 0x10100000 0x100FFFFFFF 0x10000000</p>  <p>Area 19</p>
<p>2M × 8-bit × 2 = 4MB (ADDR[2:0] = "100") Bank 1 cannot be accessed.</p> <p>0x005FFFFFFF 0x00400000</p>  <p>Area 7</p>	<p>2M × 8-bit × 2 = 4MB (ADDR[2:0] = "100")</p> <p>0x1FFFFFFF 0x10400000 0x103FFFFFFF 0x10200000 0x101FFFFFFF 0x10000000</p>  <p>Area 19</p>
<p>4M × 16-bit × 1 = 8MB (ADDR[2:0] = "001") Banks 1–3 cannot be accessed.</p> <p>0x005FFFFFFF 0x00400000</p>  <p>Area 7</p>	<p>4M × 16-bit × 1 = 8MB (ADDR[2:0] = "001")</p> <p>0x1FFFFFFF 0x10800000 0x107FFFFFFF 0x10600000 0x105FFFFFFF 0x10400000 0x103FFFFFFF 0x10200000 0x101FFFFFFF 0x10000000</p>  <p>Area 19</p>
<p>8M × 16-bit × 1 = 16MB (ADDR[2:0] = "010") or 8M × 8-bit × 2 = 16MB (ADDR[2:0] = "101") Upper half of bank 0 and banks 1–3 cannot be accessed.</p> <p>0x005FFFFFFF 0x00400000</p>  <p>Area 7</p>	<p>8M × 16-bit × 1 = 16MB (ADDR[2:0] = "010") or 8M × 8-bit × 2 = 16MB (ADDR[2:0] = "101")</p> <p>0x1FFFFFFF 0x11000000 0x10FFFFFFF 0x10C00000 0x10BFFFFFFF 0x10800000 0x107FFFFFFF 0x10400000 0x103FFFFFFF 0x10000000</p>  <p>Area 19</p>
<p>16M × 16-bit × 1 = 32MB (ADDR[2:0] = "011") or 16M × 8-bit × 2 = 32MB (ADDR[2:0] = "110") Upper part of bank 0 and banks 1–3 cannot be accessed.</p> <p>0x005FFFFFFF 0x00400000</p>  <p>Area 7</p>	<p>16M × 16-bit × 1 = 32MB (ADDR[2:0] = "011") or 16M × 8-bit × 2 = 32MB (ADDR[2:0] = "110")</p> <p>0x1FFFFFFF 0x12000000 0x11FFFFFFF 0x11800000 0x117FFFFFFF 0x11000000 0x10FFFFFFF 0x10800000 0x107FFFFFFF 0x10000000</p>  <p>Area 19</p>
<p>32M × 16-bit × 1 = 64MB (ADDR[2:0] = "111") Upper part of bank 0 and banks 1–3 cannot be accessed.</p> <p>0x005FFFFFFF 0x00400000</p>  <p>Area 7</p>	<p>32M × 16-bit × 1 = 64MB (ADDR[2:0] = "111")</p> <p>0x1FFFFFFF 0x14000000 0x13FFFFFFF 0x13000000 0x12FFFFFFF 0x12000000 0x11FFFFFFF 0x11000000 0x10FFFFFFF 0x10000000</p>  <p>Area 19</p>

Figure II.4.1.3.4 SDRAM Map

II BUS MODULES: SDRAM CONTROLLER (SDRAMC)

Timing setup

The following parameters can be set in conformity with SDRAM specifications before use.

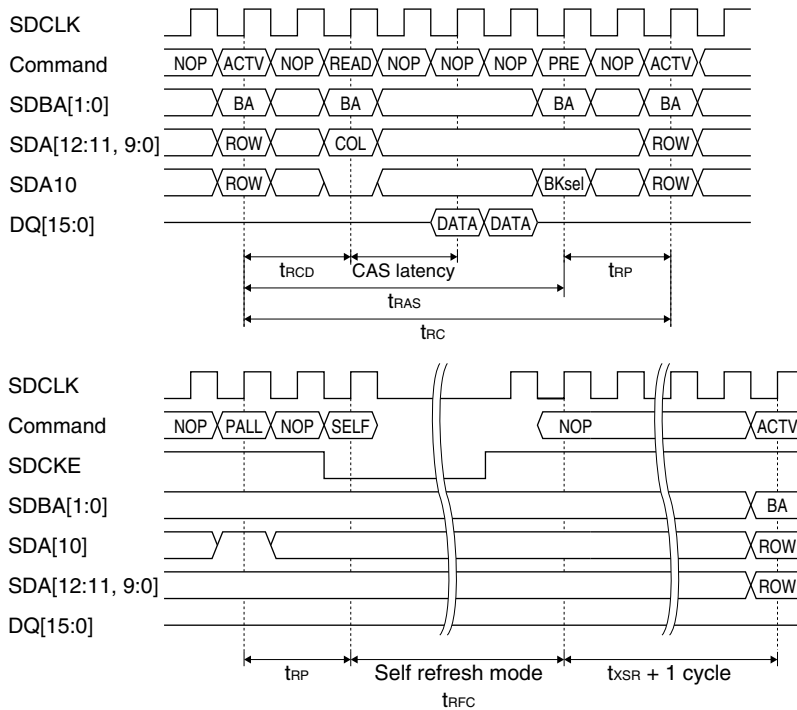


Figure II.4.1.3.5 SDRAM Timing Parameters

(1) CAS latency

CAS latency refers to the number of SDCLK clocks until data is output from SDRAM after issuing the READ command. For the SDRAMC's SDRAM interface, CAS latency can be set from 1 to 3 using CAS[1:0] (D[3:2]/0x301610).

* **CAS[1:0]**: CAS Latency Setup Bits in the SDRAM Application Configuration Register (D[3:2]/0x301610)

Table II.4.1.3.3 CAS Latency Settings

CAS1	CAS0	CAS latency
1	1	3
1	0	2
0	1	1
0	0	Reserved

When initially reset, CAS latency is initialized to 2.

(2) tRC, tRFC, tXSR

tRC: ACTIVE to ACTIVE command cycle time

tRFC: Auto-refresh cycle time

tXSR: Exit SELF REFRESH to ACTIVE command period

These timing parameters can be set from 1 to 16 cycles (in SDCLK) using T80NS[3:0] (D[7:4]/0x301604).

* **T80NS[3:0]**: Number of tRC, tRFC and tXSR Cycles Setup Bits in the SDRAM Configuration Register (D[7:4]/0x301604)

Table II.4.1.3.4 t_{RC} , t_{RFC} and t_{XSR} Settings

T80NS3	T80NS2	T80NS1	T80NS0	t_{RC} , t_{RFC} , t_{XSR}
1	1	1	1	16 cycles
1	1	1	0	15 cycles
1	1	0	1	14 cycles
1	1	0	0	13 cycles
1	0	1	1	12 cycles
1	0	1	0	11 cycles
1	0	0	1	10 cycles
1	0	0	0	9 cycles
0	1	1	1	8 cycles
0	1	1	0	7 cycles
0	1	0	1	6 cycles
0	1	0	0	5 cycles
0	0	1	1	4 cycles
0	0	1	0	3 cycles
0	0	0	1	2 cycles
0	0	0	0	1 cycle

When initially reset, t_{RC} , t_{RFC} and t_{XSR} are initialized to 15 cycles.

(3) t_{RAS}

t_{RAS} : ACTIVE to PRECHARGE command period

This timing parameter can be set from 1 to 8 cycles (in SDCLK) using T60NS[2:0] (D[10:8]/0x301604).

* **T60NS[2:0]**: Number of t_{RAS} Cycles Setup Bits in the SDRAM Configuration Register (D[10:8]/0x301604)

Table II.4.1.3.5 t_{RAS} Settings

T60NS2	T60NS1	T60NS0	t_{RAS}
1	1	1	8 cycles
1	1	0	7 cycles
1	0	1	6 cycles
1	0	0	5 cycles
0	1	1	4 cycles
0	1	0	3 cycles
0	0	1	2 cycles
0	0	0	1 cycle

When initially reset, t_{RAS} is initialized to 1 cycle.

(4) t_{RP} , t_{RCD}

t_{RP} : PRECHARGE to ACTIVE command period

t_{RCD} : ACTIVE to READ or WRITE delay time

These timing parameters can be set from 1 to 4 cycles (in SDCLK) using T24NS[1:0] (D[13:12]/0x301604).

* **T24NS[1:0]**: Number of t_{RP} and t_{RCD} Cycles Setup Bits in the SDRAM Configuration Register (D[13:12]/0x301604)

Table II.4.1.3.6 t_{RP} and t_{RCD} Settings

T24NS1	T24NS0	t_{RP} , t_{RCD}
1	1	4 cycles
1	0	3 cycles
0	1	2 cycles
0	0	1 cycle

When initially reset, t_{RP} and t_{RCD} are initialized to 1 cycle.

II.4.1.4 SDRAMC Operating Clock and SDRAM Clock

Operating clock of the SDRAMC

The SDRAMC is clocked by the following clocks generated by the CMU.

For details on how to set and control the clocks, see Section III.1, “Clock Management Unit (CMU).”

The SDRAMC operating clock supply to the SDRAMC is disabled by default setting. Each clock supply can be controlled in the CMU. Use the respective control bits to turn on only the required clocks to reduce the amount of power consumed on the chip.

1. SDAPP_CPU_CLK clock

This is the MCLK clock used for interfacing between the CPU and SDRAMC. Turn this clock on when using the SDRAMC. The clock supply can be controlled by SDAPCPU_CKE (D6/0x301B00).

- * **SDAPCPU_CKE**: SDRAMC CPU APP Clock Control Bit in the Gated Clock Control Register 0 (D6/0x301B00)

Furthermore, the SDAPP_CPU_CLK can automatically be stopped in HALT mode. By setting SDAPCPU_HCKE (D7/0x301B00) to 0, the SDAPP_CPU_CLK stops when the CPU enters HALT mode and it resumes when the CPU exits HALT mode.

- * **SDAPCPU_HCKE**: SDRAMC CPU APP Clock Control (HALT) Bit in the Gated Clock Control Register 0 (D7/0x301B00)

2. SDAPP_LCDC_CLK clock

This is the MCLK clock used for interfacing between the LCDC and SDRAMC. Turn this clock on when using the SDRAM as the video RAM. The clock supply can be controlled by SDAPLCDC_CKE (D5/0x301B00).

- * **SDAPLCDC_CKE**: SDRAMC LCDC APP Clock Control Bit in the Gated Clock Control Register 0 (D5/0x301B00)

3. Clocks for SDRAM interface and instruction/data queue buffers

The SDRAMC inputs the OSC_W clock (source clock for MCLK) to operate the SDRAM interface. Also this clock is used as SDCLK (SDRAM synchronous clock). So the SDRAM can be accessed using a clock two times faster than the CPU clock when MCLK is generated by dividing OSC_W by 2.

The OSC_W clock supply can be controlled by SDAPCPU_CKE (D6/0x301B00) and SDAPLCDC_CKE (D5/0x301B00). Either one or both are set to 1, the OSC_W clock is supplied to the SDRAMC.

Note: If the operating clock (SDCLK) is stopped while the SDRAM is being accessed, a system failure may occur due to stoppage of the SDRAM operation in uncontrolled status. The following operations stop the SDCLK, therefore, do not perform these operations when the SDRAM may be accessed.

- Setting the S1C33E08 in SLEEP status
- Switching the P21 port function from SDCLK output to general-purpose input/output
- Disabling the clock supply to the SDRAMC module

Besides the CPU, the DMA controller (when DMA transfer from/to the SDRAM is enabled) and the LCD controller (when SDRAM is configured as the VRAM for the LCDC) access the SDRAM. In this case, before performing an above operation, disable the DMA transfer and the LCDC so that the SDRAM will not be accessed.

4. SDSAPB_CLK clock

The SDSAPB_CLK is used to operate the SDRAMC control registers. To setup the registers, this clock is required. After the registers are set up, the clock supply can be stopped to reduce power consumption by setting SDSAPB_CKE (D4/0x301B00) to 0.

- * **SDSAPB_CKE**: SDRAMC SAPB I/F Clock Control Bit in the Gated Clock Control Register 0 (D4/0x301B00)

Setting any of the clock control bits above (initially 0) to 1 turns on the corresponding clock supply to the SDRAMC.

The SDRAMC operating clocks stop depending on the type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

It can be stopped by setting the CMU register.

SLEEP mode: The clock supply stops.

Therefore, the SDRAMC also stops operating in SLEEP mode.

Double frequency mode

The SDRAMC supports double frequency mode in which the SDRAM can be operated with a clock two times faster than the CPU clock. For example, when the CPU runs with a 45 MHz clock, the SDRAM can be operated with a 90 MHz clock.

To set the SDRAMC in double frequency mode:

- (1) Configure MCLK as $OSC_W \times 1/2$.
- (2) Set DBF (D5/0x301610) to 1.

- * **DBF**: Double Frequency Mode Enable Bit in the SDRAM Application Configuration Register (D5/0x301610)

Note: The SDCLK clock frequency is limited to 90 MHz, therefore, double frequency mode cannot be set when the CPU clock (MCLK) is higher than 45 MHz. In this case (normal mode), a clock up to 60 MHz (CPU maximum operating frequency) can be used for the SDRAMC.

II.4.1.5 Control and Operation of SDRAM Interface

Initializing SDRAM

To use SDRAM, it must be initialized by following the procedure below after switching power on.

1. Setting SDRAM interface pins

Switch over the pins shared with general-purpose input/output ports or other peripheral functions for SDRAM use by setting the relevant Port Function Select Register. For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

2. Initializing the SDRAMC registers

Set up the SDRAMC registers in the following order:

(1) SDRAM Configuration Register (0x301604)

Set SDRAM size/address-related parameters and access timing parameters.

(2) SDRAM Refresh Register (0x301608)

Set the auto-refresh and self-refresh counters.

(3) SDRAM Initial Register (0x301600)

Set SDON (D4) to 1 (SDRAMC enabled).

(4) SDRAM Application Configuration Register (0x301610)

Set CAS latency and enable the SDAPP and arbiter. Also enable double frequency mode and instruction queue buffer if necessary.

3. Wait after SDRAM power-on

After the power to SDRAM is turned on, the NOP state (#SDCS = 1) must be maintained for a certain time (e.g., 100 μ s, 200 μ s or more). Because this time varies with each SDRAM, refer to the specifications of the SDRAM being used.

4. Executing an SDRAM initial sequence

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.

Example 1: PALL → REF → REF → MRS (→ EMRS)

Example 2: PALL → MRS → REF → REF (→ REF → REF → REF → REF → REF → REF → 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 0x12 to the SDRAM Initial Register (0x301600); INIPRE (D1/0x301600) should be set to 1.

Then write any data to any address in the SDRAM. This dummy write is required as the trigger to send the PALL command.

* **INIPRE**: PALL Command Enable for Initialization Bit in the SDRAM Initial Register (D1/0x301600)

To execute the REF (Auto Refresh) command:

Write 0x11 to the SDRAM Initial Register (0x301600); INIREF (D0/0x301600) should be set to 1.

Then write any data to any address in the SDRAM. This dummy write is required as the trigger to send the REF command.

* **INIREF**: REF Command Enable for Initialization Bit in the SDRAM Initial Register (D0/0x301600)

When executing the REF command twice or more, insert the nop instruction between the executions.

Execute REF command → Execute nop → Execute REF command (→ REF → nop → REF . . .)

The SDRAM timing parameters set in the SDRAM Configuration Register (0x301604) is not effective in this manual initialization sequence. Therefore, enough number of nop instructions must be executed to satisfy the SDRAM timings.

To execute the MRS/EMRS (Mode Register Set/Extended Mode Register Set) command:

Write 0x14 to the SDRAM Initial Register (0x301600); INIMRS (D2/0x301600) should be set to 1.

Then write any data to the specific address shown below according to the CAS latency (MRS) or extended mode parameters (EMRS).

* **INIMRS**: MRS Command Enable for Initialization Bit in the SDRAM Initial Register (D2/0x301600)

Table II.4.1.5.1 Data Write Address to Execute the MRS/EMRS Command

CPU address	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
SDRAM address	BA1	BA0	SDA12	SDA11	SDA10	SDA9	SDA8	SDA7	SDA6	SDA5	SDA4	SDA3	SDA2	SDA1	SDA0
MRS	Mode		reserved			WB	Test mode		CAS latency			BT	Burst length		
CAS latency = 1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1
CAS latency = 2	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
CAS latency = 3	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1
EMRS	Mode		reserved						DS		TCSR		PASR		
	1	0	0	0	0	0	0	0	0	See SDRAM specifications.					

For example, to execute an MRS command with 2 of CAS latency specified, write data (any value) to address 0x10000442 (when the SDRAM is mapped to area 19) after writing 0x14 to the SDRAM Initial Register (0x301600).

Notes:

- The CAS latency specified in the MRS command must be the same as the CAS[1:0] (D[3:2]/0x301610) set value.

* **CAS[1:0]**: CAS Latency Setup Bits in the SDRAM Application Configuration Register (D[3:2]/0x301610)

- After the initial sequence commands are executed, the command enable bits must be set to 0. Write 0x10 to the SDRAM Initial Register (0x301600) after the last initialization command has been executed.

- The self-refresh function must be disabled until the SDRAM has finished initialization.

5. Checking if the SDRAM has been initialized

SDEN (D3/0x301600) is reset to 0 after power-on, and is set to 1 upon completion of the initialization sequence shown above. Make sure that SDEN (D3/0x301600) is set to 1 before the SDRAM is accessed.

In addition to being reset at power-on, SDEN (D3/0x301600) is reset to 0 by writing 0 to SDON (D4/0x301600).

* **SDEN**: SDRAM Initialize Flag in the SDRAM Initial Register (D3/0x301600)

* **SDON**: SDRAM Controller Enable Bit in the SDRAM Initial Register (D4/0x301600)

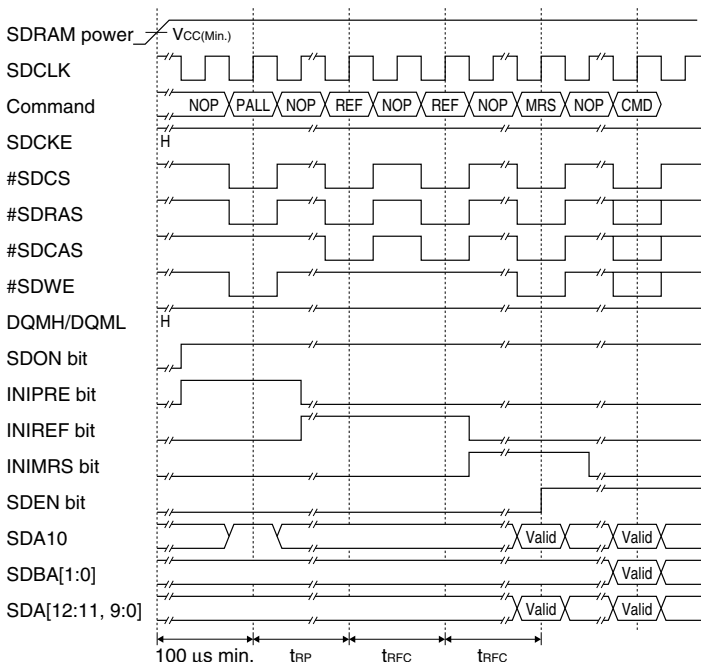


Figure II.4.1.5.1 SDRAM Power-up and Initialization

SDRAM commands

The SDRAM is controlled by commands that are comprised of a combination of high or low logic level signals. Table II.4.1.5.2 lists the commands output by the SDRAM controller.

Table II.4.1.5.2 List of the Supported SDRAM Commands

Command		Pins								
Function	Symbol	SDCKE	DQM (DQMH/L)	SDBA[1:0]	SDA10	SDA[12:11] SDA[9:0]	#SDCS	#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.

Bus operations of SDRAM

The external data bus of the S1C33E08 is 16 bits wide. Depending on the device size and data size of the instruction executed, two or more bus operations may occur. Table II.4.1.5.3 shows bus operations in the SDRAM area.

Table II.4.1.5.3 Bus Operations

Device size	Data size	R/W	A1	A0	Little endian			Access count	
					Valid signal	D[15:8] pins	D[7:0] pins		
16 bits	Byte	W	* 0	0	DQML	–	D[7:0]	1	
			* 1	0	DQMH	D[7:0]	–	1	
		R	* 0	*	Read	–	D[7:0]	1	
			* 1	*		D[7:0]	–	1	
	Half word	W	* *	*	DQMH/L	D[15:0]		1	
		R	* *	*	Read	D[15:0]		1	
	Word	W	0	*	*	DQMH/L	D[15:0]		1st
			1	*	*		D[31:16]		2nd
		R	0	*	*	Read	D[15:0]		1st
			1	*	*		D[31:16]		2nd

Read cycle

The SDRAMC always reads data from the SDRAM in bursts. The burst length is fixed at 2.

Figure II.4.1.5.2 shows an example of timing chart when reading out 2-word data from the same row address.

Example of parameter settings: CAS latency = 2, t_{RCD} = 2 cycles, t_{RAS} = 4 cycles, t_{RP} = 2 cycles

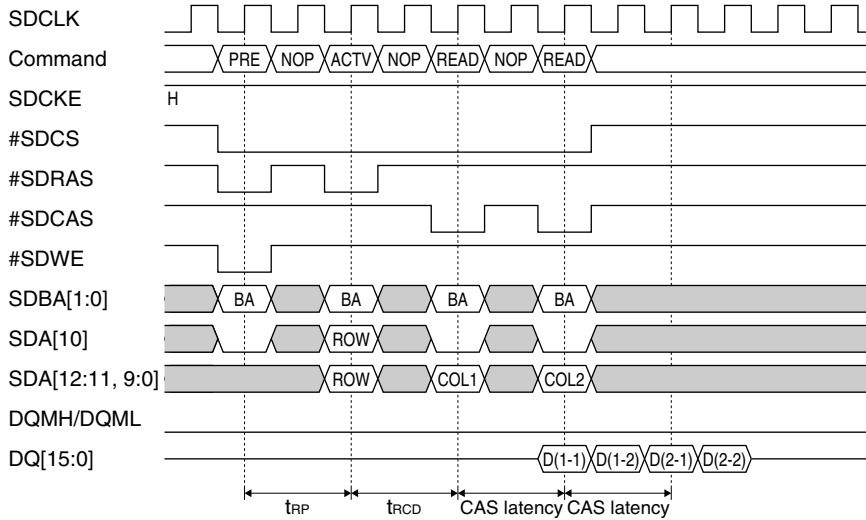


Figure II.4.1.5.2 Burst Read in the Same Page

Figure II.4.1.5.3 shows an example of a timing chart in cases where the row address is changed during burst read.

Example of parameter settings: CAS latency = 2, t_{RCD} = 2 cycles, t_{RAS} = 4 cycles, t_{RP} = 2 cycles

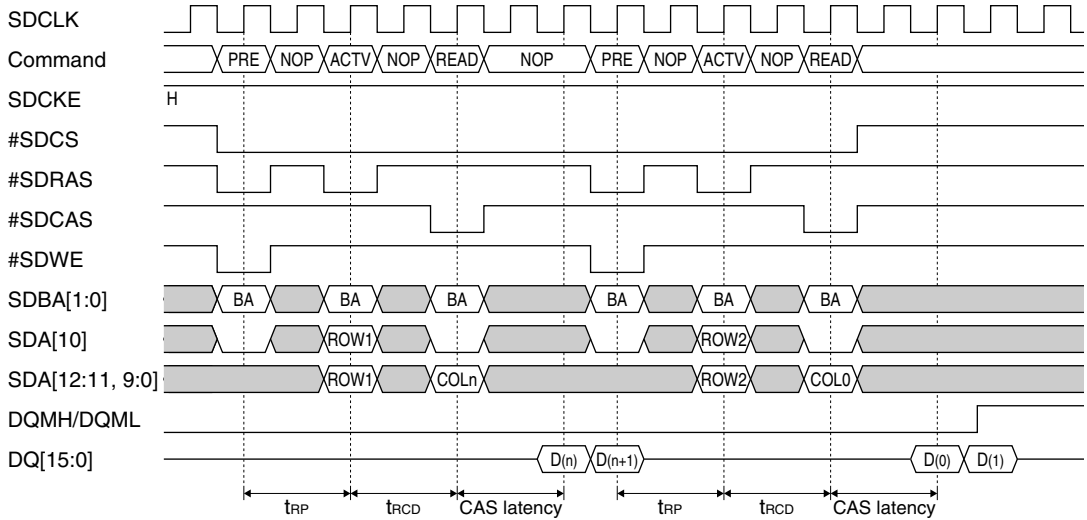


Figure II.4.1.5.3 Changing Row Address During Burst Read

Write cycle

When writing to the SDRAM, data are always written in a single operation.

Example of parameter settings: CAS latency = 2, trCD = 2 cycles, tRAS = 4 cycles, tRP = 2 cycles

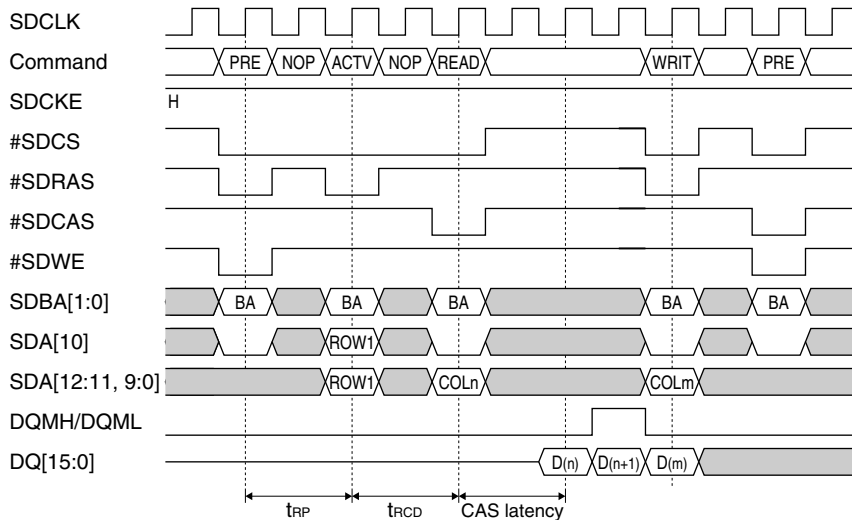


Figure II.4.1.5.4 Burst Read to Single Write (same page)

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.

Example of parameter settings: CAS latency = 2, trCD = 2 cycles, tRAS = 4 cycles, tRP = 2 cycles

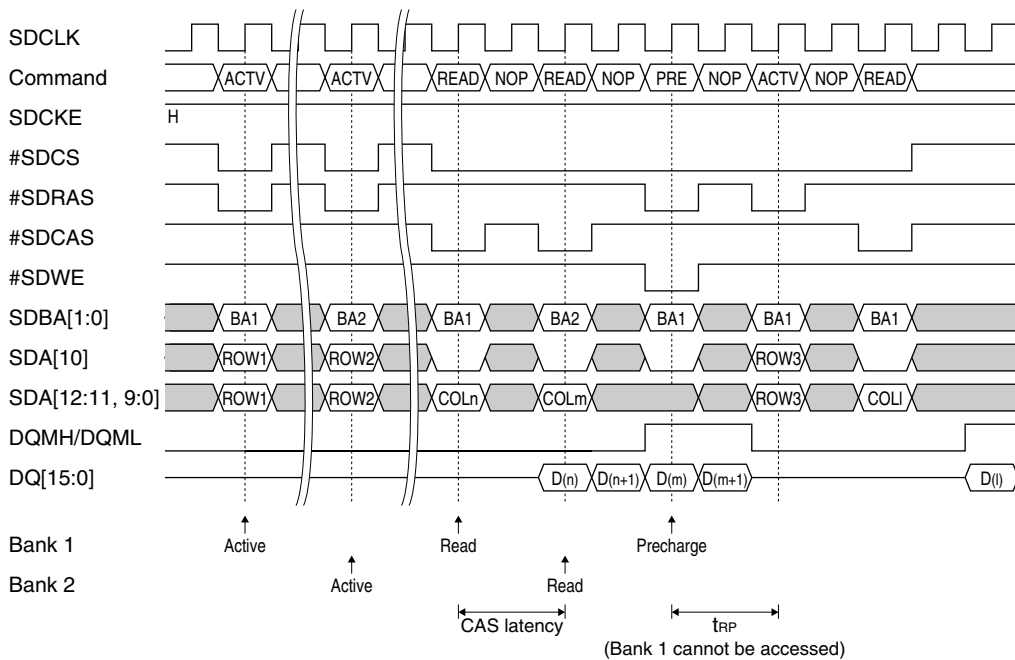


Figure II.4.1.5.5 Bank Interleaved Access

SDRAM refresh

The SDRAMC 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 SDCLK 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 SDCLK (MCLK or double of MCLK) clock frequency and the count value set in AURCO[11:0] (D[11:0]/0x301608).

* **AURCO[11:0]**: SDRAM Auto-Refresh Counter Bits in the SDRAM Refresh Register (D[11:0]/0x301608)

For AURCO[11:0] (D[11:0]/0x301608), set the appropriate value meeting the specifications of your SDRAM. The count value is obtained by the equation below.

$$\text{AURCO} \leq \frac{\text{RFP}}{\text{ROWS}} \times f_{\text{CLK}} - \text{BL} - \text{CL} - 2 \times \text{trP} - \text{trCD} - 3$$

RFP: Maximum refresh period [s]

ROWS: Row address size

fCLK: SDCLK clock frequency [Hz]

BL: Burst length (= 2)

CL: CAS latency

trP: PRECHARGE command period [Number of cycles]

trCD: ACTIVE to READ or WRITE delay time [Number of cycles]

If RFP = 64 ms, ROWS = 4,096, fCLK = 20 MHz, CL = 2, trP = 2, and trCD = 2, for example, the value to set is calculated as follows:

$$\text{AURCO} \leq \frac{0.064}{4,096} \times 20,000,000 - 2 - 2 - 2 \times 2 - 2 - 3 = 299$$

Therefore, set any value equal to or less than 299 (0x12b) for AURCO[11:0] (D[11:0]/0x301608).

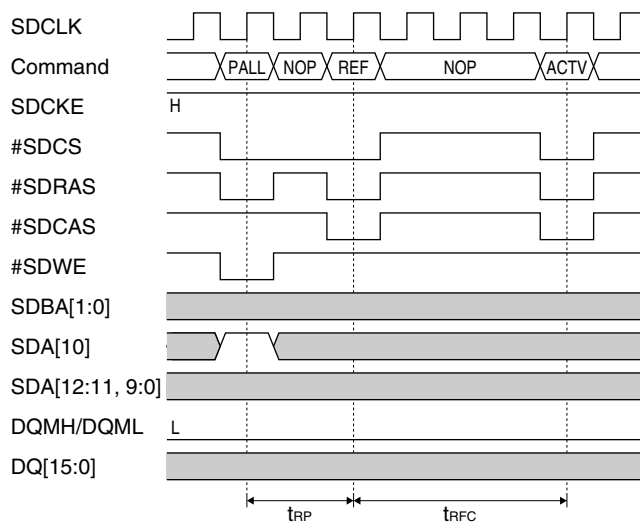


Figure II.4.1.5.6 Auto Refresh

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 SELEN (D23/0x301608) to 1. This enables the SDRAM controller to send the self-refresh command (which sets the SDCKE output to low) to the SDRAM.

* **SELEN**: SDRAM Self-Refresh Enable Bit in the SDRAM Refresh Register (D23/0x301608)

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 SDCLK clock edges, and when the designated count is reached, the SDRAM controller sends the self-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 1 to 127 by using the SELCO[6:0] (D[22:16]/0x301608). Do not set the counter to 0 when the self-refresh function is enabled.

* **SELCO[6:0]**: SDRAM Self-Refresh Counter Bits in the SDRAM Refresh Register (D[22:16]/0x301608)

When an SDRAM access occurs during self-refresh mode, SDCKE is returned high and the SDRAM is taken out of self-refresh mode. After the SDRAM access has finished, the SDRAM controller sends another self-refresh command when the designated count is reached again.

When the auto-refresh command is issued or an SDRAM access occurs, the counter will restart if the self-refresh command has not been sent to the SDRAM. Therefore, the self-refresh counter value to be set must be smaller than the auto-refresh counter value.

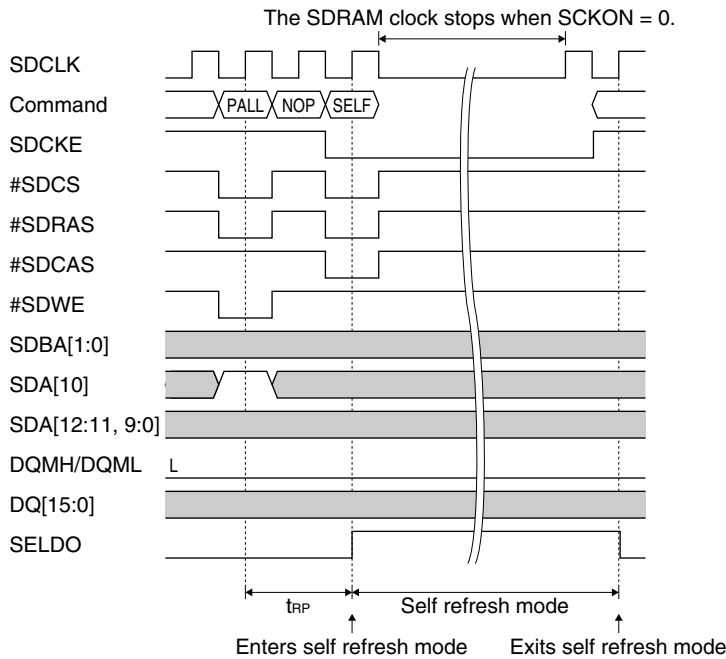


Figure II.4.1.5.7 Self Refresh

During self-refresh (while SDCKE = low), the SELDO (D25/0x301608) remains 1. Therefore, it is possible to determine whether or not self-refresh is in operation by reading this status register.

* **SELDO**: SDRAM Self-Refresh Status Bit in the SDRAM Refresh Register (D25/0x301608)

Furthermore, SDRAM clock output during self-refresh can be turned off in order to reduce the chip’s power consumption by setting the SCKON (D24/0x301608) to 0.

* **SCKON**: SDRAM Clock Enable During Self-Refresh Bit in the SDRAM Refresh Register (D24/0x301608)

Power-down mode

The S1C33E08 supports two power-down modes for the C33 PE Core (HALT and SLEEP).

HALT mode

- The LCDC will be able to access the SDRAM in HALT mode, if it is not disabled in normal mode.
- Setting SDAPCPU_CKE (D6/0x301B00) and SDAPCPU_HCKE (D7/0x301B00) determines whether the CPU and DMA will be able to access the SDRAM in HALT mode or not.
 - * **SDAPCPU_CKE**: SDRAMC CPU APP Clock Control Bit in the Gated Clock Control Register 0 (D6/0x301B00)
 - * **SDAPCPU_HCKE**: SDRAMC CPU APP Clock Control (HALT) Bit in the Gated Clock Control Register 0 (D7/0x301B00)

Table II.4.1.5.4 SDAPP_CPU_CLK Clock Status

Mode	SDAPCPU_HCKE	SDAPCPU_CKE	SDAPLCDC_CKE	SDRAM clock (*)
Normal mode	x	1	1	On (CPU, DMA and LCDC)
	x	1	0	On (CPU and DMA)
	x	0	1	On (LCDC)
	x	0	0	Off (cannot be accessed)
HALT mode	1	1	1	On (DMA and LCDC)
	1	1	0	On (DMA)
	1	0	1	On (LCDC)
	1	0	0	Off (cannot be accessed)
	0	x	x	Off (cannot be accessed)

*: () indicates the modules that can access the SDRAM.

Note: To maintain data in the SDRAM during HALT status with no SDRAM clock supplied, place the SDRAM in self-refresh mode by setting SELDO (D25/0x301608) to 1 before stopping the SDRAM clock.

SLEEP mode

In SLEEP mode, the SDRAM can be turned off to reduce power consumption by the following procedure:

1. If the CPU runs with the program stored in the SDRAM, it must be changed to a program located in the built-in RAM or a memory other than the SDRAM.
2. Turn the SDRAM power off.
3. Switch the ports used for the SDRAM to general-purpose ports.
4. Drive the data and address buses to low.
5. Set SDON (D4/0x301600) to 0 to disable the SDRAMC.
6. Execute the slp instruction.

* **SDON**: SDRAM Controller Enable Bit in the SDRAM Initial Register (D4/0x301600)

Perform the following procedure when the CPU wakes up:

1. The CPU wakes up from SLEEP status.
2. Configure the port functions for SDRAM.
3. Release the data and address buses from forced low driving.
4. Turn the SDRAM power on.
5. Wait 100 or 200 μ s for the SDRAM be stabilize according to the SDRAM specification.
6. Set SDON (D4/0x301600) to 1 to enable the SDRAMC.
7. Initialize the SDRAMC.

II.4.2 Instruction/Data Queue Buffers

II.4.2.1 Overview

The SDRAMC Module contains the SDRAMC Application Unit that is an interface unit to connect between the C33 PE Core (AHB bus) and the SDRAM Interface Unit described in Section II.4.1. It generates the read, write, address, data, and handshake signals to drive the SDRAM Interface Unit. 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 PE Core memory performance.

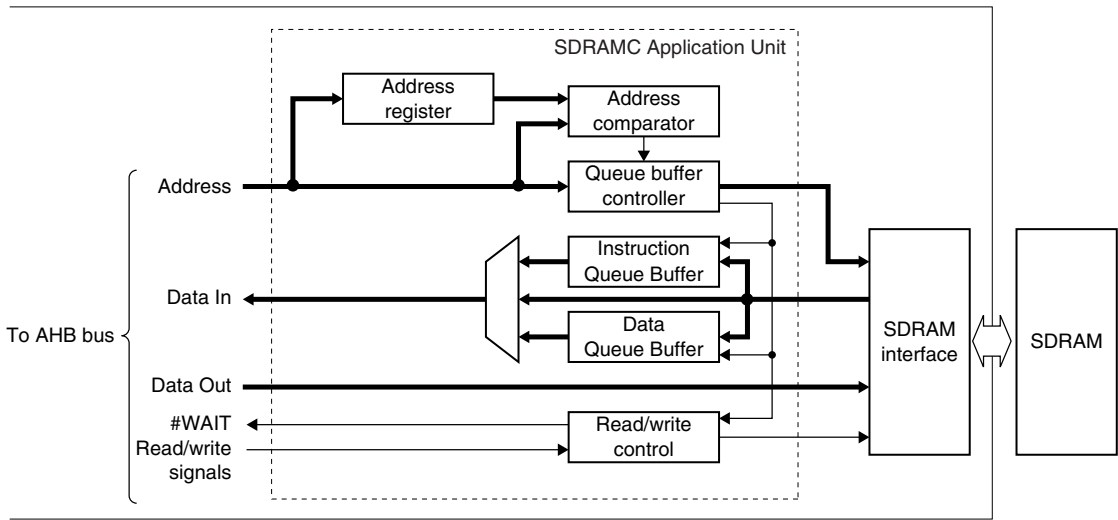


Figure II.4.2.1.1 Instruction/Data Queue Buffers

II.4.2.2 IQB (Instruction Queue Buffer)

This is a queue buffer to pre-fetch instructions and consists of 16×16 -bit D flip-flops. It is organized in 2 slots \times 8 \times 16 bits as shown in the figure below.

A[24:14]	A[13:4]	A[3:1]							
IQB address	Slot 0 address	Slot 0							
		Buf 0	Buf 1	Buf 2	Buf 3	Buf 4	Buf 5	Buf 6	Buf 7
	Slot 1 address	Slot 1							
		Buf 0	Buf 1	Buf 2	Buf 3	Buf 4	Buf 5	Buf 6	Buf 7

Figure II.4.2.2.1 Structure of IQB

IQB acts as an instruction cache located between the CPU and SDRAM when it is enabled by setting IQB (D0/0x301610) to 1.

* **IQB**: Instruction Queue Buffer Enable Bit in the SDRAM Application Configuration Register (D0/0x301610)

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 Slot 0. 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 IQB's Slot 1, then the CPU gets the needed instruction from IQB's Slot 1. 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.

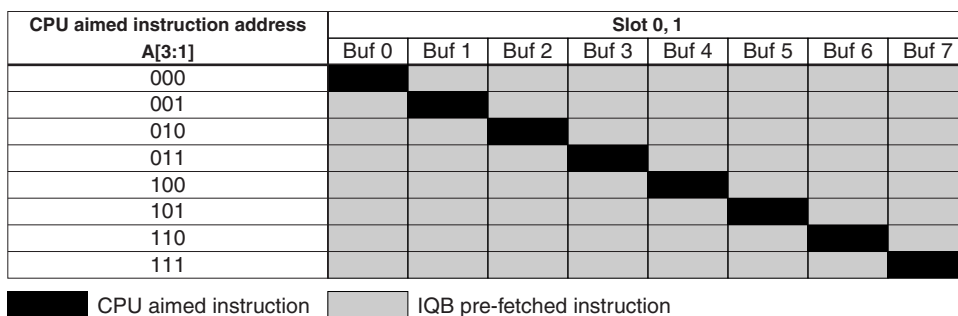


Figure II.4.2.2.2 Instructions in Slot

When IQB is disabled (IQB (D0/0x301610) = 0), the CPU can only fetch the instruction or data from the SDRAM through the Data Queue Buffer described below.

II.4.2.3 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 IQB (D0/0x301610) is set to 1 to enable IQB, DQB acts as a data buffer in which Data in the program is stored. When IQB (D0/0x301610) 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 II.4.2.3.1 lists the DQB status corresponding to the bus operation for the SDRAM.

Table II.4.2.3.1 DQB Status Corresponding to Bus Operation

Bus operation	DQB status (active or inactive)	
	When IQB is enabled	When IQB is disabled
CPU Instruction Fetch	Inactive	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.

II.4.2.4 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 PE Core and the 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 bus cycle reads the data from the IQB or DQB.

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.

Writing data to SDRAM

When the CPU writes data to the SDRAM, the internal wait signal input to the C33 PE 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.

II.4.3 Bus Arbiter

II.4.3.1 Overview

The SDRAMC contains a Bus Arbiter. When the LCDC uses the external SDRAM as the VRAM, the CPU and LCDC may access the SDRAM simultaneously. Similarly, an external bus access from the SRAMC may occur while the LCDC is accessing the SDRAM. The bus arbiter arbitrates the ownership of the external bus in such cases. The priority of the bus masters is (1) LCDC, (2) DMA, (3) CPU, and (4) SRAMC. Note, however, that the LCDC cannot interrupt a long DMA transfer unless the DMA sequential access time is set using DMAACCTIME[3:0] (D[3:0]/0x30119E).

- * **DMAACCTIME[3:0]**: IDMA and HSDMA Sequential Access Time Setup Bits in the DMA Sequential Access Time Register (D[3:0]/0x30119E)

II.4.3.2 Controlling the Bus Arbiter

At initial reset, the bus arbiter is disabled. In this case, SDRAM requests from the CPU and bus requests from the SRAMC will be accepted but the LCDC cannot request to access the SDRAM. The bus arbiter must be enabled before the LCDC can access the SDRAM. Set ARBON (D31/0x301610) to 1 to enable the bus arbiter.

- * **ARBON**: Arbiter Enable Bit in the SDRAM Application Configuration Register (D31/0x301610)

When no SDRAM access from the LCDC occurs, such as when the LCDC uses IVRAM only, the bus arbiter can be disabled by setting ARBON (D31/0x301610) to 0 to reduce current consumption.

II.4.4 Control Register Details

Table II.4.4.1 SDRAMC Register List

Address	Register name	Size	Function
0x00301600	SDRAM Initial Register (pSDRAMC_INI)	32	Enables SDRAMC and controls initialization.
0x00301604	SDRAM Configuration Register (pSDRAMC_CTL)	32	Sets SDRAM size and timing parameters.
0x00301608	SDRAM Refresh Register (pSDRAMC_REF)	32	Controls refresh.
0x00301610	SDRAM Application Configuration Register (pSDRAMC_APP)	32	Controls SDAPP module.

The following describes each SDRAMC control register.

The SDRAMC control registers are mapped to the 32-bit device area at addresses 0x301600 to 0x301610, and can be accessed in units of words, half-words or bytes.

0x301600: SDRAM Initial Register (pSDRAMC_INI)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
SDRAM initial register (pSDRAMC_INI)	00301600 (W)	D31–5	–	reserved	–			–	–	0 when being read.
		D4	SDON	SDRAM controller enable	1	Enabled	0 Disabled	0	R/W	
		D3	SDEN	SDRAM initialize flag	1	Initialized	0 Not initialized	0	R	
		D2	INIMRS	MRS command enable for init.	1	Enabled	0 Disabled	0	R/W	
		D1	INIPRE	PALL command enable for init.	1	Enabled	0 Disabled	0	R/W	
		D0	INIREF	REF command enable for init.	1	Enabled	0 Disabled	0	R/W	

D[31:5] Reserved

D4 **SDON: SDRAM Controller Enable Bit**

This bit enable the SDRAM controller.

1 (R/W): Enable

0 (R/W): Disable (default)

When SDON is set to 1, the SDRAM controller activates and outputs the SDRAM clock from the SDCLK pin. Before setting SDON to 1, the SDRAMC clocks must be supplied to the SDRAM controller.

D3 **SDEN: SDRAM Initialize Flag**

This bit indicates that the SDRAM has finished initialization (Mode Register Set).

1 (R): Initialized

0 (R): Not initialized (default)

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.

D2 **INIMRS: MRS Command Enable for Initialization Bit**

This bit enables to output the MRS (Mode Register Set) command for initialization sequence.

1 (R/W): Enable

0 (R/W): Disable (default)

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. Refer to the specifications of the SDRAM to be used for the initialization sequence.

Example 1: PALL → REF → REF → MRS (→ EMRS)

Example 2: PALL → MRS → REF → REF (→ REF → REF → REF → REF → REF → REF)

To execute the MRS/EMRS (Mode Register Set/Extended Mode Register Set) command, write 0x14 to this register (INIMRS should be set to 1). Then write any data to a specific address shown below according to the CAS latency (MRS) or extended mode parameters (EMRS).

Table II.4.4.2 Data Write Address to Execute the MRS/EMRS Command

CPU address	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
SDRAM address	BA1	BA0	SDA12	SDA11	SDA10	SDA9	SDA8	SDA7	SDA6	SDA5	SDA4	SDA3	SDA2	SDA1	SDA0
MRS	Mode		reserved			WB	Test mode		CAS latency			BT	Burst length		
CAS latency = 1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1
CAS latency = 2	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
CAS latency = 3	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1
EMRS	Mode		reserved						DS	TCSR		PASR			
	1	0	0	0	0	0	0	0	0	See SDRAM specifications.					

For example, to execute an MRS command with 2 of CAS latency specified, write data (any value) to address 0x10000442 (when the SDRAM is mapped to area 19) after writing 0x14 to the SDRAM Initial Register (0x301600).

Note: The CAS latency specified in the MRS command must be the same as the CAS[1:0] (D[3:2]/0x301610) set value.

* **CAS[1:0]:** CAS Latency Setup Bits in the SDRAM Application Configuration Register (D[3:2]/0x301610)

II BUS MODULES: SDRAM CONTROLLER (SDRAMC)

D1 INIPRE: PALL Command Enable for Initialization Bit

This bit enables to output the PALL (Precharge All) command for initialization sequence.

1 (R/W): Enable

0 (R/W): Disable (default)

To execute the PALL (Precharge All) command, write 0x12 to this register (INIPRE should be set to 1). Then write any data to any address in the SDRAM. This dummy write is required as the trigger to send the PALL command. See INIMRS (D2) for initialization sequence.

D0 INIREF: REF Command Enable for Initialization Bit

This bit enables to output the REF (Auto Refresh) command for initialization sequence.

1 (R/W): Enable

0 (R/W): Disable (default)

To execute the REF (Auto Refresh) command, write 0x11 to this register (INIREF should be set to 1). Then write any data to any address in the SDRAM. This dummy write is required as the trigger to send the REF command. See INIMRS (D2) for initialization sequence.

When executing the REF command twice or more, insert the nop instruction between the executions.

Execute REF command → Execute nop → Execute REF command (→ REF → nop → REF . . .)

- Notes:**
- The SDRAM timing parameters set in the SDRAM Configuration Register (0x301604) is not effective in this manual initialization sequence. Therefore, enough number of nop instructions must be executed to satisfy the SDRAM timings.
 - After the initial sequence commands are executed, the command enable bits must be set to 0. Write 0x10 to the SDRAM Initial Register (0x301600) after the last initialization command has been executed.
 - The self-refresh function must be disabled until the SDRAM has finished initialization.

0x301604: SDRAM Configuration Register (pSDRAMC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
SDRAM configuration register (pSDRAMC_CTL)	00301604 (W)	D31–14	–	reserved	–	–	–	0 when being read.		
		D13	T24NS1	Number of SDRAM t _{RP} and t _{RCD} cycles	T24NS[1:0] = 0 to 3 → 1 to 4 cycles	0	R/W			
		D12	T24NS0							
		D11	–	reserved	–	–	–	0 when being read.		
		D10	T60NS2	Number of SDRAM t _{RAS} cycles	T60NS[2:0] = 0 to 7 → 1 to 8 cycles	0	R/W			
		D9	T60NS1							
		D8	T60NS0							
		D7	T80NS3	Number of SDRAM t _{RC} , t _{RFC} and t _{XS} R cycles	T80NS[3:0] = 0 to 15 → 1 to 16 cycles	1	R/W			
		D6	T80NS2							
		D5	T80NS1							
		D4	T80NS0							
		D3	–	reserved	–	–	–	0 when being read.		
		D2	ADDRC2	SDRAM address configuration	ADDRC[2:0]	Configuration	0	R/W		
D1	ADDRC1	111	32M x 16 bits x 1							0
D0	ADDRC0	110	16M x 8 bits x 2							0
		101	8M x 8 bits x 2							0
		100	2M x 8 bits x 2							0
		011	16M x 16 bits x 1							0
		010	8M x 16 bits x 1							0
001	4M x 16 bits x 1	0								
000	1M x 16 bits x 1	0								

D[31:14] Reserved

D[13:12] T24NS[1:0]: Number of SDRAM t_{RP} and t_{RCD} Setup Bits

These bits set t_{RP} and t_{RCD} SDRAM timing parameters.

- t_{RP} PRECHARGE to ACTIVE command period
- t_{RCD} ACTIVE to READ or WRITE delay time

Table II.4.4.3 t_{RP} and t_{RCD} Settings

T24NS1	T24NS0	t _{RP} , t _{RCD}
1	1	4 cycles
1	0	3 cycles
0	1	2 cycles
0	0	1 cycle

(Default: 0b00 = 1 cycle)

D11 Reserved

D[10:8] T60NS[2:0]: Number of SDRAM t_{RAS} Setup Bits

These bits set t_{RAS} SDRAM timing parameters.

- t_{RAS} ACTIVE to PRECHARGE command period

Table II.4.4.4 t_{RAS} Settings

T60NS2	T60NS1	T60NS0	t _{RAS}
1	1	1	8 cycles
1	1	0	7 cycles
1	0	1	6 cycles
1	0	0	5 cycles
0	1	1	4 cycles
0	1	0	3 cycles
0	0	1	2 cycles
0	0	0	1 cycle

(Default: 0b000 = 1 cycle)

II BUS MODULES: SDRAM CONTROLLER (SDRAMC)

D[7:4] T80NS[3:0]: Number of SDRAM t_{RC}, t_{RFC}, and t_{XS}R Setup Bits

These bits set t_{RC}, t_{RFC}, and t_{XS}R SDRAM timing parameters.

- t_{RC} ACTIVE to ACTIVE command cycle time
- t_{RFC} Auto-refresh cycle time
- t_{XS}R Exit SELF REFRESH to ACTIVE command period

Table II.4.4.5 t_{RC}, t_{RFC} and t_{XS}R Settings

T80NS3	T80NS2	T80NS1	T80NS0	t _{RC} , t _{RFC} , t _{XS} R
1	1	1	1	16 cycles
1	1	1	0	15 cycles
1	1	0	1	14 cycles
1	1	0	0	13 cycles
1	0	1	1	12 cycles
1	0	1	0	11 cycles
1	0	0	1	10 cycles
1	0	0	0	9 cycles
0	1	1	1	8 cycles
0	1	1	0	7 cycles
0	1	0	1	6 cycles
0	1	0	0	5 cycles
0	0	1	1	4 cycles
0	0	1	0	3 cycles
0	0	0	1	2 cycles
0	0	0	0	1 cycle

(Default: 0b1110 = 15 cycles)

D3 Reserved

D[2:0] ADDR_C[2:0]: SDRAM Address Configuration Bits

These bits select SDRAM size and chip configuration. This selection also sets up the bank size, column address size (page size), and row address size.

Table II.4.4.6 Selecting SDRAM Size

ADDR _C 2	ADDR _C 1	ADDR _C 0	Bank	Row	Column	SDRAM configuration	Memory size
1	1	1	4	8K	1K	32M × 16-bit × 1	64M bytes
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

(Default: 0b000 = 2M bytes)

0x301608: SDRAM Refresh Register (pSDRAMC_REF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SDRAM refresh register (pSDRAMC_REF)	00301608 (W)	D31–26	–	reserved	–	–	–	0 when being read.
		D25	SELDO	SDRAM self-refresh status	1 Refresh mode 0 Done	0	R	
		D24	SCKON	SDRAM clock during self-refresh	1 Enabled 0 Disabled	0	R/W	
		D23	SELEN	SDRAM self-refresh enable	1 Enabled 0 Disabled	0	R/W	
		D22	SELCO6	SDRAM self-refresh counter	0x0 to 0x7F	1	R/W	
		D21	SELCO5					1
		D20	SELCO4					1
		D19	SELCO3					1
		D18	SELCO2					1
		D17	SELCO1					1
		D16	SELCO0					1
		D15–12	–	reserved	–	–	–	0 when being read.
		D11	AURCO11	SDRAM auto-refresh counter	0x0 to 0xFF	0	R/W	
		D10	AURCO10					0
		D9	AURCO9					0
		D8	AURCO8					0
		D7	AURCO7					1
D6	AURCO6	0						
D5	AURCO5	0						
D4	AURCO4	0						
D3	AURCO3	1						
D2	AURCO2	1						
D1	AURCO1	0						
D0	AURCO0	0						

D[31:26] Reserved**D25 SELDO: SDRAM Self-Refresh Status Bit**

This bit indicates the SDRAM self-refresh status.

1 (R): In self-refresh mode

0 (R): Not in self-refresh mode (default)

SELDO is 1 while the SDRAMC 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.

D24 SCKON: SDRAM Clock Enable During Self-Refresh Bit

This bit selects whether to stop the SDRAM clock during self-refresh or not.

1 (R/W): Enable (not stopped)

0 (R/W): Disable (stopped) (default)

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.

D23 SELEN: SDRAM Self-Refresh Enable Bit

This bit enable the SDRAM's self-refresh control function.

1 (R/W): Enable

0 (R/W): Disable (default)

Writing 1 to SELEN enables the SDRAMC 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[22:16]). SELEN = 0 disables the self-refresh function.

D[22:16] SELCO[6:0]: SDRAM Self-Refresh Counter Bits

These bits are used to set the self-refresh counter value. (Default: 0x7F)

If SELEN (D23) is set to 1 (self-refresh-enabled), the self-refresh counter starts counting up on the SDCLK 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.

D[15:12] Reserved

D[11:0] AURCO[11:0]: SDRAM Auto-Refresh Counter Bits

These bits are used to set the auto refresh counter value. (Default: 0x8C)

The auto-refresh counter counts up on the SDCLK 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.

$$\text{AURCO} \leq \frac{\text{RFP}}{\text{ROWS}} \times \text{f}_{\text{CLK}} - \text{BL} - \text{CL} - 2 \times \text{trp} - \text{trcd} - 3$$

RFP: Maximum refresh period [s]

ROWS: Row address size

f_{CLK}: SDCLK clock frequency [Hz]

BL: Burst length (= 2)

CL: CAS latency

trp: PRECHARGE command period [Number of cycles]

trcd: ACTIVE to READ or WRITE delay time [Number of cycles]

0x301610: SDRAM Application Configuration Register (pSDRAMC_APP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
SDRAM application configuration register (pSDRAMC_APP)	00301610 (W)	D31	ARBON	Arbiter enable	1	Enabled	0	Disabled	0	R/W	
		D30-6	-	reserved			-	-	-	-	0 when being read.
		D5	DBF	Double frequency mode enable	1	Enabled	0	Disabled	0	R/W	
		D4	INCR	INCR transfer enable	1	Enabled	0	Disabled	0	R/W	
		D3	CAS1 CAS0	CAS latency setup	CAS[1:0]		CAS latency		1	R/W	
		11			3	0					
		10			2						
		01			1						
D2			00	reserved							
D1	APPON	SDAPP control	1	On	0	Off	0	R/W			
D0	IQB	Instruction queue buffer enable	1	Enabled	0	Disabled	0	R/W			

D31 ARBON: Arbiter Enable Bit

This bit enables the bus arbiter.

1 (R/W): Enable

0 (R/W): Disable (default)

When ARBON is set to 1, the bus arbiter is enabled and can arbitrate the SDRAM access requests from the CPU, LCDC, and SRAMC. When ARBON is set to 0, the bus arbiter is disabled. In this case, the LCDC cannot access the SDRAM.

D[30:6] Reserved**D5 DBF: Double Frequency Mode Enable Bit**

This bit enables double frequency mode.

1 (R/W): Enable

0 (R/W): Disable (default)

When DBF is set to 1, the SDRAMC can use the SDRAM clock (90 MHz max.) two times faster than the CPU clock (45 MHz max.). Set DBF to 0 when the SDRAMC clock frequency is the same as the CPU operating clock (60 MHz max.).

D4 INCR: INCR Transfer Enable Bit

This bit enables INCR transfer. When INCR is set to 1, the SDRAMC will enhance the speed of data reading from the SDRAM by the LCDC and IDMA.

1 (R/W): Enable

0 (R/W): Disable (default)

D[3:2] CAS[1:0]: CAS Latency Setup Bits

These bits set CAS latency.

CAS latency refers to the number of SDCLK clocks until data is output from SDRAM after issuing the READ command.

Table II.4.4.7 CAS Latency Settings

CAS1	CAS0	CAS latency
1	1	3
1	0	2
0	1	1
0	0	Reserved

(Default: 0b10 = 2)

D1 APPON: SDRAM Application Unit Control Bit

This bit turns the SDRAM application unit on and off.

1 (R/W): On

0 (R/W): Off (default)

APPON must be set to 1 when using the SDRAMC.

D0 IQB: Instruction Queue Buffer Enable Bit

This bit enables the IQB (Instruction Queue Buffer).

1 (R/W): Enable

0 (R/W): Disable (default)

Setting IQB 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 IQB is set to 0, IQB is disabled and DQB is used as an instruction/data buffer.

II.4.5 Precautions

If the operating clock (SDCLK) is stopped while the SDRAM is being accessed, a system failure may occur due to stoppage of the SDRAM operation in uncontrolled status. The following operations stop the SDCLK, therefore, do not perform these operations when the SDRAM may be accessed.

- Setting the S1C33E08 in SLEEP status
- Switching the P21 port function from SDCLK output to general-purpose input/output
- Disabling the clock supply to the SDRAMC module

Besides the CPU, the DMA controller (when DMA transfer from/to the SDRAM is enabled) and the LCD controller (when SDRAM is configured as the VRAM for the LCDC) access the SDRAM. In this case, before performing an above operation, disable the DMA transfer and the LCDC so that the SDRAM will not be accessed.

THIS PAGE IS BLANK.

S1C33E08 Technical Manual

III PERIPHERAL MODULES 1 (SYSTEM)

III.1 Clock Management Unit (CMU)

III.1.1 Overview of the CMU

The Clock Management Unit (CMU) controls the operating clock supplied to each functional module. The main functions of the CMU are outlined below.

- Controls reset and NMI inputs
- Selects the system clock source (OSC3, PLL, or OSC1)
- Controls on/off of the OSC3 and OSC1 oscillator circuits
- Controls on/off and frequency multiplication rate of the PLL
- Controls SSCG
- Clock control corresponding to standby modes (SLEEP and HALT)
- Selects divide ratio of the main system clock
- Selects an external bus clock
- Controls on/off of clock supply for each functional module

Through system clock selection, oscillator circuit, and PLL control, and main system clock divide ratio selection and clock on/off control for each functional module, the CMU enables the most suitable operating clock frequency to be selected for the processing involved, as well as to turn off unnecessary clock supply, which combined with standby mode, helps to significantly reduce power consumption on the chip.

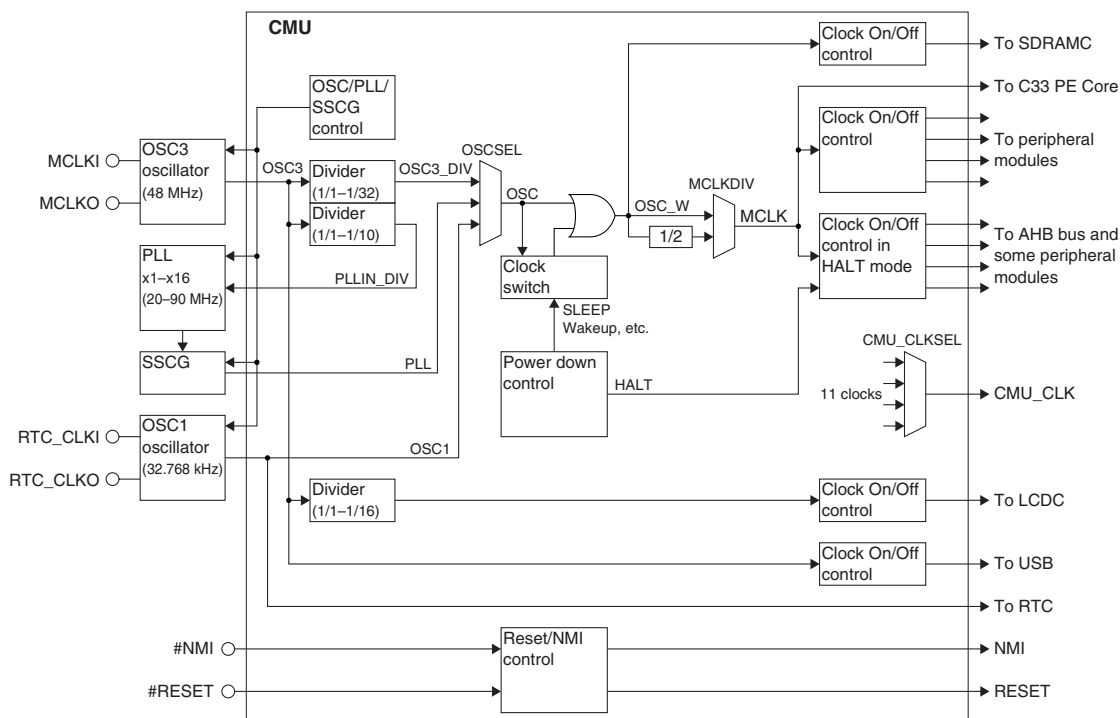


Figure III.1.1.1 CMU Block Diagram

Note: The CMU Control Registers at addresses 0x301B00–0x301B14 are write-protected. Before the CMU control registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to the Clock Control Protect Register (0x301B24). Note that since unnecessary rewrites to addresses 0x301B00–0x301B14 could lead to erratic system operation, the Clock Control Protect Register (0x301B24) should be set to other than 0x96 unless said CMU control registers must be rewritten.

III.1.2 Reset Input and Initial Reset

The CMU also has a function to generate an internal reset signal from external reset input (#RESET).

III.1.2.1 Initial Reset Pin

The #RESET pin is used for initial reset input from outside the IC. Set the #RESET pin to 0 (low) to reset the IC. The #RESET input signal is sampled with the OSC3 clock. Therefore, the chip cannot be reset when the OSC3 clock is not input or generated. Moreover, to assert the internal reset signal #RESET = 0 must be continuously detected at least three times in this sampling. The #RESET signal should be held low for at least three OSC3 clock cycles to confirm that the chip is reset. Also the internal reset signal is negated when #RESET = 1 (high) is continuously detected three times.

The S1C33E08 is reset by the low state (= 0) on the internal reset signal, and starts operating when the reset signal is released back to high (= 1).

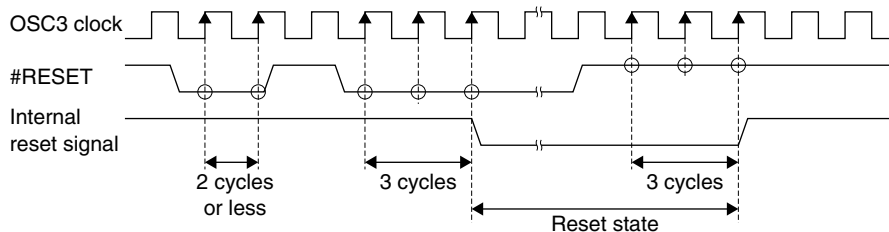


Figure III.1.2.1.1 #RESET Sampling

III.1.2.2 Initial Reset Status

The C33 PE Core and internal peripheral circuits are initialized while the internal reset signal is kept 0. The following shows the internal reset status:

CPU TTBR:	Initialized to 0xC00000
CPU PC:	The reset vector at address 0xC00000 is loaded to the PC.
CPU PSR:	All the PSR bits are reset to 0.
Other CPU registers:	Undefined
CPU operating clock:	The CPU operates with the $OSC3 \times 1/1$ clock.
Oscillator circuit:	Both the high-speed (OSC3) and low-speed (OSC1) oscillator circuits are turned on (PLL and SSCG are turned off).
Clock supply to peripheral modules:	All clocks are enabled except for the USB, SDRAMC, and LCDC.
I/O pin status:	Initialized (see Section I.3.2, "Pin Functions.")
Other peripheral modules:	Initialized or undefined (see each I/O map.)

Note: The S1C33E08 does not support a hot reset feature that maintains I/O pin status and the TTBR value.

III.1.2.3 Power-on Reset

When turning on the power for the chip, always be sure to cold reset the chip to ensure that it will start operating normally.

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

Initial reset (#RESET = 0) causes the high-speed (OSC3) oscillator circuit to start oscillating, and when the reset signal is released back high, the CPU starts operating with the OSC3 clock. The high-speed (OSC3) oscillator circuit requires a finite time until its oscillation stabilizes after it starts operating. To confirm that the CPU is started, the initial reset can only be deasserted after this oscillation stabilization time elapses.

Note: The oscillation start time of the high-speed (OSC3) oscillator circuit varies with the device used, board patterns, and operating environment. Therefore, sufficient time should be provided before the reset signal is deasserted.

Power-on sequence

To ensure that the chip will operate normally, observe the timing requirements given below when turning on the power for the chip.

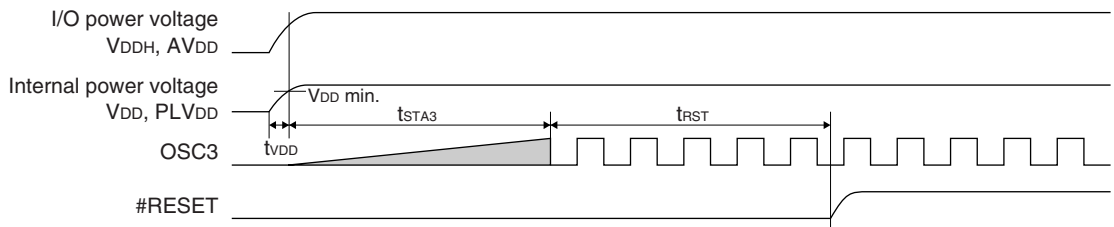


Figure III.1.2.3.1 Power-on Sequence

- (1) tvDD: The time until the power supply for the chip stabilizes after being turned on.
Turn on the power supplies in order of the following (or at the same time):
Internal core power supply (VDD, PLVDD) → I/O power supply (VDDH, AVDD) → input signal applied
- (2) tSTA3: OSC3 oscillation start time
- (3) trST: Minimum reset pulse width

Make sure #RESET is held low (= 0) for at least 3 clock cycles after the OSC3 clock supplied to the CMU has stabilized.

III.1.2.4 Precautions to be Taken during Initial Reset

Core CPU

When initially reset, all internal registers of the core CPU (except PSR) become unstable. Therefore, these registers must be initialized in a program. In particular, the Stack Pointer (SP) should always be initialized before accessing the stack. Note that NMI requests are masked in hardware until data is written to the SP after initial reset, to prevent erratic operation.

Internal RAM

The content of internal RAM becomes unstable when initially reset. Internal RAM must be initialized as required.

High-speed (OSC3) oscillator circuit

When initially reset, the high-speed (OSC3) oscillator circuit starts oscillating, and when the reset signal is deasserted, the CPU starts operating with the OSC3 clock. To prevent erratic operation due to an instable clock when the chip is reset at power-on or while the high-speed (OSC3) oscillator circuit is idle, the reset signal should not be deasserted until after oscillation stabilizes.

Low-speed (OSC1) oscillator circuit

When the chip is reset at power-on or while the low-speed (OSC1) oscillator circuit is idle, the low-speed (OSC1) oscillator circuit also starts oscillating. The low-speed (OSC1) oscillator circuit requires a longer time for oscillation to stabilize than the high-speed (OSC3) oscillator circuit. (See the electrical characteristics table.) To prevent erratic operation due to an instable clock, the OSC1 clock should not be used until after this stabilization time elapses.

Input/output ports and input/output pins

Initial reset initializes the control and data registers of the input/output ports, therefore, be set up back again in a program.

Other internal peripheral circuits

The control and data registers of other peripheral circuits are initialized or made unstable by initial reset. Therefore, these registers should be set up as required in a program.

For details on how peripheral circuits are initialized by initial reset, see each I/O map or circuit description.

III.1.3 NMI Input

The external NMI signal is input from the #NMI pin to the CMU, then forwarded to the CPU. For details about NMI exception handling by the CPU, refer to the S1C33 Family C33 PE Core Manual.

- Notes:**
- At least a 3-system clock width of low pulse is required to generate NMI. After the NMI signal falls, maintain it at a low level for 3 or more clock cycles.
 - NMI cannot be nested. The CPU keeps NMI input masked out until the reti instruction is executed after an NMI exception occurred.

III.1.4 Selecting the System Clock Source

The CMU has the following three clock inputs, one of which can be selected as the source clock (OSC) for the system.

1. OSC3 clock

This clock is generated by the OSC3 oscillator circuit or supplied from an external source through the MCLKI pin. For details about the OSC3 oscillator circuit, see Section III.1.5.1, “OSC3 Oscillator Circuit.”

2. OSC1 clock

This is the source clock (32.768 kHz, typ.) for the Real Time Clock (RTC). When high-speed operation is unnecessary, this low-speed clock may be used to operate the system, thus helping to reduce power consumption on the chip. For details about the OSC1 oscillator circuit, see Section III.1.5.3, “OSC1 Oscillator Circuit.”

3. PLL clock

This is the PLL output clock through the SSCG module. The PLL multiplies the OSC3 divided clock frequency by a given value to generate a clock for high-speed operation. The frequency multiplication rate can be set to one from $\times 1$ to $\times 16$, note, however, that it depends on the OSC3 divided clock frequency (maximum output frequency is 90 MHz).

For details about the PLL, see Section III.1.6, “Controlling the PLL.”

The clock source can be selected as shown in Table III.1.4.1 by using OSCSEL[1:0] (D[3:2]/0x301B08).

* **OSCSEL[1:0]**: OSC Clock Select Bits in the System Clock Control Register (D[3:2]/0x301B08)

Table III.1.4.1 Selection of the System Clock Source

OSCSEL1	OSCSEL0	Clock source
1	1	PLL
1	0	OSC3
0	1	OSC1
0	0	OSC3

(Default: 0b00 = OSC3)

The clock source changed here is not reflected until after the CPU returns from SLEEP mode. Therefore, the slp instruction must be executed once after setting OSCSEL[1:0] (D[3:2]/0x301B08). Although the CPU returns from SLEEP mode to normal operation by an external interrupt from a port, for example, several functions are provided for use in clock source changes, thus automatically returning the CPU from SLEEP mode a certain time after slp instruction execution or leaving the OSC3 oscillator circuit turned on during SLEEP mode. Section III.1.11, “Standby Modes,” describes these methods of control in detail.

Note: When clock sources are changed, the CMU control registers must be set so that the CMU is supplied with a clock from the selected clock source upon returning from SLEEP mode immediately after the change. Otherwise, the chip does not restart after the return from SLEEP mode.

III.1.5 Controlling the Oscillator Circuit

III.1.5.1 OSC3 Oscillator Circuit

The OSC3 oscillator circuit generates the main clock with which to operate the C33 PE Core and internal peripheral circuits.

Input/output pins of the OSC3 oscillator circuit

Table III.1.5.1.1 lists the input/output pins of the OSC3 oscillator circuit.

Table III.1.5.1.1 Input/Output Pins of the OSC3 Oscillator Circuit

Pin name	I/O	Function
MCLKI	I	OSC3 oscillator input pin: Crystal/ceramic oscillator or external clock input
MCLKO	O	OSC3 oscillator output pin: Crystal/ceramic oscillator (left open when using external clock input)

Structure of the oscillator circuit

The OSC3 oscillator circuit accommodates a crystal/ceramic oscillator and external clock input. The core voltage V_{DD} supplies power for this circuit.

Figure III.1.5.1.1 shows the structure of the OSC3 oscillator circuit.

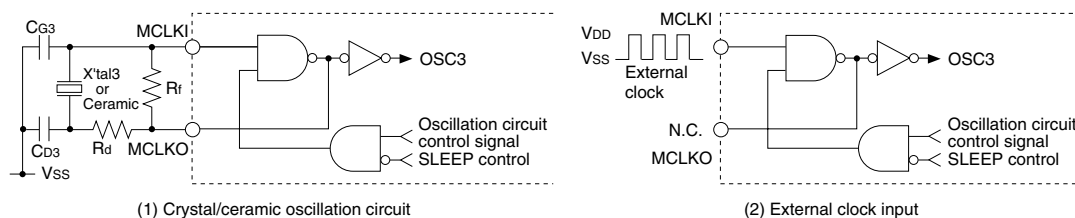


Figure III.1.5.1.1 OSC3 Oscillator Circuit

For use as a crystal or ceramic oscillator circuit, connect a crystal (X'tal3) or ceramic resonator and a feedback resistor (R_f), two capacitors (C_{G3} , C_{D3}) and, if necessary, a drain resistor (R_d) to the MCLKI and MCLKO pins and V_{SS} .

To use an external clock, leave the MCLKO pin open and input a V_{DD} -level clock (with a 50% duty cycle) to the MCLKI pin.

The range of oscillation frequencies is as follows:

- Crystal oscillator: 5 MHz (min.) to 48 MHz (max.)
- Ceramic oscillator: 48 MHz fixed
- External clock input: 5 MHz (min.) to 48 MHz (max.)
- A 48 MHz clock source with 0.25% of accuracy should be connected for using the USB function.

For details of oscillation characteristics and external clock input characteristics, see “Electrical Characteristics.”

Oscillation control

CMU register control bit SOSC3 (D1/0x301B08) is used to control OSC3 oscillation.

* **SOSC3**: High-speed Oscillation (OSC3) On/Off Control Bit in the System Clock Control Register (D1/0x301B08)

Setting this control bit to 0 causes the OSC3 oscillator circuit to stop; setting it to 1 causes the OSC3 oscillator circuit to start oscillating, thereby outputting a clock signal waveform. When initially reset, this bit is set to 1 for enabling OSC3 oscillation.

Note: When the oscillator is made to start oscillating by setting SOSC3 (D1/0x301B08) from 0 to 1, a finite time is required until oscillation stabilizes (see “Electrical Characteristics”). To prevent system malfunction, do not use the oscillator-derived clock until this oscillation stabilization time elapses.

III.1.5.2 Setting the OSC3 Divider

An OSC3 divided clock can be used as the system clock when OSC3 is selected as the system clock source. Setting the system clock to the lowest frequency possible according to the processing can reduce current consumption. The OSC3 divider generates six kinds of clocks from OSC3•1/1 to OSC3•1/32. Select a divided clock from those six clocks using OSC3DIV[2:0] (D[10:8]/0x301B08).

* **OSC3DIV[2:0]**: OSC3 Clock Divider Select Bits in the System Clock Control Register (D[10:8]/0x301B08)

Table III.1.5.2.1 Selecting an OSC3 Divided Clock

OSC3DIV2	OSC3DIV1	OSC3DIV0	CMU_CLK
1	1	1	OSC3•1/1
1	1	0	OSC3•1/1
1	0	1	OSC3•1/32
1	0	0	OSC3•1/16
0	1	1	OSC3•1/8
0	1	0	OSC3•1/4
0	0	1	OSC3•1/2
0	0	0	OSC3•1/1

(Default: 0b000 = OSC3•1/1)

A divided clock can be selected at any time. However, up to 32 OSC3 clock cycles are required before the clocks are actually changed after altering the register values.

III.1.5.3 OSC1 Oscillator Circuit

The S1C33E08 contains an oscillator circuit (OSC1) used to generate a 32.768 kHz (typ.) clock as the clock source for timekeeping operation of the RTC.

The OSC1 clock can also be used as a power-saving operating clock for the core system or peripheral circuits.

Input/output pins of the OSC1 oscillator circuit

Table III.1.5.3.1 lists the input/output pins of the OSC1 oscillator circuit.

Table III.1.5.3.1 Input/Output Pins of the Low-speed (OSC1) Oscillator Circuit

Pin name	I/O	Function
RTC_CLKI	I	OSC1 input pin: Crystal oscillator or external clock input
RTC_CLKO	O	OSC1 output pin: Crystal oscillator output (left open when external clock is input)

Structure of the OSC1 oscillator circuit

The OSC1 oscillator circuit accommodates a crystal oscillator and external clock input. The core voltage VDD supplies power for this circuit. Figure III.1.5.3.1 shows the structure of the OSC1 oscillator circuit.

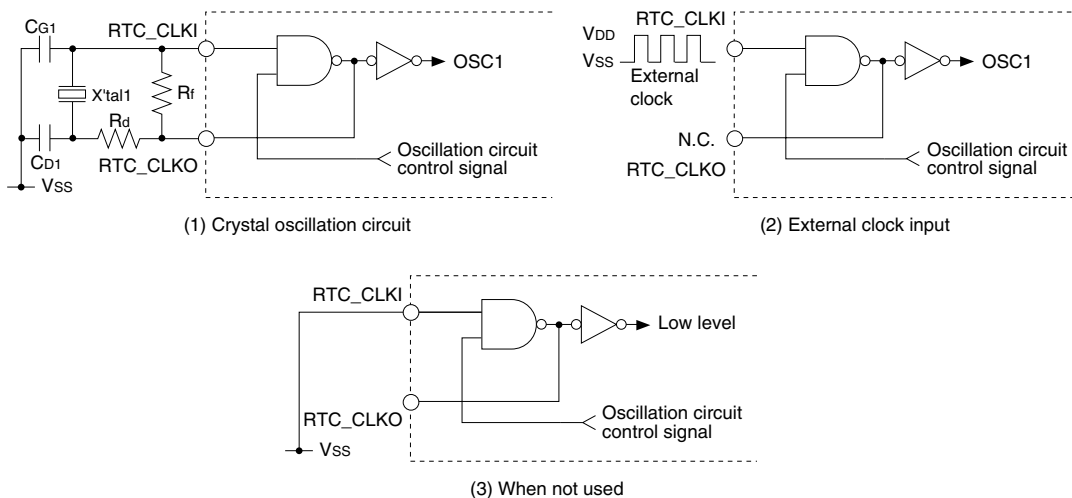


Figure III.1.5.3.1 OSC1 Oscillator Circuit

For use as a crystal oscillator circuit, connect a crystal resonator X'tal1 (32.768 kHz, typ.), feedback resistor (R_f), two capacitors (C_{G1}, C_{D1}), and, if necessary, a drain resistor (R_d) to the RTC_CLKI and RTC_CLKO pins and V_{SS}, as shown in Figure III.1.5.3.1 (1).

To use an external clock, leave the RTC_CLKO pin open and input a V_{DD} level clock (whose duty cycle is 50%) to the RTC_CLKI pin. Do not input V_{DDH} or other I/O level clocks.

The oscillator frequency/input clock frequency is 32.768 kHz (typ.). Make sure the crystal resonator or external clock used in the RTC has this clock frequency. With any other clock frequencies, the RTC cannot be used for timekeeping purposes.

For details of oscillation characteristics and the input characteristics of external clock, see “Electrical Characteristics.”

When not using the OSC1 oscillator circuit, connect the RTC_CLKI pin to V_{SS} and leave the RTC_CLKO pin open.

Oscillation control

Internal control bit SOSC1 (D0/0x301B08) of the CMU register is used to control OSC1 oscillation.

* **SOSC1**: Low-speed Oscillation (OSC1) On/Off Control Bit in the System Clock Control Register (D0/0x301B08)

Setting this control bit to 0 causes the OSC1 oscillator circuit to stop; setting it to 1 causes the OSC1 oscillator circuit to start oscillating, thereby outputting a clock signal waveform. When initially reset, this bit is set to 1, so that the OSC1 oscillator circuit continues oscillating.

Note: When the oscillator is made to start oscillating by setting SOSC1 (D0/0x301B08) from 0 to 1, a finite time (of up to 3 seconds) is required until oscillation stabilizes. To prevent system malfunction, do not use the oscillator-derived clock until this oscillation stabilization time elapses.

III.1.6 Controlling the PLL

The PLL multiplies the OSC3 clock frequency by a given value to generate a source clock for high-speed operation.

III.1.6.1 On/Off Control of the PLL

PLLPOWR (D0/0x301B0C) can be used to turn the PLL on or off.

* **PLLPOWR**: PLL On/Off Control Bit in the PLL Control Register (D0/0x301B0C)

Setting PLLPOWR (D0/0x301B0C) to 1 initiates PLL operation. When initially reset, PLLPOWR (D0/0x301B0C) is set to 0 (power-down mode), with the PLL turned off.

Note: Immediately after the PLL is started by setting PLLPOWR (D0/0x301B0C) to 1, an output clock stabilization wait time is required (e.g., 200 μ s in the S1C33E08). When the clock source for the system is switched over to the PLL, allow for this wait time after the PLL has turned on.

III.1.6.2 Selecting the PLL Input Clock

The PLL input clock can be selected from among 10 kinds of OSC3 divided clocks, OSC3•1/1 to OSC3•1/10, using PLLINDIV[3:0] (D[23:20]/0x301B08).

* **PLLINDIV[3:0]**: PLL Input Clock Source Divider Select Bits in the System Clock Control Register (D[23:20]/0x301B08)

Table III.1.6.2.1 Selecting the PLL Input Clock

PLLINDIV3	PLLINDIV2	PLLINDIV1	PLLINDIV0	PLL input clock
1	0	0	1	OSC3•1/10
1	0	0	0	OSC3•1/9
0	1	1	1	OSC3•1/8
0	1	1	0	OSC3•1/7
0	1	0	1	OSC3•1/6
0	1	0	0	OSC3•1/5
0	0	1	1	OSC3•1/4
0	0	1	0	OSC3•1/3
0	0	0	1	OSC3•1/2
0	0	0	0	OSC3•1/1
Other				OSC3•1/8

(Default: 0b0111 = OSC3•1/8)

Notes:

- The PLL input clock can only be selected when the PLL is turned off (PLLPOWR (D0/0x301B0C) = 0) and the clock source is other than the PLL (OSCSEL[1:0] (D[3:2]/0x301B08) = 0–2). If the PLL input clock is changed while the system is operating with the PLL clock, the system may operate erratically.

- For the range of the input clock frequency, see “Electrical Characteristics.”

III.1.6.3 Setting the Frequency Multiplication Rate

The PLL frequency multiplication rate can be specified as shown in Table III.1.6.3.1 by using PLLN[3:0] (D[7:4]/0x301B0C).

* **PLLN[3:0]**: PLL Multiplication Rate Setup Bits in the PLL Control Register (D[7:4]/0x301B0C)

Table III.1.6.3.1 PLL Frequency Multiplication Rates

PLLN3	PLLN2	PLLN1	PLLN0	Multiplication rate
1	1	1	1	x16
1	1	1	0	x15
1	1	0	1	x14
1	1	0	0	x13
1	0	1	1	x12
1	0	1	0	x11
1	0	0	1	x10
1	0	0	0	x9
0	1	1	1	x8
0	1	1	0	x7
0	1	0	1	x6
0	1	0	0	x5
0	0	1	1	x4
0	0	1	0	x3
0	0	0	1	x2
0	0	0	0	x1

(Default: 0b0000 = x1)

PLL output clock frequency = PLL input clock frequency × multiplication rate

- Notes:**
- The frequency multiplication rate must be set so that the PLL output clock frequency does not exceed the upper-limit operating clock frequency. For the multiplication rates that can be set and the range of the output clock frequency, see “Electrical Characteristics.”
 - The frequency multiplication rate can only be set when the PLL is turned off (PLLPOWER (D0/0x301B0C) = 0) and the clock source is other than the PLL (OSCSEL[1:0] (D[3:2]/0x301B08) = 0–2). If the frequency multiplication rate is changed while the system is operating with the PLL clock, the system may operate erratically.

III.1.6.4 Other PLL Settings

V-Divider

To ensure that frequency f_{vco} obtained by $\langle \text{output frequency} \times W \rangle$ falls within the range of 100 to 400 MHz, set the proper W value by using PLLV[1:0] (D[3:2]/0x301B0C). Lower value is better for low power consumption.

* PLLV[1:0]: PLL V-Divider Setup Bits in the PLL Control Register (D[3:2]/0x301B0C)

Table III.1.6.4.1 Settings of the W Value

PLLV1	PLLV0	W
1	1	8
1	0	4
0	1	2
0	0	Not allowed

(Default: 0b01 = 2)

VCO Kv constant (VC value)

According to the range of f_{vco} frequencies obtained by $\langle \text{output frequency} \times W \rangle$, set the VCO Kv circuit constant (VC value) by using PLLVC[3:0] (D[15:12]/0x301B0C).

* PLLVC[3:0]: PLL VCO Kv Setup Bits in the PLL Control Register (D[15:12]/0x301B0C)

Table III.1.6.4.2 Settings of the VC Value

PLLVC3	PLLVC2	PLLVC1	PLLVC0	f_{vco} [MHz]
1	0	0	0	$360 < f_{vco} \leq 400$
0	1	1	1	$320 < f_{vco} \leq 360$
0	1	1	0	$280 < f_{vco} \leq 320$
0	1	0	1	$240 < f_{vco} \leq 280$
0	1	0	0	$200 < f_{vco} \leq 240$
0	0	1	1	$160 < f_{vco} \leq 200$
0	0	1	0	$120 < f_{vco} \leq 160$
0	0	0	1	$100 \leq f_{vco} \leq 120$
Other				Not allowed

(Default: 0b0001)

LPF resistance value (RS value)

According to the input clock frequency, set the LPF resistance value (RS value) of the PLL by using PLLRS[3:0] (D[11:8]/0x301B0C).

* PLLRS[3:0]: PLL LPF Resistance Setup Bits in the PLL Control Register (D[11:8]/0x301B0C)

Table III.1.6.4.3 Settings of the RS Value

PLLRS3	PLLRS2	PLLRS1	PLLRS0	f_{REFCK} [MHz]
1	0	1	0	$5 \leq f_{REFCK} < 20$
1	0	0	0	$20 \leq f_{REFCK} \leq 150$
Other				Not allowed

(Default: 0b1000)

LPF capacitance value (CS value)

Bits to set the LPF capacitance value (CS value) is provided in the CMU control registers, PLLCS[1:0] (D[23:22]/0x301B0C). However, do not alter the value of these bits, and leave them as initially set (0b00).

* PLLCS[1:0]: PLL LPF Capacitance Setup Bits in the PLL Control Register (D[23:22]/0x301B0C)

Charge pump current value (CP value)

Bits to set the charge pump current value (CP value) is provided in the CMU control registers, PLLCP[4:0] (D[20:16]/0x301B0C). However, do not alter the value of these bits, and leave them as initially set (0b10000).

* PLLCP[4:0]: PLL Charge Pump Current Setup Bits in the PLL Control Register (D[20:16]/0x301B0C)

Table III.1.6.4.4 Example PLL Settings

PLL input clock	PLL output clock	PLLN[3:0]	PLLV[1:0]	PLLVC[3:0]	PLLRS[3:0]
6 MHz	90 MHz	x15, 0b1110	0b01	0b0011	0b1010
	60 MHz	x10, 0b1001	0b01	0b0001	0b1010
10 MHz	80 MHz	x8, 0b0111	0b01	0b0010	0b1010
	40 MHz	x4, 0b0011	0b10	0b0010	0b1010
20 MHz	80 MHz	x4, 0b0011	0b01	0b0010	0b1000
	40 MHz	x2, 0b0001	0b10	0b0010	0b1000
45 MHz	90 MHz	x2, 0b0001	0b01	0b0011	0b1000

Note: The PLL can only be set up when the PLL is turned off (PLLPOWR (D0/0x301B0C) = 0) and the clock source is other than the PLL (OSCSEL[1:0] (D[3:2]/0x301B08) = 0–2). If settings are changed while the system is operating with the PLL clock, the system may operate erratically.

III.1.6.5 Power Supply for PLL

The power for PLL is supplied from the PLV_{DD} and PLV_{SS} pins (separately from the core power supply) to prevent the effects of noise. Make sure that the following voltages are supplied to the respective pins.

PLV_{DD} pin: Supply V_{DD} level voltage.

PLV_{SS} pin: Connect to V_{SS} level.

For pin assignments, see Section I.3, “Pin Description.”

III.1.7 Control of the SSCG

The Spread Spectrum Clock Generator (SSCG) is a circuit used to reduce Electromagnetic Interference (EMI) noise by spreading the spectrum (or performing SS modulation) for the PLL output clock signal. The SS modulation is effective for all operating clocks for the core and peripheral circuits (except the RTC that uses the OSC1 clock) when the PLL output clock has been selected as the system clock source and only this case has the effect of reducing noise.

Note: When the OSC3 or OSC1 clock is selected as the system clock source, SS modulation is not performed for the operating clock (system clock).

* About spectrum spread (SS modulation)

The SSCG performs SS modulation by adjusting the width of the high section of the input clock. This adjustment is made by increasing or reducing the set value of the internal delay adjust circuit of the SSCG. The maximum width within which the set value is changed constitutes the maximum frequency change width. The relevant control register is used to set the upper-limit value of this width. In the SSCG, an interval timer adjusts the interval at which the set value changes. The relevant control register is also used to set this interval (frequency change cycle).

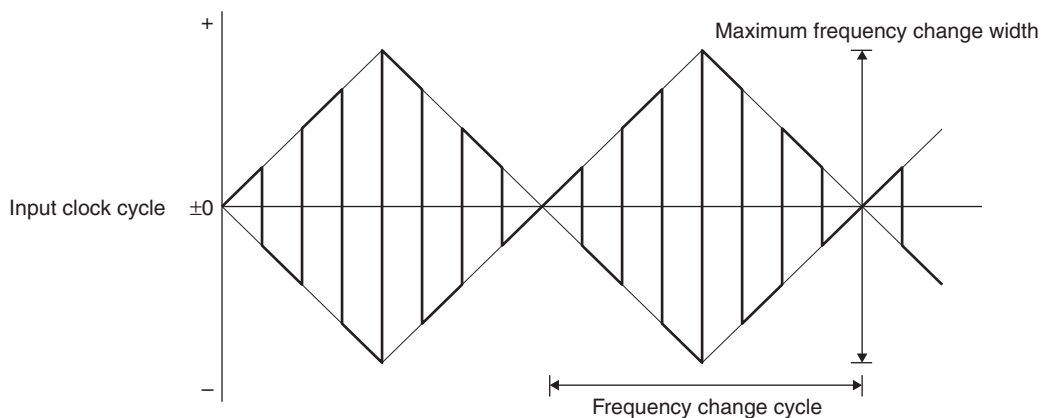


Figure III.1.7.1 SS Modulation

III.1.7.1 Turning the SSCG On/Off

The SSCG can be turned on or off by using SSMCON (D0/0x301B10).

* **SSMCON:** SSCG Macro On/Off Control Bit in the SSCG Macro Control Register (D0/0x301B10)

Setting SSMCON (D0/0x301B10) to 1 causes the SSCG to start operating. When initially reset, SSMCON (D0/0x301B10) is initialized to 0, with the SSCG turned off (bypassed).

Notes:

- A stabilized clock must be supplied to the SSCG module when turning the SSCG on and off. The following shows the operation procedure.

To turn the SSCG on

1. Turn the PLL on.
2. Wait more than the PLL stabilization time.
3. Turn the SSCG on.

To turn the SSCG off

1. Turn the SSCG off.
2. Turn the PLL off.

- The SS modulation is effective only for the PLL output clock, and is not performed for other source clocks. When the PLL output clock is not used for the system clock, turn the SSCG off.

III.1.7.2 Setting SS Modulation Parameters

As described in “About spectrum spread (SS modulation)” above, it is necessary to set the upper-limit value of the maximum frequency change width and the frequency change cycle.

The maximum frequency change width should be set to the appropriate value according to the PLL output clock frequency as shown in Table III.1.7.2.1 using SSMCIDT[3:0] (D[11:8]/0x301B10). The maximum frequency change width will be about $\pm 2\%$ of the PLL output clock by setting the appropriate value.

* **SSMCIDT[3:0]**: SSCG Macro Maximum Frequency Change Width Setting Bits in the SSCG Macro Control Register (D[11:8]/0x301B10)

Table III.1.7.2.1 Maximum Frequency Change Width Settings

PLL output clock frequency f [MHz]	SSMCIDT3	SSMCIDT2	SSMCIDT1	SSMCIDT0
$f \leq 19.8$	1	1	1	1
$19.8 < f \leq 21.2$	1	1	1	0
$21.2 < f \leq 22.5$	1	1	0	1
$22.5 < f \leq 24.2$	1	1	0	0
$24.2 < f \leq 25.9$	1	0	1	1
$25.9 < f \leq 28.4$	1	0	1	0
$28.4 < f \leq 30.8$	1	0	0	1
$30.8 < f \leq 34.2$	1	0	0	0
$34.2 < f \leq 37.8$	0	1	1	1
$37.8 < f \leq 43.1$	0	1	1	0
$43.1 < f \leq 48.9$	0	1	0	1
$48.9 < f \leq 58.5$	0	1	0	0
$58.5 < f \leq 69.7$	0	0	1	1
$69.7 < f \leq 90.0$	0	0	1	0
–	0	0	0	1
–	0	0	0	0

(Default: 0b0000)

SSMCITM[3:0] (D[15:12]/0x301B10) is used to set the frequency change cycle. However, always set it to 0b0001.

* **SSMCITM[3:0]**: SSCG Macro Interval Timer Setting Bits in the SSCG Macro Control Register (D[15:12]/0x301B10)

- Notes:**
- SSMCIDT[3:0] (D[11:8]/0x301B10) must be set according to the PLL output clock frequency as shown in Table III.1.7.2.1. Using the SSCG with an improper setting may cause a malfunction of the IC.
 - When the PLL is off, the initial values and the written values cannot be read correctly from SSMCIDT[3:0] (D[11:8]/0x301B10) and SSMCITM[3:0] (D[15:12]/0x301B10) since the source clock is not supplied from the PLL (different values are read out). The correct values can be read out when the PLL is on.

III.1.8 Setting the Main System Clock (MCLK)

MCLK is the main system clock for the C33 PE Core and internal modules.

The source clock OSC for the system (selected by OSCSEL[1:0] (D[3:2]/0x301B08)) is divided by 1 or 2 by a clock frequency divider, generating two kinds of clocks. Select MCLK from those two clocks using MCLKDIV (D12/0x301B08).

* **MCLKDIV**: MCLK Clock Divider Select Bit in the System Clock Control Register (D12/0x301B08)

Table III.1.8.1 Selecting MCLK

MCLKDIV	MCLK
1	OSC•1/2
0	OSC•1/1

(Default: 0 = OSC•1/1)

When using the SDRAMC in double frequency mode (MCLK : SDRAM clock = 1 : 2), MCLK should be set to OSC•1/2 (OSC is used for the SDRAM clock).

MCLK can be selected at any time. However, up to 2 clock cycles are required before the clocks are actually changed after altering the register values.

III.1.9 Controlling Clock Supply

To reduce power consumption on the chip, a function is provided to turn off clock supply independently for each functional module.

III.1.9.1 MCLK Clock Supply to Each Module

Table III.1.9.1.1 lists the register bits used for on/off control of MCLK clock supply to the internal modules. The modules listed here have one controllable clock path, so they can be turned on/off using the corresponding control bit only. See Sections III.1.9.3 to III.1.9.9 for controlling the LCDC, SDRAMC, SRAMC, GPIO, EFSIO, USB, and RTC clocks.

Table III.1.9.1.1 MCLK Clock Supply Control Bits

Module	Control bit	Control register
DST RAM (Area 3 RAM)	DSTRAM_CKE (D3)	Gated Clock Control Register 0 (0x301B00)
16-bit timer 5	TM5_CKE (D18)	Gated Clock Control Register 1 (0x301B04)
16-bit timer 4	TM4_CKE (D17)	
16-bit timer 3	TM3_CKE (D16)	
16-bit timer 2	TM2_CKE (D15)	
16-bit timer 1	TM1_CKE (D14)	
16-bit timer 0	TM0_CKE (D13)	
Extended GPIO	EGPIO_CKE (D12)	
I ² S	I2S_CKE (D11)	
DCSIO	DCSIO_CKE (D10)	
Watchdog timer	WDT_CKE (D9)	
SPI	SPI_CKE (D6)	
CARD	CARD_CKE (D4)	
A/D converter	ADC_CKE (D3)	
ITC	ITC_CKE (D2)	
DMAC	DMA_CKE (D1)	

When initially reset, these control bits are set to 1 (on), with clocks supplied to each module. If any module is unused, set the corresponding control bit to 0, thus turning the clock for that module off.

- Notes:**
- These clocks do not stop in HALT mode. To stop supplying the clock in HALT mode, the control bit should be set to 0 before executing the halt instruction. All these clocks stop in SLEEP mode.
 - The clock supply to any module can only be stopped when the module is not operating or unused. If clock supply to any module is stopped when the module is being operated or used, the chip may hang.

III.1.9.2 Automatic Clock Control in HALT Mode

The clocks for the functions listed in Table III.1.9.2.1 can be automatically stopped in HALT mode by setting the control bits.

Table III.1.9.2.1 Clock Supply Control Bits to Disable in HALT Mode

Function	Control bit	Control register
SDRAMC CPU_AHB bus I/F	SDAPCPU_HCKE (D7)	Gated Clock Control Register 0 (0x301B00)
CPU_AHB bus control	CPUAHB_HCKE (D29)	Gated Clock Control Register 1 (0x301B04)
LCDC_AHB bus control	LCDCAHB_HCKE (D28)	
GPIO input/interrupt control	GPIONSTP_HCKE (D27)	
SRAMC	SRAMC_HCKE (D26)	
EFSIO baud rate timer	EFSIOBR_HCKE (D25)	
MISC registers	MISC_HCKE (D24)	

When initially reset, these control bits are set to 1 (on) to enable clock supply in HALT mode. If any clock is unused in HALT mode, set the corresponding control bit to 0 (off). The clock supply stops when the CPU enters HALT mode.

Note: All these clocks stop in SLEEP mode regardless how these control bits are set.

III.1.9.3 Clock Supply to the LCDC

The CMU provides the clock paths with a control bit shown below for the LCDC. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) LCDC AHB bus interface clock (LCDC_AHBIF_CLK)

The LCDC uses this clock (MCLK) to access IVRAM (internal VRAM) or an SDRAM (external VRAM). This clock is required for displaying a screen on the LCD panel. LCDCAHBIF_CKE (D2/0x301B00) is used for clock supply control (default: off).

* **LCDCAHBIF_CKE**: LCDC AHB Bus Interface Clock Control Bit in the Gated Clock Control Register 0 (D2/0x301B00)

(2) Control register clock (LCDC_SAPB_CLK)

This clock (MCLK) is used to control the LCDC registers located in area 6. This clock is required for accessing the LCDC registers and it can be stopped when not in use. LCDCSAPB_CKE (D1/0x301B00) is used for clock supply control (default: off).

* **LCDCSAPB_CKE**: LCDC SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 0 (D1/0x301B00)

(3) IVRAM arbiter clock (IVRAM_ARB_CLK)

This clock (MCLK) is used when the LCDC or CPU accesses IVRAM. When IVRAM is configured as AORAM accessed by the CPU only, the clock supply can be stopped. IVRAMARB_CKE (D19/0x301B04) is used for clock supply control (default: on).

* **IVRAMARB_CKE**: IVRAM Arbiter Clock Control Bit in the Gated Clock Control Register 1 (D19/0x301B04)

(4) LCD interface clock (LCDC_CLK)

This is the LCD interface clock (LCDC_CLK) generated by dividing the OSC3 clock. The frequency divider generates 16 kinds of clocks from OSC3•1/1 to OSC3•1/16. Select a divided clock according to the frame rate using LCDCDIV[3:0] (D[19:16]/0x301B08).

$$\text{Frame rate} = \frac{f_{\text{LCDC}}}{\text{HT} \times \text{VT}} \text{ [Hz]}$$

f_{LCDC}: LCDC_CLK frequency

HT: Horizontal total period (including non-display period) [pixels]

VT: Vertical total period (including non-display period) [pixels]

* **LCDCDIV[3:0]**: LCDC Clock Divider Select Bits in the System Clock Control Register (D[19:16]/0x301B08)

Table III.1.9.3.1 Selecting the LCDC Clock

LCDCDIV3	LCDCDIV2	LCDCDIV1	LCDCDIV0	LCDC_CLK
1	1	1	1	OSC3•1/16
1	1	1	0	OSC3•1/15
1	1	0	1	OSC3•1/14
1	1	0	0	OSC3•1/13
1	0	1	1	OSC3•1/12
1	0	1	0	OSC3•1/11
1	0	0	1	OSC3•1/10
1	0	0	0	OSC3•1/9
0	1	1	1	OSC3•1/8
0	1	1	0	OSC3•1/7
0	1	0	1	OSC3•1/6
0	1	0	0	OSC3•1/5
0	0	1	1	OSC3•1/4
0	0	1	0	OSC3•1/3
0	0	0	1	OSC3•1/2
0	0	0	0	OSC3•1/1

(Default: 0b0111 = OSC3•1/8)

LCDC_CKE (D0/0x301B00) is used for clock supply control (default: off).

* **LCDC_CKE**: LCDC Main Clock Control Bit in the Gated Clock Control Register 0 (D0/0x301B00)

Note: Make sure that the LCD interface clock supply is stopped (LCDC_CKE (D0/0x301B00) = 0) when changing the clock divide ratio using LCDCDIV[3:0] (D[19:16]/0x301B08).

(5) LCDC_AHB bus clock (LCDC_AHBBUS_CLK)

The LCDC_AHB bus clock (MCLK) is always supplied in normal operation. However, it can be automatically turned off in HALT mode (see Section III.1.9.2) by setting LCDCAHB_HCKE (D28/0x301B04) to 0 (default: on).

* **LCDCAHB_HCKE**: LCDC_AHB Bus Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D28/0x301B04)

Note: The LCDC clock supply cannot be stopped while the LCD displays a screen. Before the LCDC clock supply can be stopped, the LCDC must enter power save mode.

III.1.9.4 Clock Supply to the SDRAMC

The CMU provides the clock paths with a control bit shown below for the SDRAMC. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) SDRAMC AHB bus interface clocks (SDAPP_CPU_CLK, SDAPP_LCDC_CLK)

The SDRAMC uses these clocks (MCLK) for the CPU_AHB bus and LCDC_AHB bus interface. These clocks are required for accessing SDRAM and queue buffers. SDAPCPU_CKE (D6/0x301B00) and SDAPLCDC_CKE (D5/0x301B00) are respectively used for clock supply control (default: off).

* **SDAPCPU_CKE**: SDRAMC CPU APP Clock Control Bit in the Gated Clock Control Register 0 (D6/0x301B00)

* **SDAPLCDC_CKE**: SDRAMC LCDC APP Clock Control Bit in the Gated Clock Control Register 0 (D5/0x301B00)

Furthermore, the CPU_AHB bus clock (SDAPP_CPU_CLK) can be automatically turned off in HALT mode (see Section III.1.9.2) by setting SDAPCPU_HCKE (D7/0x301B00) to 0 (default: off).

* **SDAPCPU_HCKE**: SDRAMC CPU APP Clock Control (HALT) Bit in the Gated Clock Control Register 0 (D7/0x301B00)

(2) Control register clock (SDSAPB_CLK)

This clock (MCLK) is used to control the SDRAMC registers located in area 6. This clock is required for accessing the SDRAMC registers and it can be stopped when not in use. SDSAPB_CKE (D4/0x301B00) is used for clock supply control (default: off).

* **SDSAPB_CKE**: SDRAMC SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 0 (D4/0x301B00)

(3) SDRAM clock (SDIP_CLK)

This clock (OSC_W) is used in the SDRAM interface. By setting MCLK to $OSC \cdot 1/2$ ($= OSC_W \cdot 1/2$), the SDRAM bus can be driven in double frequency mode (SDRAM: 90 MHz max., CPU: 45 MHz). SDAPCPU_CKE (D6/0x301B00) or SDAPLCDC_CKE (D5/0x301B00) shown in (1) above is used for clock supply control (default: off).

III.1.9.5 Clock Supply to the SRAMC

The CMU provides the clock paths with a control bit shown below for the SRAMC. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) SRAMC clock (SRAMC_CLK)

The SRAMC controls the SAPB bus and external bus, so the SRAMC clock (MCLK) cannot be stopped while the IC is running. However, the SRAMC clock can be automatically turned off in HALT mode (see Section III.1.9.2) by setting SRAMC_HCKE (D26/0x301B04) to 0 (default: on).

* **SRAMC_HCKE**: SRAMC Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D26/0x301B04)

(2) Control register clock (SRAMSAPB_CLK)

This clock (MCLK) is used to control the SRAMC registers located in area 6. This clock is required for accessing the SRAMC registers and it can be stopped when not in use. SRAMSAPB_CKE (D7/0x301B04) is used for clock supply control (default: on).

- * **SRAMSAPB_CKE:** SRAMC SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 1 (D7/0x301B04)

III.1.9.6 Clock Supply to the GPIO

The CMU provides the clock paths with a control bit shown below for the GPIO. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) GPIO clock (PORT_CLK)

This clock (MCLK) is used for the GPIO circuit and is required for accessing the GPIO control registers. GPIO_CKE (D8/0x301B04) is used for clock supply control (default: on).

- * **GPIO_CKE:** GPIO Normal Clock Control Bit in the Gated Clock Control Register 1 (D8/0x301B04)

(2) GPIO no stop clock (PORT_NOSTOP_CLK)

This clock (MCLK) is used for reading input ports and generating input interrupts. This clock can be automatically turned off in HALT mode (see Section III.1.9.2) by setting GPIONSTP_HCKE (D27/0x301B04) to 0 (default: on).

- * **GPIONSTP_HCKE:** GPIO No Stop Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D27/0x301B04)

Note, however, that the GPIO no stop clock is required in HALT mode when using an input interrupt to cancel HALT mode.

III.1.9.7 Clock Supply to the EFSIO

The CMU provides the clock paths with a control bit shown below for the EFSIO. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) Control register clock (EFSIOSAPB_CLK)

This clock (MCLK) is used to control the EFSIO registers located in area 6. This clock is required for accessing the EFSIO registers and it can be stopped when not in use. EFSIOSAPB_CKE (D5/0x301B04) is used for clock supply control (default: on).

- * **EFSIOSAPB_CKE:** EFSIO SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 1 (D5/0x301B04)

(2) EFSIO baud rate timer clock (EFSIO_BAUDRATE_CLK)

This clock (MCLK) is used as the source clock for the baud rate timer in EFSIO. This clock can be automatically turned off in HALT mode (see Section III.1.9.2) by setting EFSIOBR_HCKE (D25/0x301B04) to 0 (default: on).

- * **EFSIOBR_HCKE:** EFSIO Baud Rate Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D25/0x301B04)

III.1.9.8 Clock Supply to the USB

The CMU provides the clock paths with the control bit shown below for the USB module. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) USB clock (USB_CLK)

This clock (OSC3 = 48 MHz) is used for the USB interface module. USB_CKE (D8/0x301B00) is used for clock supply control (default: off).

- * **USB_CKE:** USB IP 48 MHz Clock Control Bit in the Gated Clock Control Register 0 (D8/0x301B00)

(2) Control register clock (USBSAPB_CLK)

This clock (MCLK) is used to control the USB registers located in area 6. This clock is required for accessing the USB registers and it can be stopped when not in use. USBSAPB_CKE (D9/0x301B00) is used for clock supply control (default: off).

- * **USBSAPB_CKE**: USB SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 0 (D9/0x301B00)

III.1.9.9 Clock Supply to the RTC

The CMU provides the clock paths with a control bit shown below for the RTC. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) 32.768 kHz clock (OSC1)

This clock (OSC1 = 32.768 kHz) is used for timekeeping operations of the RTC. The clock supply can be stopped only when the RTC is not used and the OSC1 clock is not used as the system clock. To stop the OSC1 clock, turn the OSC1 oscillator circuit off by setting SOSC1 (D0/0x301B08) to 0 (default: on).

- * **SOSC1**: Low-speed Oscillation (OSC1) On/Off Control Bit in the System Clock Control Register (D0/0x301B08)

(2) Control register clock (RTCSAPB_CLK)

This clock (MCLK) is used to control the RTC registers located in area 6. This clock is required for accessing the RTC registers and it can be stopped when not in use. RTCSAPB_CKE (D0/0x301B04) is used for clock supply control (default: on).

- * **RTCSAPB_CKE**: RTC SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 1 (D0/0x301B04)

III.1.10 Setting the External Clock Output (CMU_CLK)

CMU_CLK is an external output clock for the external devices.

CMU_CLK can be selected from 11 clocks using CMU_CLKSEL[4:0] (D[28:24]/0x301B08).

* **CMU_CLKSEL[4:0]**: CMU_CLK Output Clock Source Select Bits in the System Clock Control Register (D[28:24]/0x301B08)

Table III.1.10.1 Selecting CMU_CLK

CMU_CLKSEL4	CMU_CLKSEL3	CMU_CLKSEL2	CMU_CLKSEL1	CMU_CLKSEL0	CMU_CLK
0	1	0	1	0	OSC3_DIV•1/32
0	1	0	0	1	OSC3_DIV•1/16
0	1	0	0	0	OSC3_DIV•1/8
0	0	1	1	1	OSC3_DIV•1/4
0	0	1	1	0	OSC3_DIV•1/2
0	0	1	0	1	OSC3_DIV•1/1
0	0	1	0	0	LCDC_CLK
0	0	0	1	1	MCLK
0	0	0	1	0	PLL
0	0	0	0	1	OSC1
0	0	0	0	0	OSC3
Other					Reserved

(Default: 0b00000 = OSC3)

CMU_CLK can be selected at any time. However, switching over the clocks creates hazards.

When CMU_CLK must be output to external devices, it is also necessary to select a port function. For details on how to control clock output and the port to be used, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Note: Settings other than those listed in Table III.1.10.1 are reserved for testing. Do not set undescribed values to CMU_CLKSEL[4:0] (D[28:24]/0x301B08) as undesired clocks may output.

III.1.11 Standby Modes

The S1C33E08 supports two standby modes: HALT and SLEEP. Power consumption on the chip can be greatly reduced by placing the CPU in one of these standby modes.

Moreover, the CPU must be placed in SLEEP mode before clock sources for the system (OSC3, OSC1, or PLL) are switched over.

III.1.11.1 HALT Mode

The CPU suspends program execution upon executing the halt instruction and enters HALT mode.

In HALT mode, the CPU and A0RAM (area 0 no-wait RAM) stop operating. Furthermore, the circuit for which the clock supply is automatically stopped in HALT mode (see Section III.1.9.2) stops operating. The other internal peripheral circuits remain in the state (idle or operating) held when the halt instruction was executed.

The CPU is released from HALT mode by initial reset, an NMI or other interrupt, or a forcible break from the debugger.

HALT mode is effective in reducing power consumption on the chip when running the CPU is unnecessary, such as when waiting for external input or responses from peripheral circuits. When the CPU is released from HALT mode by an interrupt, it enters a program executable state by trap processing and executes an interrupt handling routine for the interrupt generated. In trap processing of the CPU, the address for the instruction next to halt is saved to the stack as a return address from the interrupt handling routine, so that the reti instruction in the interrupt handling routine branches to the instruction next to halt.

The CPU is released from HALT mode when the interrupt controller (ITC) asserts the interrupt signal to be sent to the CPU. In other words, when a cause-of-interrupt flag of the interrupts that have been enabled by the interrupt enable bits in the ITC is set to 1, the CPU can be released from HALT mode even if the PSR is set to disable interrupts. However, in this case the CPU does not execute the interrupt handling routine.

The #NMI signal releases the CPU from HALT mode when it goes low level.

III.1.11.2 SLEEP Mode

The CPU suspends program execution upon executing the slp instruction and enters SLEEP mode. In SLEEP mode, the CPU stops operating and the CMU stops supplying a clock to each functional module. Therefore, all peripheral circuits (except the oscillator circuit and RTC) stop operating. Note that before the CMU actually stops clock output after initiating processing to enter SLEEP mode, up to 8 clock cycles of the source clock (OSC) then selected are required.

The CPU is reawaken from SLEEP mode by initial reset, RTC interrupt, NMI, or other interrupt from an external device (when WAKEUPWT = 1).

When the CPU is reawaken from SLEEP mode by an interrupt, it enters a program executable state by trap processing and executes an interrupt handling routine for the interrupt generated. In trap processing of the CPU, the address for the instruction next to slp is saved to the stack as a return address from the interrupt handling routine, so that the reti instruction in the interrupt handling routine branches to the instruction next to slp.

Cause-of-interrupt flags in the interrupt controller (ITC) cannot be set in SLEEP mode as the clock is not supplied to the ITC in SLEEP mode.

Therefore, when the clock is not supplied to the ITC, the interrupt signals from the interrupt sources that have been enabled to generate an interrupt are input to the CMU through the ITC and used to wake up the CPU from a standby mode. In this case, the cause-of-interrupt flag is set after the clock has started supplying to the ITC. The CPU can wake up from SLEEP mode by a cause of interrupt as described above even if the PSR is set to disable interrupts, note however, that the CPU does not execute the interrupt handling routine.

The #NMI signal releases the CPU from SLEEP mode when it goes low level.

- Notes:**
- In SLEEP mode, there is a time lag between input of an interrupt signal for wakeup and the start of the clock supply to the ITC, so a delay will occur until the interrupt controller (ITC) sets the cause-of-interrupt flag. Therefore, no interrupt will occur if the interrupt signal is deasserted before the clock is supplied to the ITC, as the cause-of-interrupt flag in the ITC is not set. Furthermore, additional time is needed for the CPU to accept the interrupt request from the ITC, the CPU may execute a few instructions that follow the slp instruction before it starts the interrupt processing. The same problem may occur when the CPU wakes up from SLEEP mode by NMI. No interrupt will occur if the #NMI signal is deasserted before the clock is supplied, as the NMI flag is not set.
 - Before setting the IC to SLEEP mode, the clock supply for the USB and LCDC must be disabled.

Stopping OSC3 oscillation and waiting for oscillation stabilization at wakeup

By default, neither the low-speed (OSC1) oscillator circuit nor the high-speed (OSC3) oscillator circuit stops operating when in SLEEP mode. OSC3 oscillation can be made to stop during SLEEP mode by setting OSC3OFF (D3/0x301B14).

* **OSC3OFF:** OSC3 Disable During SLEEP in the Clock Option Register (D3/0x301B14)

Setting OSC3OFF (D3/0x301B14) to 1 causes OSC3 oscillation to stop during SLEEP mode. In this case, the OSC3 oscillator circuit starts oscillating when the CPU is reawaken from SLEEP mode. However, since the CPU may operate erratically if it starts operating with the OSC3 or PLL clock before the oscillation stabilizes, an OSC oscillation start wait timer is provided to keep the CPU waiting a while before it starts operating. The wait time can be set by using OSCTM[7:0] (D[15:8]/0x301B14) and TMHSP (D2/0x301B14).

- * **OSCTM[7:0]:** OSC Oscillation Stabilization-Wait Timer in the Clock Option Register (D[15:8]/0x301B14)
- * **TMHSP:** Stabilization-Wait Timer High-Speed Mode Select Bit in the Clock Option Register (D2/0x301B14)

Table III.1.11.2.1 Oscillation Stabilization Wait Time at Wakeup

TMHSP	OSCTM[7:0]	Number of clocks	Time
1	0x0	0	0
	0x1	16	800 ns
	0x2	32	1.6 μs
	:	:	:
	0xFF	4080	0.204 ms
0	0x0	0	0
	0x1	8192	0.409 ms
	0x2	16384	0.819 ms
	:	:	:
	0xFF	2M	104.5 ms

(The time shown here is an example when operating with a 20 MHz OSC3.)

SLEEP control when clock sources are switched over

When the CPU reawakes from SLEEP mode, the clock sources (OSC3, OSC1, or PLL) also are switched over depending on how OSCSEL[1:0] (D[3:2]/0x301B08) is set. Before the clock sources can be switched over, the CPU must be placed once in SLEEP mode, then released. Therefore, a function is provided that automatically reawakes the CPU from SLEEP mode without using an interrupt, etc. To use this function, set WAKEUPWT (D0/0x301B14) to 0. (By default, it is set to 1.)

- * **OSCSEL[1:0]:** OSC Clock Select Bits in the System Clock Control Register (D[3:2]/0x301B08)
- * **WAKEUPWT:** Wakeup-Wait Function Enable Bit in the Clock Option Register (D0/0x301B14)

When the slp instruction is executed with WAKEUPWT (D0/0x301B14) set to 0, the CPU automatically reawakes from SLEEP mode several 10 clock cycles after that time, then restarts with the source clock selected by OSCSEL[1:0] (D[3:2]/0x301B08) after the oscillation stabilization time described above has elapsed.

The OSC oscillation start wait timer configured using OSCTM[7:0] (D[15:8]/0x301B14) and TMHSP (D2/0x301B14) is effective even if WAKEUPWT (D0/0x301B14) is 0. To restart the CPU in the shortest time possible, set OSCTM[7:0] (D[15:8]/0x301B14) to 0x0 and TMHSP (D2/0x301B14) to 1.

When WAKEUPWT (D0/0x301B14) is set to 1, the CPU is reawaken from SLEEP mode by initial reset, RTC interrupt, NMI, or other interrupt from an external device.

For details about clock switchover and SLEEP control procedures, see Section III.1.12, “Clock Setup Procedure.”

III.1.11.3 Precautions

Interrupt

The standby mode is released by an interrupt from the ITC, NMI, or reset. Note that the ITC must be configured so that the interrupt to be used for releasing the standby mode can be generated to the CPU. When the clock has not been supplied to the ITC, the interrupt signal from the interrupt source that has been enabled to interrupt is passed through the ITC and is input to the CMU. This signal is used to release the standby mode and to start supplying clocks. The ITC can operate with the supplied clock in HALT mode, so the cause-of-interrupt flag is set immediately after the interrupt source asserts the interrupt signal and the ITC requests an interrupt to the CPU without a delay. In SLEEP mode, the ITC will be able to set the cause-of-interrupt flag and to request an interrupt to the CPU after the CMU starts supplying the clock to the ITC. Therefore, the delay in the interrupt request to the CPU after waking up from SLEEP mode may cause the CPU to execute a few instructions that follows the slp instruction before the CPU executes the interrupt processing. Moreover, if the interrupt source deasserts the interrupt signal before the CMU starts supplying the clock to the ITC, an interrupt does not occur since the cause-of-interrupt flag is not set. The IE and IL[3:0] bits in the CPU's PSR register do not affect the releasing of standby mode by an interrupt. For example, by setting the ITC to enable the interrupt used for releasing and setting the IE bit to disable interrupts, the CPU can wake up from SLEEP mode without an interrupt processing.

Oscillator circuits

When OSC3 oscillation is set to stop during SLEEP mode, the OSC3 oscillator circuit starts oscillating upon exiting SLEEP mode. This is because the high-speed (OSC3) oscillator circuit requires a finite time before its oscillation stabilizes after starting operation. To restart the CPU using the OSC3 or PLL as the source clock, OSCTM[7:0] (D[15:8]/0x301B14) and TMHSP (D2/0x301B14) must be properly set so that the CPU starts operating after this oscillation stabilization time elapses. When using the PLL, note that the PLL requires a lock-in time (e.g., 200 μ s in the S1C33E08) after OSC3 oscillation has stabilized. The oscillation start time of the high-speed (OSC3) oscillator circuit varies with the device used, board patterns, and operating environment. Therefore, the set time above should have a sufficient allowance.

Bus and DMA

When in standby mode, the bus module stops operating after the bus cycle in progress is completed. All chip enable signals become inactive.

In HALT mode, the SRAMC is active, so the bus clock signals can be output and the DMA can also be run.

In SLEEP mode, the SRAMC is inactive, so no bus clock signals are output, nor is the DMA active.

Be sure to disable the HSDMA and IDMA before setting the chip in SLEEP mode (executing the slp instruction). HALT mode can be set even if the HSDMA and/or IDMA are enabled.

Switching over the clock sources

Use the automatic SLEEP cancellation function when executing the slp instruction for switching over the clock sources. When the SLEEP mode is cancelled, the OSC oscillation start wait timer that has been configured using OSCTM[7:0] (D[15:8]/0x301B14) starts operating with the clock source after switch over. Use the switched clock frequency for calculating the oscillation wait time.

Other

The status of the core CPU registers and input/output ports are retained even during standby mode. The contents of the control and data registers in internal peripheral circuits are also basically retained, but some contents are altered upon entering SLEEP mode. See the description of each peripheral circuit.

III.1.12 Clock Setup Procedure

This section describes the procedure for setting up clocks or altering clock settings.

When initially reset, the clocks are set to the following states:

OSC3 oscillator circuit: On
 PLL: Off
 OSC1 oscillator circuit: On
 System clock source: OSC3
 MCLK: OSC3•1/1
 CMU_CLK: OSC3•1/1

III.1.12.1 Changing the Clock Source from OSC3 to PLL

1. Clock Control Protect Register (0x301B24) = 0x96
 Disable write protection of the clock control registers.
2. PLLPOWR (D0/0x301B0C) = 0
 Turn off the PLL.
3. Setting PLL Control Register (0x301B0C)
 - PLLN[3:0] (D[7:4]) = 0b0000–0b1111
 Set the frequency multiplication rate of the PLL (x1 to x16).
 - PLLV[1:0] (D[3:2]) = 0b01–0b11
 Set the W value of the PLL (2, 4 or 8).
 - PLLVC[3:0] (D[15:12]) = 0b0001–0b1000
 Set the VCO Kv circuit constant.
 - PLLRS[3:0] (D[11:8]) = 0b1000 or 0b1010
 Set the LPF resistance value.
4. PLLPOWR (D0/0x301B0C) = 1
 Turn on the PLL.
5. OSCSEL[1:0] (D[3:2]/0x301B08) = 0b11
 Select the PLL for the clock source.
6. Setting the Clock Option Register (0x301B14)
 - OSCTM[7:0] (D[15:8]) = * * Set appropriate values so that the wait timer exceeds the stabilization time
 - OSC3OFF (D3) = 0 of the PLL output clock (e.g. 200 μs in the S1C33E08). Be aware that the
 - TMHSP (D2) = * wait timer operates with the PLL clock. For details about the PLL output
 - WAKEUPWPT (D0) = 0 stabilization time, see “Electrical Characteristics.”

This setting causes the CPU to automatically exit SLEEP mode and restart after the set time has passed without waiting for an interrupt.
7. Stop any peripheral circuits that are operating, except the RTC.
8. Execute the slp instruction.
 The chip enters SLEEP mode and the CMU temporarily stops clock output. The CPU automatically reawakens from SLEEP mode after the set time has passed from execution of the slp instruction, and restarts using the PLL as the clock source.
9. Newly setting the clock control registers again
 Newly alter the MCLK or CMU_CLK settings, and set other clock control registers again, as required.
10. Clock Control Protect Register (0x301B24) = other than 0x96
 Reenable write protection of the clock control registers.

III.1.12.2 Changing the Clock Source from PLL to OSC3, then Turning Off the PLL

1. Clock Control Protect Register (0x301B24) = 0x96
Disable write protection of the clock control registers.
2. OSCSEL[1:0] (D[3:2]/0x301B08) = 0b00
Select OSC3 for the clock source.
3. Setting the Clock Option Register (0x301B14)
 - OSCTM[7:0] (D[15:8]) = 0x0
 - OSC3OFF (D3) = 0
 - TMHSP (D2) = 1
 - WAKEUPWT (D0) = 0

This setting causes the CPU to automatically exit SLEEP mode and restart in the shortest time possible (several 10 clock cycles) without waiting an interrupt.
4. Stop any peripheral circuits that are operating, except the RTC.
5. Execute the slp instruction.
The chip enters SLEEP mode and the CMU temporarily stops clock output. The CPU automatically reawakes from SLEEP mode several 10 clock cycles after the slp instruction is executed, and restarts using OSC3 as the clock source.
6. PLLPOWR (D0/0x301B0C) = 0
Turn off the PLL.
7. Newly setting the clock control registers again
Newly alter the MCLK or CMU_CLK settings, and set other clock control registers again, as required.
8. Clock Control Protect Register (0x301B24) = other than 0x96
Reenable write protection of the clock control registers.

III.1.12.3 Changing the Clock Source from OSC3 or PLL to OSC1, then Turning Off OSC3 and PLL

1. Clock Control Protect Register (0x301B24) = 0x96
Disable write protection of the clock control registers.
2. SOSC1 (D0/0x301B08) = 1
Turn on the OSC1 oscillator circuit if turned off.
3. OSCSEL[1:0] (D[3:2]/0x301B08) = 0b01
Select OSC1 for the clock source.
4. Setting the Clock Option Register (0x301B14)
 - OSCTM[7:0] (D[15:8]) = * * Set appropriate values so that the wait timer exceeds the stabilization time of OSC1 oscillation (e.g., 3 seconds in the S1C33E08). Be aware that the wait timer operates with the OSC1 clock. For details about the OSC1
 - OSC3OFF (D3) = 0
 - TMHSP (D2) = *
 - WAKEUPWT (D0) = 0 oscillation start time, see “Electrical Characteristics.”
This setting causes the CPU to automatically exit SLEEP mode and restart after the set time has passed without waiting for an interrupt.
5. Stop any peripheral circuits that are operating.
6. Execute the slp instruction.
The chip enters SLEEP mode and the CMU temporarily stops clock output. The CPU automatically reawakens from SLEEP mode after the set time has passed from execution of the slp instruction, and restarts using OSC1 as the clock source.
7. PLLPOWR (D0/0x301B0C) = 0
Turn off the PLL.
8. SOSC3 (D1/0x301B08) = 0
Turn off the OSC3 oscillator circuit.
9. Newly setting the clock control registers again
Newly alter the MCLK or CMU_CLK settings, and set other clock control registers again, as required.
10. Clock Control Protect Register (0x301B24) = other than 0x96
Reenable write protection of the clock control registers.

III.1.12.4 Changing the Clock Source from OSC1 to OSC3

1. Clock Control Protect Register (0x301B24) = 0x96
Disable write protection of the clock control registers.
2. SOSC3 (D1/0x301B08) = 1
Turn on the OSC3 oscillator circuit if turned off.
3. OSCSEL[1:0] (D[3:2]/0x301B08) = 0b00
Select OSC3 for the clock source.
4. Setting the Clock Option Register (0x301B14)
 - OSCSTM[7:0] (D[15:8]) = * * Set appropriate values so that the wait timer exceeds the stabilization time of OSC3 oscillation (e.g., 25 ms in the S1C33E08). Be aware that the wait timer operates with the OSC3 clock. For details about the OSC3
 - OSC3OFF (D3) = 0
 - TMHSP (D2) = *
 - WAKEUPWT (D0) = 0 oscillation start time, see “Electrical Characteristics.”

This setting causes the CPU to automatically exit SLEEP mode and restart after the set time has passed without waiting for an interrupt.
5. Stop any peripheral circuits that are operating, except the RTC.
6. Execute the slp instruction.
The chip enters SLEEP mode and the CMU temporarily stops clock output. The CPU automatically reawakens from SLEEP mode after the set time has passed from execution of the slp instruction, and restarts using OSC3 as the clock source.
7. Newly setting the clock control registers again
Newly alter the MCLK or CMU_CLK settings, and set other clock control registers newly again, as required.
8. Clock Control Protect Register (0x301B24) = other than 0x96
Reenable write protection of the clock control registers.

III.1.12.5 Changing the Clock Source from OSC1 to PLL

1. Clock Control Protect Register (0x301B24) = 0x96
Disable write protection of the clock control registers.
2. SOSC3 (D1/0x301B08) = 1
Turn on the OSC3 oscillator circuit if turned off.
3. Wait until OSC3 oscillation stabilizes (only when OSC3 has been turned on; e.g., 25 ms in the S1C33E08). For details about the OSC3 oscillation start time, see “Electrical Characteristics.”
4. PLLPOWR (D0/0x301B0C) = 0
Turn off the PLL.
5. Setting PLL Control Register (0x301B0C)
 - PLLN[3:0] (D[7:4]) = 0b0000–0b1111
Set the frequency multiplication rate of the PLL (x1 to x16).
 - PLLV[1:0] (D[3:2]) = 0b01–0b11
Set the W value of the PLL (2, 4 or 8).
 - PLLVC[3:0] (D[15:12]) = 0b0001–0b1000
Set the VCO Kv circuit constant.
 - PLLRS[3:0] (D[11:8]) = 0b1000 or 0b1010
Set the LPF resistance value.
6. PLLPOWR (D0/0x301B0C) = 1
Turn on the PLL.
7. OSCSEL[1:0] (D[3:2]/0x301B08) = 0b11
Select PLL for the clock source.
8. Setting the Clock Option Register (0x301B14)
 - OSCTM[7:0] (D[15:8]) = * * Set appropriate values so that the wait timer exceeds the stabilization time of the PLL output clock (e.g. 200 μs in the S1C33E08). Be aware that the wait timer operates with the PLL clock. For details about the PLL output stabilization time, see “Electrical Characteristics.”
 - OSC3OFF (D3) = 0
 - TMHSP (D2) = *
 - WAKEUPWT (D0) = 0
This setting causes the CPU to automatically exit SLEEP mode and restart after the set time has passed without waiting for an interrupt.
9. Stop any peripheral circuits that are operating, except the RTC.
10. Execute the slp instruction.
The chip enters SLEEP mode and the CMU temporarily stops clock output. The CPU automatically reawakens from SLEEP mode after the set time has passed from execution of the slp instruction, and restarts using the PLL as the clock source.
11. Newly setting the clock control registers again
Newly alter the MCLK or CMU_CLK settings, and set other clock control registers again, as required.
12. Clock Control Protect Register (0x301B24) = other than 0x96
Reenable write protection of the clock control registers.

III.1.12.6 Turning Off OSC3 during SLEEP

To turn off OSC3 oscillation during SLEEP mode when operating with OSC3 or PLL as the clock source, follow the control procedure described below.

1. If the current clock source is PLL, it must be changed to OSC3 and the PLL turned off (see note below).
See Section III.1.12.2, “Changing the Clock Source from PLL to OSC3, then Turning Off the PLL,” for the procedure to change the clock source.
Stop PLL even if the current clock source is OSC3.
2. Clock Control Protect Register (0x301B24) = 0x96
Disable write protection of the clock control registers.
3. Setting the Clock Option Register (0x301B14)
 - OSCTM[7:0] (D[15:8]) and TMHSP (D2)
Set the wait time until the oscillation stabilizes after exiting SLEEP mode.
Example: TMHSP = 1, OSCTM[7:0] = 0x40 (wait time = about 26 ms when OSC3 = 20 MHz)
 - OSC3OFF (D3/0x301B14) = 1
Turn off OSC3 oscillation when in SLEEP mode.
 - WAKEUPWT (D0/0x301B14) = 1
Set the CPU to awake from SLEEP mode by using an RTC interrupt, NMI, or other interrupt from an external device.
4. Clock Control Protect Register (0x301B24) = other than 0x96
Reenable write protection of the clock control registers.
5. Stop any peripheral circuits that are operating, except the RTC.
6. Execute the slp instruction.
The chip enters SLEEP mode and the CMU temporarily stops clock output.

The CPU is reawaken from SLEEP mode by an RTC interrupt, forced break from the debugger, NMI, or other interrupt from an external device. After the set oscillation start wait time elapses, the CPU restarts using the same clock source (OSC3) that was selected before entering SLEEP mode.
7. If the application needs PLL as the clock source, change the clock source to PLL after the CPU wakes up with the OSC3 clock (see note below).
See Section III.1.12.1, “Changing the Clock Source from OSC3 to PLL,” for the procedure to change the clock source.

Note: To turn the OSC3 oscillation off in SLEEP mode, the conditions shown below must be satisfied before entering SLEEP mode and at wakeup from SLEEP mode.

- The CPU operates with OSC3 as the clock source.
- The PLL has been turned off.

If both OSC3 and PLL turn on at wakeup from SLEEP mode and the CPU starts operating with PLL as the clock source, the PLL operation may become unstable.

Therefore, to set PLL as the clock source, steps 1 and 7 above are required.

III.1.12.7 SLEEP Keeping Oscillation On (without Clock Change)

To enter SLEEP mode without a clock source change and turning off the oscillation, follow the control procedure described below. This is the control to reduce power consumption as much as possible by stopping the core and peripheral functions, with no restart time penalty.

1. Clock Control Protect Register (0x301B24) = 0x96
Disable write protection of the clock control registers.
2. Setting the Clock Option Register (0x301B14)
 - OSCTM[7:0] (D[15:8]) = 0x0
 - OSC3OFF (D3) = 0
 - TMHSP (D2) = 1
 - WAKEUPWT (D0) = 1

This setting causes the CPU to exit SLEEP mode using an RTC interrupt, NMI, or other interrupt from an external device, and to restart in the shortest time possible (several 10 clock cycles).
3. Clock Control Protect Register (0x301B24) = other than 0x96
Reenable write protection of the clock control registers.
4. Stop any peripheral circuits that are operating, except the RTC.
5. Execute the slp instruction.
The chip enters SLEEP mode and the CMU temporarily stops clock output.

The CPU is brought out of SLEEP mode by an RTC interrupt, forced break from the debugger, NMI, or other interrupt from an external device, and it restarts using the clock source selected with OSCSEL[1:0] (D[3:2]/0x301B08).

III.1.13 Power-Down Control

The amount of current consumed on the chip varies significantly with the CPU operation mode, operating clock frequency, and peripheral circuit to be operated. The following summarizes points on how to reduce power consumption on the chip.

1. Reducing the operating clock frequency as much as possible

The CMU allows one of the three available clock sources to be selected, and the system clock frequency to be set. (See III.1.4 to III.1.10.)

Reduce the operating clock frequency to as low a frequency as permitted for the intended processing content of the system.

- When the OSC1 clock can suffice, turn off the PLL and OSC3, and use only OSC1 to operate the system.
- When the OSC3 clock can suffice (although high-speed processing is needed), turn off the PLL and use OSC3 to operate the system.
- When the PLL is needed, use it with as small a frequency multiplication rate as possible.
- The divide ratios with which to generate MCLK from the source clock can be selected. Select the lowest divide ratios permitted for the intended processing content.
- Some peripheral circuits may use the prescaler, while others have an exclusive clock control function incorporated in the module. For details, refer to the description of each peripheral circuit.

2. Turning off unnecessary clock supply

The CMU allows clock supplies to be turned on or off independently for each functional module. (See III.1.9.)

- Turn off clock supplies for unused functional modules.
- Some peripheral circuits may use the prescaler, while others have an exclusive clock control function incorporated in the module. For details, refer to the description of each peripheral circuit.

3. Placing the CPU in standby mode

Place the CPU in standby mode by executing the halt or slp instruction as much as possible when, for example, waiting for key input. (See III.1.11.)

- Optimum power saving effects may be obtained by placing the CPU in SLEEP mode whenever OSC3 oscillation is turned off. In such case, however, basically only the RTC can be used without other peripheral circuits.
- To wake CPU quickly from SLEEP mode with no oscillation stabilization time inserted, enter SLEEP mode after setting OSC3 so that it does not stop in SLEEP mode. The core and peripheral circuits other than RTC and OSC3 cell unit enter power-down state.
- When peripheral circuits must be operated, use the halt instruction to place the CPU in HALT mode. In HALT mode, the CPU and AORAM all stop operating. Furthermore, the circuit for which the clock supply is automatically stopped in HALT mode (see Section III.1.9.2) stops operating.

4. Turning off unnecessary external port pull-ups

When input ports are driven to low level, their pull-up resistors consume some amount of current. Use the pull-up control register to turn off unnecessary pull-ups. However, care should be taken to prevent the input ports from becoming left open (floating state).

III.1.14 Details of Control Registers

Table III.1.14.1 List of CMU Registers

Address	Register name	Size	Function
0x00301B00	Gated Clock Control Register 0 (pCMU_GATEDCLK0)	32	Control clock supply for functional modules 0
0x00301B04	Gated Clock Control Register 1 (pCMU_GATEDCLK1)	32	Control clock supply for functional modules 1
0x00301B08	System Clock Control Register (pCMU_CLKCNTL)	32	Set system clock
0x00301B0C	PLL Control Register (pCMU_PLL)	32	Set PLL constant, on/off control
0x00301B10	SSCG Macro Control Register (pCMU_SSCG)	32	Control SSCG
0x00301B14	Clock Option Register (pCMU_OPT)	32	Set standby and wakeup conditions
0x00301B24	Clock Control Protect Register (pCMU_PROTECT)	32	Enable/disable write protection of clock control registers

The following describes each CMU control register.

The CMU control registers are mapped to the 32-bit device area at addresses 0x301B00 to 0x301B24, and can be accessed in units of words, half-words or bytes.

Note: The CMU registers (0x301B00–0x301B14) are write-protected. Before these register can be rewritten, write protection must be removed by writing data 0x96 to the Clock Control Protect Register (0x301B24). Note that since unnecessary rewrites to addresses 0x301B00–0x301B14 could lead to erratic system operation, the Clock Control Protect Register (0x301B24) should be set to other than 0x96 unless said CMU control registers must be rewritten.

0x301B00: Gated Clock Control Register 0 (pCMU_GATEDCLK0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Gated clock control register 0 (pCMU_GATEDCLK0) Protected	00301B00 (W)	D31-10	--	reserved		--	--	0 when being read.	
		D9	USBSAPB_CKE	USB SAPB I/F clock control	1 On	0 Off	0	R/W	
		D8	USB_CKE	USB IP 48 MHz clock control			0	R/W	
		D7	SDAPCPU_HCKE	SDRAMC CPU APP clock control (HALT)			0	R/W	
		D6	SDAPCPU_CKE	SDRAMC CPU APP clock control			0	R/W	
		D5	SDAPLCDC_CKE	SDRAMC LCDC APP clock control			0	R/W	
		D4	SDSAPB_CKE	SDRAMC SAPB I/F clock control			0	R/W	
		D3	DSTRAM_CKE	DST RAM clock control			1	R/W	
		D2	LCDCAHBIF_CKE	LCDC AHB I/F clock control			0	R/W	
		D1	LCDCSAPB_CKE	LCDC SAPB I/F clock control			0	R/W	
		D0	LCDC_CKE	LCDC main clock control			0	R/W	

D[31:10] Reserved**D9 USBSAPB_CKE: USB SAPB Interface Clock Control Bit**

Controls clock (MCLK) supply to the USB SAPB bus interface.

1 (R/W): On

0 (R/W): Off (default)

D8 USB_CKE: USB IP 48 MHz Clock Control Bit

Controls clock (OSC3 = 48 MHz) supply to the USB module.

1 (R/W): On

0 (R/W): Off (default)

D7 SDAPCPU_HCKE: SDRAMC CPU APP Clock Control (HALT) Bit

Controls clock (MCLK) supply to the SDRAMC CPU_AHB bus interface in HALT mode.

1 (R/W): On

0 (R/W): Off (default)

D6 SDAPCPU_CKE: SDRAMC CPU APP Clock Control Bit

Controls clock (MCLK) supply to the SDRAMC CPU_AHB bus interface.

1 (R/W): On

0 (R/W): Off (default)

D5 SDAPLCDC_CKE: SDRAMC LCDC APP Clock Control Bit

Controls clock (MCLK) supply to the SDRAMC LCDC_AHB bus interface.

1 (R/W): On

0 (R/W): Off (default)

D4 SDSAPB_CKE: SDRAMC SAPB Interface Clock Control Bit

Controls clock (MCLK) supply to the SDRAMC SAPB bus interface.

1 (R/W): On

0 (R/W): Off (default)

D3 DSTRAM_CKE: DST RAM Clock Control Bit

Controls clock (MCLK) supply to the DST RAM in area 3.

1 (R/W): On (default)

0 (R/W): Off

D2 LCDCAHBIF_CKE: LCDC AHB Bus Interface Clock Control Bit

Controls clock (MCLK) supply to the LCDC AHB bus interface.

1 (R/W): On

0 (R/W): Off (default)

III PERIPHERAL MODULES 1 (SYSTEM): CLOCK MANAGEMENT UNIT (CMU)

D1 LCDCSAPB_CKE: LCDC SAPB Interface Clock Control Bit

Controls clock (MCLK) supply to the LCDC SAPB bus interface.

1 (R/W): On

0 (R/W): Off (default)

D0 LCDC_CKE: LCDC Main Clock Control Bit

Controls clock (LCDC_CLK) supply to the LCDC module.

1 (R/W): On

0 (R/W): Off (default)

0x301B04: Gated Clock Control Register 1 (pCMU_GATEDCLK1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Gated clock control register 1 (pCMU_GATEDCLK1) Protected	00301B04 (W)	D31-30	–	reserved	–	–	–	0 when being read.	
		D29	CPUAHB_HCKE	CPU_AHB bus clk control (HALT)	1 On	0 Off	1 R/W		
		D28	LCDCAHB_HCKE	LCDC_AHB bus clk control (HALT)			1 R/W		
		D27	GPIONSTP_HCKE	GPIO no stop clock control (HALT)			1 R/W		
		D26	SRAMC_HCKE	SRAMC clock control (HALT)			1 R/W		
		D25	EFSIOBR_HCKE	EFSIO baud rate clk control (HALT)			1 R/W		
		D24	MISC_HCKE	Misc clock control (HALT)			1 R/W		
		D23-20	–	reserved	–	–	–	–	0 when being read.
		D19	IVRAMARB_CKE	IVRAM arbiter clock control	1 On	0 Off	1 R/W		
		D18	TM5_CKE	16-bit timer 5 clock control			1 R/W		
		D17	TM4_CKE	16-bit timer 4 clock control			1 R/W		
		D16	TM3_CKE	16-bit timer 3 clock control			1 R/W		
		D15	TM2_CKE	16-bit timer 2 clock control			1 R/W		
		D14	TM1_CKE	16-bit timer 1 clock control			1 R/W		
		D13	TM0_CKE	16-bit timer 0 clock control			1 R/W		
		D12	EGPIO_CKE	EGPIO clock control			1 R/W		
		D11	I2S_CKE	I ² S clock control			1 R/W		
		D10	DCSIO_CKE	DCSIO clock control			1 R/W		
		D9	WDT_CKE	Watchdog timer clock control			1 R/W		
		D8	GPIO_CKE	GPIO normal clock control			1 R/W		
		D7	SRAMSAPB_CKE	SRAMC SAPB I/F clock control			1 R/W		
		D6	SPI_CKE	SPI clock control			1 R/W		
		D5	EFSIOSAPB_CKE	EFSIO SAPB I/F clock control			1 R/W		
		D4	CARD_CKE	CARD I/F clock control			1 R/W		
		D3	ADC_CKE	ADC clock control			1 R/W		
		D2	ITC_CKE	ITC clock control			1 R/W		
		D1	DMA_CKE	DMAC clock control			1 R/W		
		D0	RTCSAPB_CKE	RTC SAPB I/F clock control			1 R/W		

D[31:30] Reserved

D29 CPUAHB_HCKE: CPU_AHB Bus Clock Control (HALT) Bit

Controls clock (MCLK) supply to the CPU_AHB bus in HALT mode.

1 (R/W): On (default)

0 (R/W): Off

D28 LCDCAHB_HCKE: LCDC_AHB Bus Clock Control (HALT) Bit

Controls clock (MCLK) supply to the LCDC_AHB bus in HALT mode.

1 (R/W): On (default)

0 (R/W): Off

D27 GPIONSTP_HCKE: GPIO No Stop Clock Control (HALT) Bit

Controls clock (MCLK) supply to the GPIO input/interrupt circuits in HALT mode.

1 (R/W): On (default)

0 (R/W): Off

D26 SRAMC_HCKE: SRAMC Clock Control (HALT) Bit

Controls clock (MCLK) supply to the SRAMC in HALT mode.

1 (R/W): On (default)

0 (R/W): Off

D25 EFSIOBR_HCKE: EFSIO Baud Rate Clock Control (HALT) Bit

Controls clock (MCLK) supply to the EFSIO baud rate timer in HALT mode.

1 (R/W): On (default)

0 (R/W): Off

D24 MISC_HCKE: Misc Clock Control (HALT) Bit

Controls clock (MCLK) supply to the Misc registers in HALT mode.

1 (R/W): On (default)

0 (R/W): Off

D[23:20] Reserved

D19 IVRAMARB_CKE: IVRAM Arbiter Clock Control Bit

Controls clock (MCLK) supply to the IVRAM arbiter.

1 (R/W): On (default)

0 (R/W): Off

D18 TM5_CKE: 16-bit Timer 5 Clock Control Bit

Controls clock (MCLK) supply to the 16-bit timer 5.

1 (R/W): On (default)

0 (R/W): Off

D17 TM4_CKE: 16-bit Timer 4 Clock Control Bit

Controls clock (MCLK) supply to the 16-bit timer 4.

1 (R/W): On (default)

0 (R/W): Off

D16 TM3_CKE: 16-bit Timer 3 Clock Control Bit

Controls clock (MCLK) supply to the 16-bit timer 3.

1 (R/W): On (default)

0 (R/W): Off

D15 TM2_CKE: 16-bit Timer 2 Clock Control Bit

Controls clock (MCLK) supply to the 16-bit timer 2.

1 (R/W): On (default)

0 (R/W): Off

D14 TM1_CKE: 16-bit Timer 1 Clock Control Bit

Controls clock (MCLK) supply to the 16-bit timer 1.

1 (R/W): On (default)

0 (R/W): Off

D13 TM0_CKE: 16-bit Timer 0 Clock Control Bit

Controls clock (MCLK) supply to the 16-bit timer 0.

1 (R/W): On (default)

0 (R/W): Off

D12 EGPIO_CKE: EGPIO Clock Control Bit

Controls clock (MCLK) supply to the EGPIO.

1 (R/W): On (default)

0 (R/W): Off

D11 I2S_CKE: I²S Clock Control Bit

Controls clock (MCLK) supply to the I²S interface.

1 (R/W): On (default)

0 (R/W): Off

D10 DCSIO_CKE: DCSIO Clock Control Bit

Controls clock (MCLK) supply to the DCSIO.

1 (R/W): On (default)

0 (R/W): Off

D9 WDT_CKE: Watchdog Timer Clock Control Bit

Controls clock (MCLK) supply to the watchdog timer.

1 (R/W): On (default)

0 (R/W): Off

- D8 GPIO_CKE: GPIO Normal Clock Control Bit**
Controls clock (MCLK) supply to the GPIO.
1 (R/W): On (default)
0 (R/W): Off
- D7 SRAMSAPB_CKE: SRAMC SAPB Interface Clock Control Bit**
Controls clock (MCLK) supply to the SRAMC SAPB interface.
1 (R/W): On (default)
0 (R/W): Off
- D6 SPI_CKE: SPI Clock Control Bit**
Controls clock (MCLK) supply to the SPI.
1 (R/W): On (default)
0 (R/W): Off
- D5 EFSIOSAPB_CKE: EFSIO SAPB Interface Clock Control Bit**
Controls clock (MCLK) supply to the EFSIO SAPB interface.
1 (R/W): On (default)
0 (R/W): Off
- D4 CARD_CKE: Card Interface Clock Control Bit**
Controls clock (MCLK) supply to the card interface.
1 (R/W): On (default)
0 (R/W): Off
- D3 ADC_CKE: A/D Converter Clock Control Bit**
Controls clock (MCLK) supply to the A/D converter.
1 (R/W): On (default)
0 (R/W): Off
- D2 ITC_CKE: ITC Clock Control Bit**
Controls clock (MCLK) supply to the ITC.
1 (R/W): On (default)
0 (R/W): Off
- D1 DMA_CKE: DMAC Clock Control Bit**
Controls clock (MCLK) supply to the DMA controllers.
1 (R/W): On (default)
0 (R/W): Off
- D0 RTCSAPB_CKE: RTC SAPB Interface Clock Control Bit**
Controls clock (MCLK) supply to the RTC SAPB interface.
1 (R/W): On (default)
0 (R/W): Off

0x301B08: System Clock Control Register (pCMU_CLKCNTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks					
System clock control register (pCMU_CLKCNTL) Protected	00301B08 (W)	D31-29	–	reserved	–	–	–	0 when being read.					
		D28	CMU_CLKSEL4	CMU_CLK output clock source selection	CMU_CLKSEL[4:0]	Clock source	0	R/W					
		D27	CMU_CLKSEL3		Other	reserved	0						
		D26	CMU_CLKSEL2		01010	OSC3_DIV*1/32	0						
		D25	CMU_CLKSEL1		01001	OSC3_DIV*1/16	0						
		D24	CMU_CLKSEL0		01000	OSC3_DIV*1/8	0						
					00111	OSC3_DIV*1/4							
					00110	OSC3_DIV*1/2							
					00101	OSC3_DIV*1/1							
					00100	LCDC_CLK							
					00011	MCLK							
					00010	PLL							
					00001	OSC1							
					00000	OSC3							
					D23	PLLINDIV3	PLL input clock source divider selection			PLLINDIV[3:0]	Divider	0	R/W
					D22	PLLINDIV2				Other	OSC3*1/8	1	
				D21	PLLINDIV1	1001		OSC3*1/10	1				
				D20	PLLINDIV0	1000		OSC3*1/9	1				
						0111		OSC3*1/8					
						0110		OSC3*1/7					
						0101		OSC3*1/6					
						0100		OSC3*1/5					
						0011		OSC3*1/4					
						0010		OSC3*1/3					
						0001	OSC3*1/2						
						0000	OSC3*1/1						
				D19	LCDCDIV3	LCDC clock divider selection	LCDCDIV[3:0]	Divider	0	R/W			
				D18	LCDCDIV2		1111	OSC3*1/16	1				
				D17	LCDCDIV1		1110	OSC3*1/15	1				
				D16	LCDCDIV0		1101	OSC3*1/14	1				
							1100	OSC3*1/13					
							1011	OSC3*1/12					
				1010	OSC3*1/11								
				1001	OSC3*1/10								
				1000	OSC3*1/9								
				0111	OSC3*1/8								
				0110	OSC3*1/7								
				0101	OSC3*1/6								
				0100	OSC3*1/5								
				0011	OSC3*1/4								
				0010	OSC3*1/3								
				0001	OSC3*1/2								
				0000	OSC3*1/1								
		D15-13	–	reserved	–	–	–	0 when being read.					
		D12	MCLKDIV	MCLK clock divider selection	1 1/2	0 1/1	0	R/W					
		D11	–	reserved	–	–	–	0 when being read.					
		D10	OSC3DIV2	OSC3 clock divider selection	OSC3DIV[2:0]	Divider	0	R/W					
		D9	OSC3DIV1		111	OSC3*1/1	0						
		D8	OSC3DIV0		110	OSC3*1/1	0						
					101	OSC3*1/32							
					100	OSC3*1/16							
					011	OSC3*1/8							
					010	OSC3*1/4							
					001	OSC3*1/2							
				000	OSC3*1/1								
		D7-4	–	reserved	–	–	–	0 when being read.					
		D3	OSCSEL1	OSC clock selection	OSCSEL[1:0]	Clock source	0	R/W					
		D2	OSCSEL0		11	PLL	0						
					10	OSC3							
					01	OSC1							
					00	OSC3							
		D1	SOSC3	High-speed oscillation (OSC3) On/Off	1 On	0 Off	1	R/W					
		D0	SOSC1	Low-speed oscillation (OSC1) On/Off	1 On	0 Off	1	R/W					

D[31:29] Reserved

D[28:24] **CMU_CLKSEL[4:0]: CMU_CLK Output Clock Source Select Bits**

CMU_CLK is the clock for the external bus. It can be selected from the 11 clocks listed in Table III.1.14.2.

Table III.1.14.2 Selecting CMU_CLK

CMU_CLKSEL4	CMU_CLKSEL3	CMU_CLKSEL2	CMU_CLKSEL1	CMU_CLKSEL0	CMU_CLK
0	1	0	1	0	OSC3_DIV•1/32
0	1	0	0	1	OSC3_DIV•1/16
0	1	0	0	0	OSC3_DIV•1/8
0	0	1	1	1	OSC3_DIV•1/4
0	0	1	1	0	OSC3_DIV•1/2
0	0	1	0	1	OSC3_DIV•1/1
0	0	1	0	0	LCDC_CLK
0	0	0	1	1	MCLK
0	0	0	1	0	PLL
0	0	0	0	1	OSC1
0	0	0	0	0	OSC3
Other					Reserved

(Default: 0b00000)

CMU_CLK can be selected at any time. However, switching over the clocks creates hazards. When CMU_CLK must be output to external devices, it is also necessary to select a port function. For details on how to control clock output and about the port to be used, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Note: Other settings than that listed in Table III.1.14.2 are reserved for testing. Do not set undescribed values to CMU_CLKSEL[4:0] as undesired clocks may output.

D[23:20] **PLLINDIV[3:0]: PLL Input Clock Source Divider Select Bits**

PLLINDIV[3:0] is used to select the PLL input clock from among 10 kinds of OSC3 divided clocks.

Table III.1.14.3 Selecting the PLL Input Clock

PLLINDIV3	PLLINDIV2	PLLINDIV1	PLLINDIV0	PLL input clock
1	0	0	1	OSC3•1/10
1	0	0	0	OSC3•1/9
0	1	1	1	OSC3•1/8
0	1	1	0	OSC3•1/7
0	1	0	1	OSC3•1/6
0	1	0	0	OSC3•1/5
0	0	1	1	OSC3•1/4
0	0	1	0	OSC3•1/3
0	0	0	1	OSC3•1/2
0	0	0	0	OSC3•1/1
Other				OSC3•1/8

(Default: 0b0111)

Note: The PLL input clock can only be selected when the PLL is turned off (PLLPOWER (D0/0x301B0C) = 0) and the clock source is other than the PLL (OSCSEL[1:0] (D[3:2]) = 0–2). If the PLL input clock is changed while the system is operating with the PLL clock, the system may operate erratically.

D[19:16] LCDCDIV[3:0]: LCDC Clock Divider Select Bits

LCDCDIV[3:0] is used to select the LCD interface clock (LCDC_CLK) from among 16 kinds of OSC3 divided clocks. Select a divided clock according to the frame rate.

$$\text{Frame rate} = \frac{f_{\text{LCDC}}}{\text{HT} \times \text{VT}} \text{ [Hz]}$$

f_{LCDC}: LCDC_CLK frequency

HT: Horizontal total period (including non-display period) [pixels]

VT: Vertical total period (including non-display period) [pixels]

Table III.1.14.4 Selecting the LCDC Clock

LCDCDIV3	LCDCDIV2	LCDCDIV1	LCDCDIV0	LCDC_CLK
1	1	1	1	OSC3•1/16
1	1	1	0	OSC3•1/15
1	1	0	1	OSC3•1/14
1	1	0	0	OSC3•1/13
1	0	1	1	OSC3•1/12
1	0	1	0	OSC3•1/11
1	0	0	1	OSC3•1/10
1	0	0	0	OSC3•1/9
0	1	1	1	OSC3•1/8
0	1	1	0	OSC3•1/7
0	1	0	1	OSC3•1/6
0	1	0	0	OSC3•1/5
0	0	1	1	OSC3•1/4
0	0	1	0	OSC3•1/3
0	0	0	1	OSC3•1/2
0	0	0	0	OSC3•1/1

(Default: 0b0111)

D[15:13] Reserved

D12 MCLKDIV: MCLK Clock Divider Select Bit

Selects the main system clock.

1 (R/W): OSC•1/2

0 (R/W): OSC•1/1 (default)

MCLK is the main system clock for the S1C33E08. It is derived from the system's source clock OSC (selected using OSCSEL[1:0] (D[3:2])) by dividing its frequency by 1 or 2.

When using the SDRAM in double frequency mode (MCLK max. = 45 MHz, SDRAM clock max. = 90 MHz), MCLKDIV should be set to 1.

D11 Reserved

D[10:8] OSC3DIV[2:0]: OSC3 Clock Divider Select Bits

OSC3DIV[2:0] is used to select the system clock frequency when OSC3 is selected as the system clock source. It is derived from the OSC3 clock by dividing its frequency by a given value. Use OSC3DIV[2:0] to select this clock divide ratio.

Table III.1.14.5 Selecting an OSC3 Divided Clock

OSC3DIV2	OSC3DIV1	OSC3DIV0	CMU_CLK
1	1	1	OSC3•1/1
1	1	0	OSC3•1/1
1	0	1	OSC3•1/32
1	0	0	OSC3•1/16
0	1	1	OSC3•1/8
0	1	0	OSC3•1/4
0	0	1	OSC3•1/2
0	0	0	OSC3•1/1

(Default: 0b000)

A divided clock can be selected at any time. However, up to 32 OSC3 clock cycles are required before the clocks are actually changed after altering the register values.

D[7:4] **Reserved**

D[3:2] **OSCSEL[1:0]: OSC Clock Select Bits**

Selects the clock source for the system (OSC).

Table III.1.14.6 Selecting the System Clock Source

OSCSEL1	OSCSEL0	Clock source
1	1	PLL
1	0	OSC3
0	1	OSC1
0	0	OSC3

(Default: 0b00)

The clock sources changed here are not switched over immediately, but are actually switched over upon returning from SLEEP mode. Therefore, the CPU must be placed in SLEEP mode after setting up OSCSEL[1:0].

Note: When clock sources are changed, the clock control registers must be set so that the CMU is supplied with a clock from the selected clock source upon returning from SLEEP mode immediately after the change. Otherwise, the chip does not restart after return from SLEEP mode.

D1 **SOSC3: High-speed Oscillation (OSC3) On/Off Bit**

Turns the OSC3 oscillator circuit on or off.

1 (R/W): On (default)

0 (R/W): Off

D0 **SOSC1: Low-speed Oscillation (OSC1) On/Off Bit**

Turns the OSC1 oscillator circuit on or off.

1 (R/W): On (default)

0 (R/W): Off

Note: When SOSC3 (D1) or SOSC1 (D0) is set from 0 to 1 for initiating oscillation by the oscillator, a finite time is required until the oscillation stabilizes. To prevent erratic operation, do not use the oscillator-derived clock until the oscillation start time stipulated in the electrical characteristics table elapses.

0x301B0C: PLL Control Register (pCMU_PLL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
PLL control register (pCMU_PLL) Protected	00301B0C (W)	D31-24	–	reserved		–	–	0 when being read.				
		D23	PLLCS1	PLL LPF capacitance setup	Fixed at "00" (default)	0	R/W					
		D22	PLLCS0				0					
		D21	PLLBYP	PLL bypass mode setup	Fixed at "0" (default)	0	R/W					
		D20	PLLCP4	PLL charge pump current setup	Fixed at "10000" (default)		1	R/W				
		D19	PLLCP3						0			
		D18	PLLCP2						0			
		D17	PLLCP1						0			
		D16	PLLCP0						0			
		D15	PLLVC3	PLL VCO Kv setup		PLLVC[3:0]	fvco [MHz]	0	R/W			
		D14	PLLVC2							1000	360 < fvco ≤ 400	0
		D13	PLLVC1							0111	320 < fvco ≤ 360	0
		D12	PLLVC0							0110	280 < fvco ≤ 320	1
										0101	240 < fvco ≤ 280	
				0100	200 < fvco ≤ 240							
				0011	160 < fvco ≤ 200							
				0010	120 < fvco ≤ 160							
				0001	100 ≤ fvco ≤ 120							
				Other	Not allowed							
		D11	PLLRS3	PLL LPF resistance setup		PLLRS[3:0]	fREFCK [MHz]	1	R/W			
D10	PLLRS2	1010	5 ≤ fREFCK < 20							0		
D9	PLLRS1	1000	20 ≤ fREFCK ≤ 150							0		
D8	PLLRS0	Other	Not allowed							0		
D7	PLLN3	PLL multiplication rate setup		PLLN[3:0]	Multiplication rate	0	R/W					
D6	PLLN2							1111	x16	0		
D5	PLLN1							1110	x15	0		
D4	PLLN0							:	:	0		
		0001	x2									
		0000	x1									
D3	PLLV1	PLL V-divider setup		PLLV[1:0]	W	0	R/W					
D2	PLLV0							11	8	1		
								10	4			
								01	2			
		00	Not allowed									
D1	–	reserved			–	–	0 when being read.					
D0	PLLPOWR	PLL on/off control		1 On	0 Off	0	R/W					

Note: When D[23:2] in this register must be altered, turn off the PLL (PLLPOWR (D0) = 0) before changing the bits.

D[31:24] Reserved

D[23:22] PLLCS[1:0]: PLL LPF Capacitance Setup Bits

Sets the LPF capacitance value (CS value). (Default: 0b00)

This bit should be left as at initial reset, without altering its settings while in use.

D21 PLLBYP: PLL Bypass Mode Setup Bit

Sets PLL bypass mode. (Default: 0)

This bit should be left as at initial reset, without altering its settings while in use.

D[20:16] PLLCP[4:0]: PLL Charge Pump Current Setup Bits

Sets the charge pump current value (CP value). (Default: 0b10000)

This bit should be left as at initial reset, without altering its settings while in use.

D[15:12] PLLVC[3:0]: PLL VCO Kv Setup Bits

Sets the VCO Kv circuit constant (VC value) according to the range of fvco frequencies obtained by <output clock frequency × W>.

Table III.1.14.7 Settings of the VC Value

PLLVC3	PLLVC2	PLLVC1	PLLVC0	f _{vco} [MHz]
1	0	0	0	360 < f _{vco} ≤ 400
0	1	1	1	320 < f _{vco} ≤ 360
0	1	1	0	280 < f _{vco} ≤ 320
0	1	0	1	240 < f _{vco} ≤ 280
0	1	0	0	200 < f _{vco} ≤ 240
0	0	1	1	160 < f _{vco} ≤ 200
0	0	1	0	120 < f _{vco} ≤ 160
0	0	0	1	100 ≤ f _{vco} ≤ 120
Other				Not allowed

(Default: 0b0001)

D[11:8] PLLRS[3:0]: PLL LPF Resistance Setup Bits

Sets the LPF resistance value of the PLL (RS value) according to the input clock (OSC3) frequency.

Table III.1.14.8 Settings of the RS Value

PLLRS3	PLLRS2	PLLRS1	PLLRS0	f _{REFCK} [MHz]
1	0	1	0	5 ≤ f _{REFCK} < 20
1	0	0	0	20 ≤ f _{REFCK} ≤ 150
Other				Not allowed

(Default: 0b1000)

D[7:4] PLLN[3:0]: PLL Multiplication Rate Setup Bits

Sets the frequency multiplication rate of the PLL. (Default: 0b0000 = ×1)

PLL frequency multiplication rate = PLLN[3:0] + 1 (×1 to ×16)

Note: The frequency multiplication rate must be set so that the PLL output clock frequency does not exceed the upper-limit operating clock frequency. For the multiplication rates that can be set and the range of the output clock frequency, see “Electrical Characteristics.”

D[3:2] PLLV[1:0]: PLL V-Divider Setup Bits

 Sets the W value so that the f_{vco} frequency obtained by <output clock frequency × W> falls within the range of 100 to 400 MHz.

Table III.1.14.9 Settings of the W Value

PLLV1	PLLV0	W
1	1	8
1	0	4
0	1	2
0	0	Not allowed

(Default: 0b01)

D1 Reserved
D0 PLLPOWR: PLL On/Off Control Bit

Turns the PLL on or off.

1 (R/W): On

0 (R/W): Off (default)

Up to 200 μs is required before the output clock of the PLL stabilizes after PLLPOWR is set to 1. Provide this wait time in a program before changing the clock source for the system to the PLL.

When not using the PLL, turn off the PLL (power-down mode) to reduce the amount of current consumed on the chip.

Table III.1.14.10 Example PLL Settings

PLL input clock	PLL output clock	PLLN[3:0]	PLLV[1:0]	PLLVC[3:0]	PLLRS[3:0]
6 MHz	90 MHz	x15, 0b1110	0b01	0b0011	0b1010
	60 MHz	x10, 0b1001	0b01	0b0001	0b1010
10 MHz	80 MHz	x8, 0b0111	0b01	0b0010	0b1010
	40 MHz	x4, 0b0011	0b10	0b0010	0b1010
20 MHz	80 MHz	x4, 0b0011	0b01	0b0010	0b1000
	40 MHz	x2, 0b0001	0b10	0b0010	0b1000
45 MHz	90 MHz	x2, 0b0001	0b01	0b0011	0b1000

0x301B10: SSCG Macro Control Register (pCMU_SSCG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
SSCG macro control register (pCMU_SSCG)	00301B10 (W)	D31-16	–	reserved		–	–	0 when being read.	
		D15	SSMCITM3	SSCG macro interval timer (ITM) setting	0 to 0xF	1	R/W		
		D14	SSMCITM2				1		
		D13	SSMCITM1				1		
		D12	SSMCITM0				1		
		D11	SSMCIDT3	SSCG macro maximum frequency change width setting	0 to 0xF	0	R/W		
		D10	SSMCIDT2				0		
		D9	SSMCIDT1				0		
		D8	SSMCIDT0				0		
		D7-1	–	reserved		–	–	–	0 when being read.
D0	SSMCON	SSCG macro On/Off		1 On 0 Off	0	R/W			

Note: When the PLL is off, the initial values and the written values cannot be read correctly from SSMCITM[3:0] (D[11:8]) and SSMCITM[3:0] (D[15:12]) since the source clock is not supplied from the PLL (different values are read out). The correct values can be read out when the PLL is on.

D[31:16] Reserved

D[15:12] SSMCITM[3:0]: SSCG Macro Interval Timer Setting Bits

These bits set the frequency change cycle in SS modulation of the SSCG. (See Section III.1.7, “Control of the SSCG.”)

Always set these bits to 0b0001. (Default: 0b1111)

D[11:8] SSMCIDT[3:0]: SSCG Macro Maximum Frequency Change Width Setting Bits

These bits set the maximum frequency change width in SS modulation of the SSCG. (See Section III.1.7, “Control of the SSCG.”)

Table III.1.14.11 Maximum Frequency Change Width Settings

PLL output clock frequency f [MHz]	SSMCIDT3	SSMCIDT2	SSMCIDT1	SSMCIDT0
$f \leq 19.8$	1	1	1	1
$19.8 < f \leq 21.2$	1	1	1	0
$21.2 < f \leq 22.5$	1	1	0	1
$22.5 < f \leq 24.2$	1	1	0	0
$24.2 < f \leq 25.9$	1	0	1	1
$25.9 < f \leq 28.4$	1	0	1	0
$28.4 < f \leq 30.8$	1	0	0	1
$30.8 < f \leq 34.2$	1	0	0	0
$34.2 < f \leq 37.8$	0	1	1	1
$37.8 < f \leq 43.1$	0	1	1	0
$43.1 < f \leq 48.9$	0	1	0	1
$48.9 < f \leq 58.5$	0	1	0	0
$58.5 < f \leq 69.7$	0	0	1	1
$69.7 < f \leq 90.0$	0	0	1	0
–	0	0	0	1
–	0	0	0	0

(Default: 0b0000)

Note: SSMCIDT[3:0] must be set according to the PLL output clock frequency as shown in Table III.1.14.11. Using the SSCG with an improper setting may cause a malfunction of the IC.

D[7:1] Reserved

D0 SSMCON: SSCG Macro On/Off Control Bit

This bit turns the SSCG on or off.

1 (R/W): On

0 (R/W): Off (default)

Setting this bit to 1 causes the SSCG to start operating. Setting this bit to 0 causes the SSCG to stop, so that the clock generator bypasses the SSCG.

0x301B14: Clock Option Register (pCMU_OPT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Clock option register (pCMU_OPT) Protected	00301B14 (W)	D31-16	–	reserved		–	–	0 when being read.	
		D15	OSCTM7	OSC oscillation stabilization-wait timer	0 to 255	0	R/W		
		D14	OSCTM6			0			
		D13	OSCTM5			0			
		D12	OSCTM4			0			
		D11	OSCTM3			0			
		D10	OSCTM2			0			
		D9	OSCTM1			0			
		D8	OSCTM0			0			
		D7-4	–	reserved		–	–	0 when being read.	
		D3	OSC3OFF	OSC3 disable during SLEEP	1 Stop	0 Run	0	R/W	
		D2	TMHSP	Wait-timer high-speed mode	1 High speed	0 Normal	0	R/W	
		D1	–	reserved		–	–	–	0 when being read.
D0	WAKEUPWT	Wakeup-wait function enable	1 Wait interrupt	0 No wait	0	R/W			

D[31:16] Reserved

D[15:8] OSCTM[7:0]: OSC Oscillation Stabilization-Wait Timer

Sets an oscillation stabilization wait time during which the CPU is kept waiting before it starts operating upon returning from SLEEP mode. This wait time can be set in increments of 16 OSC clock cycles when TMHSP (D2) = 1, or 8,192 clock cycles when TMHSP (D2) = 0. (Default: 0b00 = no wait time)

Table III.1.14.12 Oscillation Stabilization Wait Time at Wakeup

TMHSP	OSCTM[7:0]	Number of clocks	Time
1	0x0	0	0
	0x1	16	800 ns
	0x2	32	1.6 μs
	:	:	:
	0xFF	4080	0.204 ms
0	0x0	0	0
	0x1	8192	0.409 ms
	0x2	16384	0.819 ms
	:	:	:
	0xFF	2M	104.5 ms

(The time shown here is an example when operating with a 20 MHz OSC3.)

When the OSC3 oscillation is to be turned off during SLEEP mode, make sure the wait time set by these bits is equal to or greater than the OSC3 oscillation start time stipulated in the electrical characteristics table.

Note: The OSC oscillation start wait timer operates with the operating clock activated after the SLEEP mode is released. Therefore, use the switched clock frequency for calculating the oscillation wait time to be set to OSCTM[7:0] when executing the slp instruction for switching over the clock sources.

D[7:4] Reserved

D3 OSC3OFF: OSC3 Disable During SLEEP

Selects whether to turn off the OSC3 oscillator circuit during SLEEP mode.

1 (R/W): Stop

0 (R/W): Operating (default)

Continue operating OSC3 when entering SLEEP mode to switch over the clock sources (OSC), or turn it off when entering SLEEP mode for power-down purposes.

D2 TMHSP: Stabilization-Wait Timer High-Speed Mode Select Bit

Sets count mode for the oscillation stabilization wait timer (OSCTM[7:0]).

1 (R/W): High-speed mode

0 (R/W): Normal mode (default)

The oscillation stabilization wait timer counts from 0 to 2M in units of 8,192 OSC clock cycles during normal mode, or from 0 to 4,080 in units of 16 OSC clock cycles during high-speed mode. Select either mode in which the OSC3 oscillation start time can be secured with the OSC frequency used.

D1 Reserved

D0 WAKEUPWT: Wakeup-Wait Function Enable Bit

Enables the SLEEP mode wakeup-wait function used for switching over the clocks.

1 (R/W): Wait an interrupt

0 (R/W): No wait (default)

When the `slp` instruction is executed while WAKEUPWT is set to 0, the CPU automatically reawakes from SLEEP mode several 10 clock cycles after instruction execution, and restarts with the source clock selected by OSCSEL[1:0] (D[3:2]/0x301B08). Since even in this case the oscillation stabilization wait time set by OSCTM[7:0] (D[15:8]) is effective, OSCTM[7:0] should be set to 0x0 when clocks must be switched over in the shortest time possible.

When WAKEUPWT is set to 1, the CPU can only be reawaken from SLEEP mode by an interrupt such as initial reset, RTC interrupt, forced break from the debugger, NMI, and other interrupt from an external source.

0x301B24: Clock Control Protect Register (pCMU_PROTECT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock control protect register (pCMU_PROTECT)	00301B24 (W)	D31–8	–	reserved		–	–	0 when being read.
		D7	CLGP7	Clock control register protect flag	Writing 10010110 (0x96) removes the write protection of the clock control registers (0x301B00–0x301B14). 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		

D[31:8] Reserved**D[7:0] CLGP[7:0]: Clock Control Register Protect Flag**

Enables/disables write protection of the clock control registers (0x301B00–0x301B14).

0x96 (R/W): Disable write protection

Other than 0x96 (R/W): Write-protect the register (default: 0x0)

Before altering any clock control register, write data 0x96 to the register to disable write protection. If this register is set to other than 0x96, even if an attempt is made to alter any clock control register by executing a write instruction, the content of said register will not be altered even though the instruction may have been executed without a problem. Once this register is set to 0x96, the clock control registers can be rewritten any number of times until being reset to other than 0x96. When rewriting the clock control registers has finished, this register should be set to other than 0x96 to prevent accidental writing to the clock control registers.

III.1.15 Precautions

Precautions regarding clock control

- The clock control registers (0x301B00–0x301B14) are write-protected. Before these registers can be rewritten, write protection must be removed by writing data 0x96 to the Clock Control Protect Register (0x301B24). Once write protection is removed, the clock control registers can be written to any number of times until the protect register is reset to other than 0x96. Note that since unnecessary rewriting of the clock control registers could lead to erratic system operation, the Clock Control Protect Register (0x301B24) should be set to other than 0x96 unless the clock control registers must be rewritten.
- When clock sources are changed, the clock control registers must be set so that the CMU is supplied with a clock from the selected clock source upon returning from SLEEP mode immediately after the change. Otherwise, the chip may not restart after return from SLEEP mode.
Furthermore, note that the timer, which generates an oscillation stabilization wait time after the SLEEP mode is released, operates with the clock after switching over. Be sure to use the correct clock frequency for calculating the wait time to be set to OSC_{TM}[7:0] (D[15:8]/0x301B14) and TMHSP (D2/0x301B14).
- When SOSC3 (D1/0x301B08) or SOSC1 (D0/0x301B08) is set from 0 to 1 for initiating oscillation by the oscillator, a finite time is required until the oscillation stabilizes (e.g., 25 ms for OSC3 and 3 seconds for OSC1 in the S1C33E08). To prevent erratic operation, do not use the oscillator-derived clock until the oscillation start time stipulated in the electrical characteristics table elapses.
- Immediately after the PLL is started by setting PLLPOWR (D0/0x301B0C) to 1, an output clock stabilization wait time is required (e.g., 200 μs in the S1C33E08). When the clock source for the system is switched over to the PLL, allow for this wait time after the PLL has turned on.
- The frequency multiplication rate of the PLL that can be set depends on the upper-limit operating clock frequency (90 MHz) and the OSC3 oscillation frequency. When setting the frequency multiplication rate, be sure not to exceed the upper-limit operating clock frequency.
- The PLL can only be set up when the PLL is turned off (PLLPOWR (D0/0x301B0C) = 0) and the clock source is other than the PLL (OSCSEL[1:0] (D[3:2]/0x301B08) = 0–2). If settings are changed while the system is operating with the PLL clock, the system may operate erratically.

Precautions regarding reset input

- Even if the #RESET pin is pulled low (= 0), the chip may not be reset unless supplied with a clock. To reset the chip for sure, #RESET should be held low for at least 3 OSC3 clock cycles. However, the input/output port pins will be initialized by reset regardless of whether the chip is supplied with a clock.
- The oscillation start time of the high-speed (OSC3) oscillator circuit varies with the device used, board patterns, and operating environment. Therefore, a sufficient time should be provided before the reset signal is deasserted.

Precautions regarding NMI input

NMI cannot be nested. The CPU keeps NMI input masked out until the reti instruction is executed after an NMI exception occurred.

Precautions regarding SSCG control

- When using the SSCG, always set SSMCITM[3:0] (D[15:12]/0x301B10) to 0b0001.
- SSMCIDT[3:0] (D[11:8]/0x301B10) must be set according to the PLL output clock frequency as shown in Table III.1.7.2.1. Using the SSCG with an improper setting may cause a malfunction of the IC.
- When the PLL is off, the initial values and the written values cannot be read correctly from SSMCIDT[3:0] (D[11:8]/0x301B10) and SSMCITM[3:0] (D[15:12]/0x301B10) since the source clock is not supplied from the PLL (different values are read out). The correct values can be read out when the PLL is on.
- A stabilized clock must be supplied to the SSCG module when turning the SSCG on and off. The following shows the operation procedure.
 - To turn the SSCG on
 1. Turn the PLL on.
 2. Wait more than the PLL stabilization time.
 3. Turn the SSCG on.
 - To turn the SSCG off
 1. Turn the SSCG off.
 2. Turn the PLL off.
- The SS modulation is effective only for the PLL output clock, and is not performed for other source clocks. When the PLL output clock is not used for the system clock, turn the SSCG off.

THIS PAGE IS BLANK.

III.2 Interrupt Controller (ITC)

The S1C33E08 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 causes and conditions under which interrupts are generated, refer to the description of each peripheral circuit in this manual.

III.2.1 Outline of Interrupt Functions

III.2.1.1 Maskable Interrupts

The ITC can handle 55 kinds of maskable interrupts. Table III.2.1.1.1 shows the trap table in the S1C33E08.

Table III.2.1.1.1 Trap Table

Vector number (Hex address)	Exception/interrupt name (peripheral circuit)	Cause of exception/interrupt	IDMA Ch.	Priority
0(Base)	Reset	Low input to the reset pin	–	1
1	reserved	–	–	–
2(Base+8)	ext exception	ext instruction (illegal use)	–	4
3(Base+0C)	Undefined instruction exception	Undefined instruction	–	3
4–5	reserved	–	–	–
6(Base+18)	Address misaligned exception	Memory access instruction	–	2
0x60000	Debugging exception	brk instruction, etc.	–	5
7(Base+1C)	NMI	Low input to the #NMI pin or watchdog timer overflow	–	6
8–10	reserved	–	–	–
11(Base+2C)	Illegal interrupt exception	Occurrence of illegal interrupt from ITC	–	High
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)	Intelligent DMA	Intelligent DMA, end of transfer	–	
27–29	reserved	–	–	
30(Base+78)	16-bit timer 0	Timer 0 compare-match B	7	
31(Base+7C)		Timer 0 compare-match A	8	
32–33	reserved	–	–	
34(Base+88)	16-bit timer 1	Timer 1 compare-match B	9	
35(Base+8C)		Timer 1 compare-match A	10	
36–37	reserved	–	–	
38(Base+98)	16-bit timer 2	Timer 2 compare-match B	11	
39(Base+9C)		Timer 2 compare-match A	12	
40–41	reserved	–	–	
42(Base+A8)	16-bit timer 3	Timer 3 compare-match B	13	
43(Base+AC)		Timer 3 compare-match A	14	
44–45	reserved	–	–	
46(Base+B8)	16-bit timer 4	Timer 4 compare-match B	15	
47(Base+BC)		Timer 4 compare-match A	16	
48–49	reserved	–	–	
50(Base+C8)	16-bit timer 5	Timer 5 compare-match B	17	
51(Base+CC)		Timer 5 compare-match A	18	
52–55	reserved	–	–	↓ Low

III PERIPHERAL MODULES 1 (SYSTEM): INTERRUPT CONTROLLER (ITC)

Vector number (Hex address)	Exception/interrupt name (peripheral circuit)	Cause of exception/interrupt	IDMA Ch.	Priority
56(Base+E0)	Serial interface Ch.0	Receive error	–	High ↑
57(Base+E4)		Receive buffer full	23	
58(Base+E8)		Transmit buffer empty	24	
59	reserved	–	–	
60(Base+F0)	Serial interface Ch.1	Receive error	–	
61(Base+F4)		Receive buffer full	25	
62(Base+F8)		Transmit buffer empty	26	
63(Base+FC)	A/D converter	Result out of range (upper-limit and lower-limit)	–	
64(Base+100)		End of conversion	27	
65(Base+104)	RTC	1/64 second, 1 second, 1 minuet, or 1 hour 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)	reserved	–	–	
73(Base+124)	LCDC	End of frame	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	
79–80	reserved	–	–	
81(Base+144)	SPI	Receive DMA request	36	
82(Base+148)		Transmit DMA request	37	
83	reserved	–	–	
84(Base+150)	Port input interrupt 8	Edge (rising or falling) or level (High or Low)	38	
	SPI	SPI interrupt (D[1:0]/0x3003C4 = 0x10)		
85(Base+154)	Port input interrupt 9	Edge (rising or falling) or level (High or Low)	39	
	USB PDREQ	USB DMA request (D[3:2]/0x3003C4 = 0x10)		
86(Base+158)	Port input interrupt 10	Edge (rising or falling) or level (High or Low)	40	
	USB	USB interrupt (D[5:4]/0x3003C4 = 0x10)		
87(Base+15C)	Port input interrupt 11	Edge (rising or falling) or level (High or Low)	41	
	DCSIO	DCSIO interrupt (D[7:6]/0x3003C4 = 0x10)		
88(Base+160)	Port input interrupt 12	Edge (rising or falling) or level (High or Low)	42	
89(Base+164)	Port input interrupt 13	Edge (rising or falling) or level (High or Low)	43	
90(Base+168)	Port input interrupt 14	Edge (rising or falling) or level (High or Low)	44	
91(Base+16C)	Port input interrupt 15	Edge (rising or falling) or level (High or Low)	45	
92–93	reserved	–	–	
94(Base+178)	I ² S interface	I ² S FIFO empty	46	↓ Low
95–107	reserved	–	–	

Notes:

- The MP3 decoder BIOS (MP3 decoder module) uses the HSDMA Ch.0, Ch.1 and I²S and their interrupts for output data requests. Therefore, these DMA channels and interrupts cannot be used in the user program.

- IDMA Ch.19–22, 32, 47–53 are reserved.

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.

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

“Cause of exception/interrupt” indicates the cause of the interrupt occurring in each interrupt system.

“IDMA Ch.” indicates that a cause of interrupt which has a numeric value in this column can start up the intelligent DMA (IDMA) to transfer data when a cause of interrupt occurs. The numeric value indicates the IDMA's channel number. Causes of interrupt 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 causes of interrupt 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 causes of interrupt in the same interrupt system are fixed in the order that they are written here.

Multiplexed interrupts

The following vector numbers have two causes of interrupts assigned:

- No. 84: Port input interrupt 8 and SPI interrupt
- No. 85: Port input interrupt 9 and USB PDREQ interrupt
- No. 86: Port input interrupt 10 and USB interrupt
- No. 87: Port input interrupt 11 and DCSIO interrupt

At initial reset, these vector addresses are set for port interrupts. Each port interrupt allows selection of an input port to be used for generating interrupts from four different ports and SPI, USB, or DCSIO is included in this selection as a port. Therefore, when using the SPI, USB, or DCSIO interrupt, select it instead of an input port using the Port Input Interrupt Select Register 3 (0x3003C4). Also this setting changes the interrupt control registers for the port input interrupt to be used for controlling the SPI, USB, or DCSIO interrupt. For details of the Port Input Interrupt Select Register 3 (0x3003C4), see Section III.2.7, “Details of Control Registers.”

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 cause of interrupt 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 cause of interrupt 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 cause of trap having higher priority, such as NMI, has occurred.
- The cause of interrupt does not invoke IDMA (the IDMA request bit is set to 0).

When a cause of interrupt occurs, the corresponding cause-of-interrupt 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 cause of interrupt be inadvertently cleared even if the above conditions are not met when the cause of interrupt has occurred. The interrupt will occur when the above conditions are met.

However, when the cause of interrupt invokes IDMA, the cause of interrupt 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 causes of interrupt occur simultaneously, the cause of interrupt 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 cause of interrupt generating conditions, refer to the description of each peripheral circuit in this manual.

Illegal interrupt exception (vector No. 11)

There is a time lag between latching the interrupt signal and latching the interrupt vector and level signals caused by the interface specifications between the CPU and the ITC.

1. The CPU latches the interrupt signal sent from the ITC.
↓
2. The CPU latches the interrupt vector and level signals sent from the ITC.
↓
3. The CPU executes the interrupt handler.

An illegal interrupt exception (vector No. 11) occurs when a register related to the interrupt signal (ITC's interrupt enable and cause-of-interrupt flag registers) is altered before the CPU latches the interrupt vector and level signals (between Steps 1 and 2). Therefore, it is very rare but an illegal interrupt exception may occur if an interrupt related register is altered when interrupts to the CPU are in enabled status (IE bit in PSR = 1). However, the illegal interrupt exception that occurs does not affect the program execution if any processing is not performed in the exception handler.

To avoid an illegal interrupt exception occurring, disable interrupts to the CPU (set IE bit in PSR = 0) before altering an interrupt related register.

III.2.1.2 Causes of Interrupt and Intelligent DMA

Several causes of interrupt can be set so that they can invoke IDMA startup. When one of these causes of interrupt 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 Section III.2.5, "IDMA Invocation."

III.2.1.3 Nonmaskable Interrupt (NMI)

The nonmaskable interrupt (NMI) can be generated by controlling the #NMI pin 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.

For controlling the #NMI input, refer to Section III.1.3, "NMI Input."

III.2.1.4 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 cause of interrupt 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.

III.2.1.5 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.

In SLEEP mode, since the clock supply to the peripheral circuit is disabled, interrupts from the peripheral circuits except RTC and I/O ports 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 SLEEP mode: NMI, I/O port interrupts, and RTC interrupt

When the CPU is released from HALT mode by an interrupt, it enters a program executable state by trap processing and executes an interrupt handling routine for the interrupt generated. In trap processing of the CPU, the address for the instruction next to halt is saved to the stack as a return address from the interrupt handling routine, so that the `reti` instruction in the interrupt handling routine branches to the instruction next to halt.

The CPU is released from HALT mode when the ITC asserts the interrupt signal to be sent to the CPU. In other words, when a cause-of-interrupt flag of the interrupts that have been enabled by the interrupt enable bits in the ITC is set to 1, the CPU can be released from HALT mode even if the PSR is set to disable interrupts. However, in this case the CPU does not execute the interrupt handling routine.

The #NMI signal releases the CPU from HALT mode when it goes low level.

When the CPU is reawaken from SLEEP mode by an interrupt, it enters a program executable state by trap processing and executes an interrupt handling routine for the interrupt generated. In trap processing of the CPU, the address for the instruction next to slp is saved to the stack as a return address from the interrupt handling routine, so that the reti instruction in the interrupt handling routine branches to the instruction next to slp.

Cause-of-interrupt flags in the interrupt controller (ITC) cannot be set in SLEEP mode as the clock is not supplied to the ITC in SLEEP mode.

Therefore, when the clock is not supplied to the ITC, the interrupt signals from the interrupt sources that have been enabled to generate an interrupt are input to the CMU through the ITC and used to wake up the CPU from a standby mode. In this case, the cause-of-interrupt flag is set after the clock has started supplying to the ITC. The CPU can wake up from SLEEP mode by a cause of interrupt as described above even if the PSR is set to disable interrupts, note however, that the CPU does not execute the interrupt handling routine.

The #NMI signal releases the CPU from SLEEP mode when it goes low level.

Notes: • In SLEEP mode, there is a time lag between input of an interrupt signal for wakeup and the start of the clock supply to the ITC, so a delay will occur until the interrupt controller (ITC) sets the cause-of-interrupt flag. Therefore, no interrupt will occur if the interrupt signal is deasserted before the clock is supplied to the ITC, as the cause-of-interrupt flag in the ITC is not set.

Furthermore, additional time is needed for the CPU to accept the interrupt request from the ITC, the CPU may execute a few instructions that follow the slp instruction before it starts the interrupt processing.

The same problem may occur when the CPU wakes up from SLEEP mode by NMI. No interrupt will occur if the #NMI signal is deasserted before the clock is supplied, as the NMI flag is not set.

- If the cause of interrupt used to restart from the standby mode has been set to invoke the IDMA, the IDMA is started up by that interrupt.

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.

III.2.2 Trap Table

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

After an initial reset, the TTBR register is set to 0xC00000.

Bits 9 to 0 in the TTBR register are fixed at 0. Therefore, the trap table starting address always begins with a 1KB boundary address.

III.2.3 ITC Operating Clock

The ITC is clocked by the ITC operating clock (ITC_CLK = MCLK) generated by the CMU.

For details on how to set MCLK and control the clock, see Section III.1, “Clock Management Unit (CMU).”

Controlling supply of the ITC operating clock

The ITC operating clock is supplied to the ITC with default settings.

When this clock supply is turned off, the ITC registers except the interrupt cause flag registers and IDMA related registers (interrupt priority registers, interrupt enable registers, and HSDMA trigger setup registers) are disabled for writing. However, the cause-of-interrupt flags can be set by the corresponding interrupt request signals from the peripheral circuit, thus interrupt requests to the CPU can also be generated if the interrupt has been enabled. The cause-of-interrupt flag can be cleared by software. Moreover, HSDMA can be invoked normally. IDMA cannot be invoked. However, an interrupt sets the IDMA request bit when the IDMA enable bit has been set to 1 (IDMA request enabled). (The IDMA does not activate even if the IDMA request bit is set.)

The clock supply can be controlled by ITC_CKE (D2/0x301B04).

* **ITC_CKE**: ITC Clock Control Bit in the Gated Clock Control Register 1 (D2/0x301B04)

Clock state in standby mode

The supply of the ITC operating clock stops depending on the type of standby mode.

HALT mode: The clock is supplied the same way as in normal mode.

SLEEP mode: The clock supply stops.

Therefore, the ITC also stops operating in SLEEP mode.

III.2.4 Control of Maskable Interrupts

III.2.4.1 Structure of the Interrupt Controller

The interrupt controller is configured as shown in Figure III.2.4.1.1.

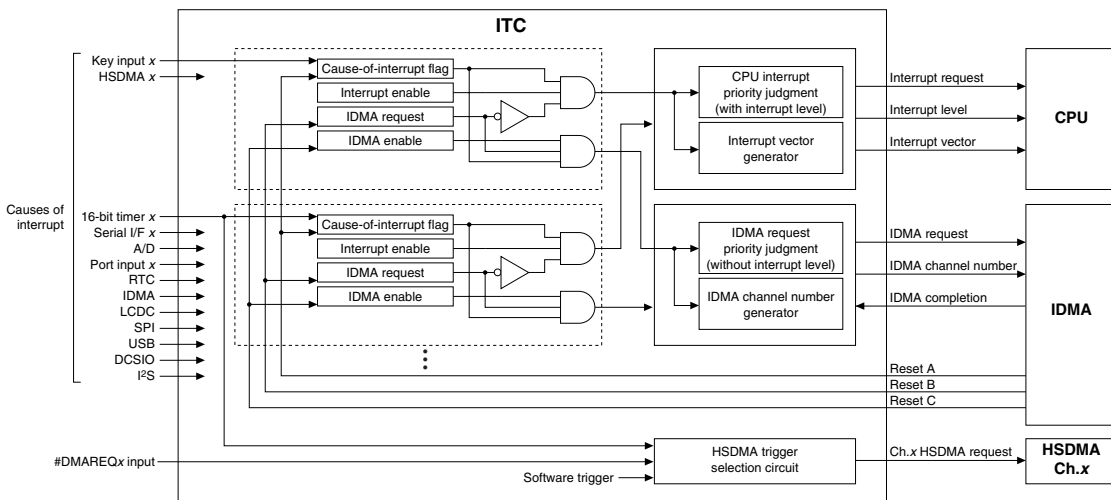


Figure III.2.4.1.1 Configuration of Interrupt Controller

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

III.2.4.2 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 Section III.2.4.4, “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 PE Core 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.

III.2.4.3 Cause-of-Interrupt Flag and Interrupt Enable Register

A cause-of-interrupt flag and an interrupt enable register are provided for each cause of maskable interrupt.

Cause-of-interrupt flag

The cause-of-interrupt flag is set to 1 when the corresponding cause of interrupt occurs. Reading the flag enables you to determine what caused an interrupt, making it unnecessary to resort to the CPU's trap processing. The cause-of-interrupt 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 (D0/0x30029F).

* **RSTONLY:** Cause-of-Interrupt Flag Reset Method Select Bit in the Flag Set/Reset Method Select Register (D0/0x30029F)

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

This method is selected (RSTONLY (D0/0x30029F) = 1) when initially reset.

With this method, the cause-of-interrupt flag is reset by writing 1. Although multiple cause-of-interrupt flags are located at the same address of the interrupt control register, the cause-of-interrupt flags for which 0 has been written can be neither set nor reset. Therefore, this method ensures that only a specific cause-of-interrupt flag is reset.

However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that a cause-of-interrupt flag that has been set to 1 is reset by writing.

In this method, no cause-of-interrupt flag can be set in the software application.

- **Read/write method**

This method is selected by writing 0 to RSTONLY (D0/0x30029F).

When this method is used, cause-of-interrupt 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 cause-of-interrupt flags for which 0 has been written are reset. Even in a read-modify-write operation, a cause of interrupt can occur between the read and the write, so be careful when using this method.

Since cause-of-interrupt 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 cause-of-interrupt flag is not reset. Consequently, if control is returned from the interrupt processing routine by the reti instruction without resetting the cause-of-interrupt flag in a program, the same cause of interrupt occurs again.

RSTONLY (D0/0x30029F), which is used to select a cause-of-interrupt flag reset method, must be set to 1 (reset-only method) when the MP3 decoder function is used.

For details about cause of interrupt 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 cause of interrupt. If the bit is set to 0, no interrupt request is made to the CPU even when the corresponding cause of interrupt 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 cause-of-interrupt flags, so when a cause of interrupt occurs the cause-of-interrupt 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 a cause of interrupt or when clearing standby mode (HALT or SLEEP mode) too, the corresponding interrupt enable bit must be set to 1.

III PERIPHERAL MODULES 1 (SYSTEM): INTERRUPT CONTROLLER (ITC)

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

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

If two or more causes of interrupt occur simultaneously, the cause of interrupt 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 cause-of-interrupt 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 cause of interrupt of higher priority occurs. They are not cleared if the CPU simply accepts the interrupt request.

III.2.4.4 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 III.2.1.1.1 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 causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt 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 a cause of interrupt occurs in two or more interrupt systems having the same value, the interrupt priority is resolved according to the default priorities in Table III.2.1.1.1. Causes of interrupt in the same interrupt system also have their priorities resolved according to the order in Table III.2.1.1.1.

Other causes of interrupt 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 cause of interrupt 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 cause of interrupt 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 a cause of interrupt works fine.

- Notes:**
- As the C33 PE Core 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.
 - 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 cause-of-interrupt flag after an interrupt has occurred, the same interrupt may occur again.

III.2.5 IDMA Invocation

The causes of interrupt for which IDMA channel numbers are written in Table III.2.1.1.1 have the function to invoke the intelligent DMA (IDMA).

IDMA request register

The IDMA request register is used to specify the cause of interrupt that invoke an IDMA transfer. If an IDMA request bit is set to 1, the IDMA request will be generated when the corresponding cause of interrupt occurs. When the IDMA request bit is set to 0, the corresponding cause of interrupt 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 (D1/0x30029F).

* **IDMAONLY:** IDMA Request Register Set Method Select Bit in the Flag Set/Reset Method Select Register (D1/0x30029F)

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

This method is selected (IDMAONLY (D1/0x30029F) = 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 (D1/0x30029F).

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 a cause of interrupt, the corresponding bit of the IDMA enable register must be set to 1. If this bit is set to 0, the cause of interrupt 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 (D2/0x30029F). See the above explanation for the set method.

* **DENONLY:** IDMA Enable Register Set Method Select Bit in the Flag Set/Reset Method Select Register (D2/0x30029F)

Invoking IDMA

Before IDMA can be invoked by the occurrence of a cause of interrupt, the corresponding bits of the IDMA request and IDMA enable registers must be set to 1. Then when a cause of interrupt occurs, the interrupt request to the CPU is made pending and the corresponding IDMA channel is invoked. The DMA transfer is performed according to the control information of that IDMA channel. The interrupt level set by the interrupt priority register of the ITC does not affect the IDMA invocation. The IDMA request can be accepted even if the interrupt level of the CPU is higher than the set value of the interrupt priority register. However, when generating the interrupt request to the CPU after the IDMA transfer is completed, the interrupt is controlled using the interrupt level set by the interrupt priority register.

An IDMA invocation request is accepted even when the interrupt enable register and PSR of the CPU is set to disable interrupts. It is also necessary that the control information for the IDMA channel has been set.

Interrupt after IDMA transfer

To generate an interrupt after completion of IDMA transfer:

The interrupt request that has been kept pending can be generated after completion of the DMA transfer. In this case, the interrupt must be enabled by the IDMA control information (DINTEN = 1) in addition to the interrupt controller and the PSR register settings.

However, if the transfer counter set for the selected IDMA channel does not reach the terminal count of 0 after the number of transfers set have been performed, the cause-of-interrupt 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 cause-of-interrupt 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 a cause of interrupt occurs next time as well. To ensure that no unwanted IDMA request occurs, this setup must be performed after resetting the cause-of-interrupt flag.

Figure III.2.5.1 shows the hardware sequence when DINTEN is set to 1.

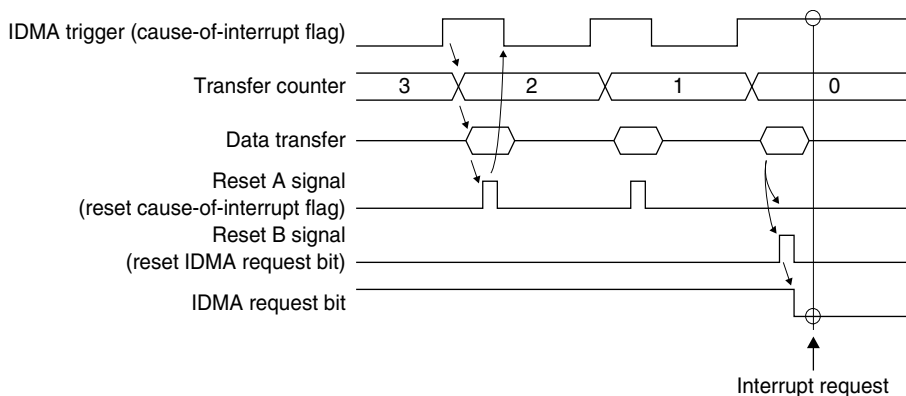


Figure III.2.5.1 Sequence when DINTEN = 1

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 cause-of-interrupt 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 cause of interrupt will be disabled.

Figure III.2.5.2 shows the hardware sequence when DINTEN is set to 0.

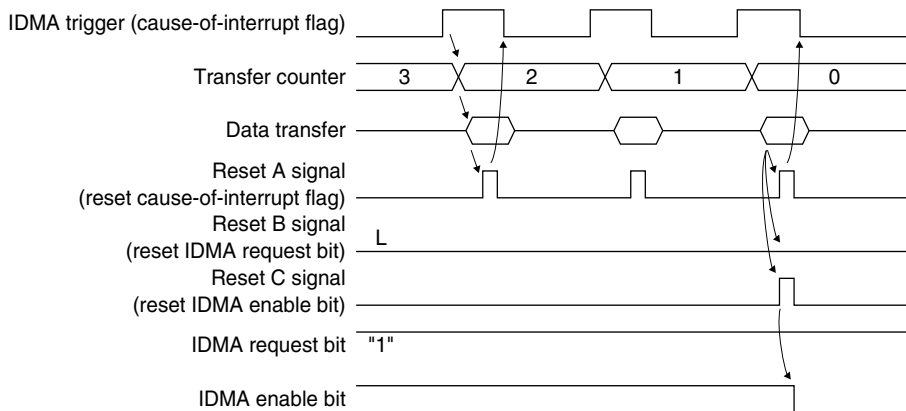


Figure III.2.5.2 Sequence when DINTEN = 0

For details on IDMA, refer to Section II.2, “Intelligent DMA (IDMA).”

III.2.6 HSDMA Invocation

Some causes of interrupt 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 a cause of interrupt as the trigger. HSDxS[3:0] (0x300298–0x300299) is used for this selection.

- * **HSD0S[3:0]**: Ch.0 Trigger Set-up Bits in the HSDMA Ch.0–1 Trigger Set-up Register (D[3:0]/0x300298)
- * **HSD1S[3:0]**: Ch.1 Trigger Set-up Bits in the HSDMA Ch.0–1 Trigger Set-up Register (D[7:4]/0x300298)
- * **HSD2S[3:0]**: Ch.2 Trigger Set-up Bits in the HSDMA Ch.2–3 Trigger Set-up Register (D[3:0]/0x300299)
- * **HSD3S[3:0]**: Ch.3 Trigger Set-up Bits in the HSDMA Ch.2–3 Trigger Set-up Register (D[7:4]/0x300299)

Table III.2.6.1 shows the setting value and the corresponding trigger source.

Table III.2.6.1 HSDMA Trigger Source

Value	Ch.0 trigger source	Ch.1 trigger source	Ch.2 trigger source	Ch.3 trigger source
0000	Software trigger	Software trigger	Software trigger	Software trigger
0001	#DMAREQ0 input (falling edge)	#DMAREQ1 input (falling edge)	#DMAREQ2 input (falling edge)	#DMAREQ3 input (falling edge)
0010	#DMAREQ0 input (rising edge)	#DMAREQ1 input (rising edge)	#DMAREQ2 input (rising edge)	#DMAREQ3 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	(reserved)	(reserved)	(reserved)	(reserved)
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	(reserved)	(reserved)
1001	I ² S left	I ² S right	SPI transmit DMA request	SPI receive DMA request
1010	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full	Serial I/F Ch.2 Rx buffer full	(reserved)
1011	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty	Serial I/F Ch.2 Tx buffer empty	(reserved)
1100	A/D conversion completion	A/D conversion completion	A/D conversion completion	A/D conversion completion
1101	Port 8 input (SPI interrupt)	Port 9 input (USB PDREQ)	Port 10 input (USB interrupt)	Port 11 input (DCSIO interrupt)
1110	Port 12 input	Port 13 input	Port 14 input	Port 15 input

Invoking HSDMA

By selecting a cause of interrupt with the HSDMA trigger set-up register, the HSDMA channel is invoked when the selected cause of interrupt occurs. The interrupt control bits (cause-of-interrupt flag, interrupt enable register, IDMA request register, interrupt priority register) do not affect this invocation.

The interrupt request to the CPU by the cause of interrupt that invokes HSDMA is output two clocks (MCLK) after the HSDMA request, so the DMA transfer and interrupt handling are performed concurrently when the CPU runs with the instructions in the cache. However, when the interrupt handler contains an instruction that accesses a peripheral circuit, the execution of the instruction is pending until the DMA transfer is completed since the bus is used by the HSDMA.

Before HSDMA can be invoked by the occurrence of a cause of interrupt, 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 Section II.1, “High-Speed DMA (HSDMA).”

III.2.7 Details of Control Registers

Table III.2.7.1 List of ITC Registers

Address	Register name	Size	Function
0x00300260	Port Input 0–1 Interrupt Priority Register (pINT_PP01L)	8	Sets interrupt level for port input 0–1 interrupts.
0x00300261	Port Input 2–3 Interrupt Priority Register (pINT_PP23L)	8	Sets interrupt level for port input 2–3 interrupts.
0x00300262	Key Input Interrupt Priority Register (pINT_PK01L)	8	Sets interrupt level for key input interrupts.
0x00300263	HSDMA Ch.0–1 Interrupt Priority Register (pINT_PHSD01L)	8	Sets interrupt level for HSDMA Ch.0–1 interrupts.
0x00300264	HSDMA Ch.2–3 Interrupt Priority Register (pINT_PHSD23L)	8	Sets interrupt level for HSDMA Ch.2–3 interrupts.
0x00300265	IDMA Interrupt Priority Register (pINT_PDM)	8	Sets interrupt level for IDMA interrupts.
0x00300266	16-bit Timer 0–1 Interrupt Priority Register (pINT_P16T01)	8	Sets interrupt level for 16-bit timer 0–1 interrupts.
0x00300267	16-bit Timer 2–3 Interrupt Priority Register (pINT_P16T23)	8	Sets interrupt level for 16-bit timer 2–3 interrupts.
0x00300268	16-bit Timer 4–5 Interrupt Priority Register (pINT_P16T45)	8	Sets interrupt level for 16-bit timer 4–5 interrupts.
0x00300269	LCDC, Serial I/F Ch.0 Interrupt Priority Register (pINT_PLCDC_PSI00)	8	Sets interrupt level for LCDC and serial I/F Ch.0 interrupts.
0x0030026A	Serial I/F Ch.1, A/D Interrupt Priority Register (pINT_PSI01_PAD)	8	Sets interrupt level for serial I/F Ch.1 and A/D converter interrupts.
0x0030026B	RTC Interrupt Priority Register (pINT_PRTC)	8	Sets interrupt level for RTC interrupts.
0x0030026C	Port Input 4–5 Interrupt Priority Register (pINT_PP45L)	8	Sets interrupt level for port input 4–5 interrupts.
0x0030026D	Port Input 6–7 Interrupt Priority Register (pINT_PP67L)	8	Sets interrupt level for port input 6–7 interrupts.
0x0030026E	Serial I/F Ch.2, SPI Interrupt Priority Register (pINT_PSI02_PSPI)	8	Sets interrupt level for serial I/F Ch.2 and SPI interrupts.
0x00300270	Key Input, Port Input 0–3 Interrupt Enable Register (pINT_EK01_EP03)	8	Enables key input and port input 0–3 interrupts.
0x00300271	DMA Interrupt Enable Register (pINT_EDMA)	8	Enables DMA interrupts.
0x00300272	16-bit Timer 0–1 Interrupt Enable Register (pINT_E16T01)	8	Enables 16-bit timer 0–1 interrupts.
0x00300273	16-bit Timer 2–3 Interrupt Enable Register (pINT_E16T23)	8	Enables 16-bit timer 2–3 interrupts.
0x00300274	16-bit Timer 4–5 Interrupt Enable Register (pINT_E16T45)	8	Enables 16-bit timer 4–5 interrupts.
0x00300276	Serial I/F Ch.0–1 Interrupt Enable Register (pINT_ESIF01)	8	Enables serial I/F Ch.0–1 interrupts.
0x00300277	Port Input 4–7, RTC, A/D Interrupt Enable Register (pINT_EP47_ERTC_EAD)	8	Enables port input 4–7, RTC and A/D interrupts.
0x00300278	LCDC Interrupt Enable Register (pINT_ELCDC)	8	Enables LCDC interrupts.
0x00300279	Serial I/F Ch.2, SPI Interrupt Enable Register (pINT_ESIF2_ESPI)	8	Enables serial I/F Ch.2 and SPI interrupts.
0x00300280	Key Input, Port Input 0–3 Interrupt Cause Flag Register (pINT_FK01_FP03)	8	Indicates/resets key input and port input 0–3 interrupt status.
0x00300281	DMA Interrupt Cause Flag Register (pINT_FDMA)	8	Indicates/resets DMA interrupt status.
0x00300282	16-bit Timer 0–1 Interrupt Cause Flag Register (pINT_F16T01)	8	Indicates/resets 16-bit timer 0–1 interrupt status.
0x00300283	16-bit Timer 2–3 Interrupt Cause Flag Register (pINT_F16T23)	8	Indicates/resets 16-bit timer 2–3 interrupt status.
0x00300284	16-bit Timer 4–5 Interrupt Cause Flag Register (pINT_F16T45)	8	Indicates/resets 16-bit timer 4–5 interrupt status.
0x00300286	Serial I/F Ch.0–1 Interrupt Cause Flag Register (pINT_FSIF01)	8	Indicates/resets serial I/F Ch.0–1 interrupt status.
0x00300287	Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (pINT_FP47_FRTC_FAD)	8	Indicates/resets port input 4–7, RTC and A/D converter interrupt status.
0x00300288	LCDC Interrupt Cause Flag Register (pINT_FLCDC)	8	Indicates/resets LCDC interrupt status.
0x00300289	Serial I/F Ch.2, SPI Interrupt Cause Flag Register (pINT_FSIF2_FSPI)	8	Indicates/resets serial I/F Ch.2 and SPI interrupt status.
0x00300290	Port Input 0–3, HSDMA Ch.0–1, 16-bit Timer 0 IDMA Request Register (pIDMAREQ_RP03_RHS_R16T0)	8	Sets IDMA invocation by port input 0–3, HSDMA Ch.0–1 and 16-bit timer 0.
0x00300291	16-bit Timer 1–4 IDMA Request Register (pIDMAREQ_R16T14)	8	Sets IDMA invocation by 16-bit timer 1–4.
0x00300292	16-bit Timer 5, Serial I/F Ch.0 IDMA Request Register (pIDMAREQ_R16T5_RSIF0)	8	Sets IDMA invocation by 16-bit timer 5 and serial I/F Ch.0.
0x00300293	Serial I/F Ch.1, A/D, Port Input 4–7 IDMA Request Register (pIDMAREQ_RSIF1_RAD_RP47)	8	Sets IDMA invocation by serial I/F Ch.1, A/D converter, and port input 4–7.

III PERIPHERAL MODULES 1 (SYSTEM): INTERRUPT CONTROLLER (ITC)

Address	Register name	Size	Function
0x00300294	Port Input 0–3, HSDMA Ch.0–1, 16-bit Timer 0 IDMA Enable Register (pIDMAEN_DEP03_DEHS_DE16T0)	8	Enables IDMA requests by port input 0–3, HSDMA Ch.0–1, and 16-bit timer 0.
0x00300295	16-bit Timer 1–4 IDMA Enable Register (pIDMAEN_DE16T14)	8	Enables IDMA requests by 16-bit timer 1–4.
0x00300296	16-bit Timer 5, Serial I/F Ch.0 IDMA Enable Register (pIDMAEN_DE16T5_DESIF0)	8	Enables IDMA requests by 16-bit timer 5 and Serial I/F Ch.0.
0x00300297	Serial I/F Ch.1, A/D, Port Input 4–7 IDMA Enable Register (pIDMAEN_DESIF1_DEAD_DEP47)	8	Enables IDMA requests by serial I/F Ch.1, A/D converter, and port input 4–7.
0x00300298	HSDMA Ch.0–1 Trigger Set-up Register (pHSDMA_HTGR1)	8	Selects HSDMA Ch.0–1 trigger sources.
0x00300299	HSDMA DMA Ch.2–3 Trigger Set-up Register (pHSDMA_HTGR2)	8	Selects HSDMA Ch.2–3 trigger sources.
0x0030029A	HSDMA Software Trigger Register (pHSDMA_HSOFTTGR)	8	Invokes HSDMA.
0x0030029B	LCDC, Serial I/F Ch.2, SPI IDMA Request Register (pIDMAREQ_RLCDC_RSIF2_RSP)	8	Sets IDMA invocation by LCDC, serial I/F Ch.2, and SPI.
0x0030029C	LCDC, Serial I/F Ch.2, SPI IDMA Enable Register (pIDMAEN_DELCDC_DESIF2_DESPI)	8	Enables IDMA requests by LCDC, serial I/F Ch.2, and SPI.
0x0030029F	Flag Set/Reset Method Select Register (pRST_RESET)	8	Selects flag set/reset method.
0x003002A0	Port Input 8–9 Interrupt Priority Register (pINT_PP89L)	8	Sets interrupt level for port input 8–9 interrupts.
0x003002A1	Port Input 10–11 Interrupt Priority Register (pINT_PP1011L)	8	Sets interrupt level for port input 10–11 interrupts.
0x003002A2	Port Input 12–13 Interrupt Priority Register (pINT_PP1213L)	8	Sets interrupt level for port input 12–13 interrupts.
0x003002A3	Port Input 14–15 Interrupt Priority Register (pINT_PP1415L)	8	Sets interrupt level for port input 14–15 interrupts.
0x003002A4	I ² S Interrupt Priority Register (pINT_PI2S)	8	Sets interrupt level for I ² S interrupts.
0x003002A6	Port Input 8–15 Interrupt Enable Register (pINT_EP815)	8	Enables port input 8–15 interrupts.
0x003002A7	I ² S Interrupt Enable Register (pINT_EI2S)	8	Enables I ² S interrupts.
0x003002A9	Port Input 8–15 Interrupt Cause Flag Register (pINT_FP815)	8	Indicates/resets port input 8–15 interrupt status.
0x003002AA	I ² S Interrupt Cause Flag Register (pINT_FI2S)	8	Indicates/resets I ² S interrupt status.
0x003002AC	Port Input 8–15 IDMA Request Register (pIDMAREQ_RP815)	8	Sets IDMA invocation by port input 8–15.
0x003002AD	I ² S IDMA Request Register (pIDMAREQ_RI2S)	8	Sets IDMA invocation by I ² S.
0x003002AE	Port Input 8–15 IDMA Enable Register (pIDMAEN_DEP815)	8	Enables IDMA requests by port input 8–15.
0x003002AF	I ² S IDMA Enable Register (pIDMAEN_DEI2S)	8	Enables IDMA requests by I ² S.
0x003003C4	Port Input Interrupt Select Register 3 (pPINTSEL_SPT811)	8	Selects ports used for FPT8–FPT11 port input interrupts. (GPIO register)

The following describes each ITC control register.

The ITC control registers are mapped in the 8-bit device area from 0x300260 to 0x3002AF, and can be accessed in units of bytes.

- Notes:**
- When setting the ITC control registers, be sure to write a 0, and not a 1, for all “reserved bits.”
 - The control registers for port input interrupts 8 to 11 change their functions for the SPI, USB, and DCSIO interrupts by setting the Port Input Interrupt Select Register 3 (0x3003C4).
 - Address 0x300275 is a reserved register. Be sure not to write 1 to D[3:0] in this address.

0x300260: Port Input 0–1 Interrupt Priority Register (pINT_PP01L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input 0–1 interrupt priority register (pINT_PP01L)	00300260 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP1L2	Port input 1 interrupt level	0 to 7	X	R/W	
		D5	PP1L1					
		D4	PP1L0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP0L2	Port input 0 interrupt level	0 to 7	X	R/W	
		D1	PP0L1					
D0	PP0L0							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PP1L[2:0]: Port Input 1 Interrupt Level Bits**
Sets the priority level of the port input 1 interrupt.

D3 **Reserved**

D[2:0] **PP0L[2:0]: Port Input 0 Interrupt Level Bits**
Sets the priority level of the port input 0 interrupt.

0x300261: Port Input 2–3 Interrupt Priority Register (pINT_PP23L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Port input 2–3 interrupt priority register (pINT_PP23L)	00300261 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PP3L2	Port input 3 interrupt level	0 to 7	X	R/W		
		D5	PP3L1						
		D4	PP3L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PP2L2	Port input 2 interrupt level	0 to 7	X	R/W		
		D1	PP2L1						
D0	PP2L0								

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PP3L[2:0]: Port Input 3 Interrupt Level Bits**

Sets the priority level of the port input 3 interrupt.

D3 **Reserved**

D[2:0] **PP2L[2:0]: Port Input 2 Interrupt Level Bits**

Sets the priority level of the port input 2 interrupt.

0x300262: Key Input Interrupt Priority Register (pINT_PK01L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Key input interrupt priority register (pINT_PK01L)	00300262 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PK1L2	Key input 1 interrupt level	0 to 7	X	R/W	
		D5	PK1L1					
		D4	PK1L0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PK0L2	Key input 0 interrupt level	0 to 7	X	R/W	
		D1	PK0L1					
D0	PK0L0							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PK1L[2:0]: Key Input 1 Interrupt Level Bits**
Sets the priority level of the key input 1 interrupt.

D3 **Reserved**

D[2:0] **PK0L[2:0]: Key Input 0 Interrupt Level Bits**
Sets the priority level of the key input 0 interrupt.

0x300263: HSDMA Ch.0–1 Interrupt Priority Register (pINT_PHSD01L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.0–1 interrupt priority register (pINT_PHSD01L)	00300263 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PHSD1L2	HSDMA Ch.1	0 to 7	X	R/W	
		D5	PHSD1L1	interrupt level		X		
		D4	PHSD1L0			X		
		D3	–	reserved	–	–	–	0 when being read.
		D2	PHSD0L2	HSDMA Ch.0	0 to 7	X	R/W	
		D1	PHSD0L1	interrupt level		X		
D0	PHSD0L0		X					

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PHSD1L[2:0]: HSDMA Ch.1 Interrupt Level Bits**

Sets the priority level of the HSDMA Ch.1 interrupt.

D3 **Reserved**

D[2:0] **PHSD0L[2:0]: HSDMA Ch.0 Interrupt Level Bits**

Sets the priority level of the HSDMA Ch.0 interrupt.

0x300264: HSDMA Ch.2–3 Interrupt Priority Register (pINT_PHSD23L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
HSDMA Ch.2–3 interrupt priority register (pINT_PHSD23L)	00300264 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PHSD3L2	HSDMA Ch.3 interrupt level	0 to 7	X	R/W		
		D5	PHSD3L1				X		
		D4	PHSD3L0				X		
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PHSD2L2	HSDMA Ch.2 interrupt level	0 to 7	X	R/W		
		D1	PHSD2L1				X		
D0	PHSD2L0				X				

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 Reserved

D[6:4] PHSD3L[2:0]: HSDMA Ch.3 Interrupt Level Bits
Sets the priority level of the HSDMA Ch.3 interrupt.

D3 Reserved

D[2:0] PHSD2L[2:0]: HSDMA Ch.2 Interrupt Level Bits
Sets the priority level of the HSDMA Ch.2 interrupt.

0x300265: IDMA Interrupt Priority Register (pINT_PDM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IDMA interrupt priority register (pINT_PDM)	00300265 (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		

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D[7:3] Reserved

D[2:0] PDM[2:0]: IDMA Interrupt Level Bits

Sets the priority level of the IDMA interrupt.

0x300266: 16-bit Timer 0–1 Interrupt Priority Register (pINT_P16T01)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit timer 0–1 interrupt priority register (pINT_P16T01)	00300266 (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						

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **P16T1[2:0]: 16-bit Timer 1 Interrupt Level Bits**
Sets the priority levels of the 16-bit timer 1 interrupt.

D3 **Reserved**

D[2:0] **P16T0[2:0]: 16-bit Timer 0 Interrupt Level Bits**
Sets the priority levels of the 16-bit timer 0 interrupt.

0x300267: 16-bit Timer 2–3 Interrupt Priority Register (pINT_P16T23)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit timer 2–3 interrupt priority register (pINT_P16T23)	00300267 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	P16T32	16-bit timer 3 interrupt level	0 to 7	X	R/W	
		D5	P16T31					
		D4	P16T30					
		D3	–	reserved	–	–	–	0 when being read.
		D2	P16T22	16-bit timer 2 interrupt level	0 to 7	X	R/W	
		D1	P16T21					
D0	P16T20							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **P16T3[2:0]: 16-bit Timer 3 Interrupt Level Bits**

Sets the priority levels of the 16-bit timer 3 interrupt.

D3 **Reserved**

D[2:0] **P16T2[2:0]: 16-bit Timer 2 Interrupt Level Bits**

Sets the priority levels of the 16-bit timer 2 interrupt.

0x300268: 16-bit Timer 4–5 Interrupt Priority Register (pINT_P16T45)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit timer 4–5 interrupt priority register (pINT_P16T45)	00300268 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	P16T52	16-bit timer 5 interrupt level	0 to 7	X	R/W	
		D5	P16T51					
		D4	P16T50					
		D3	–	reserved	–	–	–	0 when being read.
		D2	P16T42	16-bit timer 4 interrupt level	0 to 7	X	R/W	
		D1	P16T41					
D0	P16T40							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **P16T5[2:0]: 16-bit Timer 5 Interrupt Level Bits**
Sets the priority levels of the 16-bit timer 5 interrupt.

D3 **Reserved**

D[2:0] **P16T4[2:0]: 16-bit Timer 4 Interrupt Level Bits**
Sets the priority levels of the 16-bit timer 4 interrupt.

0x300269: LCDC, Serial I/F Ch.0 Interrupt Priority Register (pINT_PLCDC_PSI00)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC, serial I/F Ch.0 interrupt priority register (pINT_PLCDC_ PSI00)	00300269 (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	PLCDC2	LCDC interrupt level	0 to 7	X	R/W	
D1	PLCDC1		X					
D0	PLCDC0		X					

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PSIO0[2:0]: Serial Interface Ch.0 Interrupt Level Bits**

Sets the priority levels of the serial interface Ch.0 interrupt.

D3 **Reserved**

D[2:0] **PLCDC[2:0]: LCDC Interrupt Level Bits**

Sets the priority levels of the LCDC interrupt.

0x30026A: Serial I/F Ch.1, A/D Interrupt Priority Register (pINT_PSI01_PAD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.1, A/D interrupt priority register (pINT_PSI01_PAD)	0030026A (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PAD2	A/D converter interrupt level	0 to 7	X	R/W	
		D5	PAD1					
		D4	PAD0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PSIO12	Serial interface Ch.1 interrupt level	0 to 7	X	R/W	
		D1	PSIO11					
D0	PSIO10							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PAD[2:0]: A/D Converter Interrupt Level Bits**
Sets the priority levels of the A/D converter interrupt.

D3 **Reserved**

D[2:0] **PSIO1[2:0]: Serial Interface Ch.1 Interrupt Level Bits**
Sets the priority levels of the serial interface Ch.1 interrupt.

0x30026B: RTC Interrupt Priority Register (pINT_PRTC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC interrupt priority register (pINT_PRTC)	0030026B (B)	D7-3	–	reserved	–	–	–	Writing 1 not allowed.
		D2	PRTC2	RTC interrupt level	0 to 7	X	R/W	
		D1	PRTC1					
		D0	PRTC0					

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D[7:3] Reserved

D[2:0] PRTC[2:0]: RTC Interrupt Level Bits

Sets the priority level of the RTC interrupt.

0x30026C: Port Input 4–5 Interrupt Priority Register (pINT_PP45L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input 4–5 interrupt priority register (pINT_PP45L)	0030026C (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP5L2	Port input 5 interrupt level	0 to 7	X	R/W	
		D5	PP5L1					
		D4	PP5L0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP4L2	Port input 4 interrupt level	0 to 7	X	R/W	
		D1	PP4L1					
D0	PP4L0							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PP5L[2:0]: Port Input 5 Interrupt Level Bits**
Sets the priority level of the port input 5 interrupt.

D3 **Reserved**

D[2:0] **PP4L[2:0]: Port Input 4 Interrupt Level Bits**
Sets the priority level of the port input 4 interrupt.

0x30026D: Port Input 6–7 Interrupt Priority Register (pINT_PP67L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input 6–7 interrupt priority register (pINT_PP67L)	0030026D (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP7L2	Port input 7 interrupt level	0 to 7	X	R/W	
		D5	PP7L1					
		D4	PP7L0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP6L2	Port input 6 interrupt level	0 to 7	X	R/W	
		D1	PP6L1					
D0	PP6L0							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PP7L[2:0]: Port Input 7 Interrupt Level Bits**

Sets the priority level of the port input 7 interrupt.

D3 **Reserved**

D[2:0] **PP6L[2:0]: Port Input 6 Interrupt Level Bits**

Sets the priority level of the port input 6 interrupt.

0x30026E: Serial I/F Ch.2, SPI Interrupt Priority Register (pINT_PSI02_PSPI)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.2, SPI interrupt priority register (pINT_PSI02_PSPI)	0030026E (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PSPI2	SPI	0 to 7	X	R/W	
		D5	PSPI1	interrupt level		X		
		D4	PSPI0			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					

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PSPI[2:0]: SPI Interrupt Level Bits**
Sets the priority levels of the SPI interrupt.

D3 **Reserved**

D[2:0] **PSIO2[2:0]: Serial Interface Ch.2 Interrupt Level Bits**
Sets the priority levels of the serial interface Ch.2 interrupt.

0x300270: Key Input, Port Input 0–3 Interrupt Enable Register (pINT_EK01_EP03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Key input, port input 0–3 interrupt enable register (pINT_EK01_EP03)	00300270 (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	

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D[7:6] Reserved**D5 EK1: Key Input 1 Interrupt Enable Bit**

Enables or disables the key input 1 interrupt.

D4 EK0: Key Input 0 Interrupt Enable Bit

Enables or disables the key input 0 interrupt.

D3 EP3: Port Input 3 Interrupt Enable Bit

Enables or disables the port input 3 interrupt.

D2 EP2: Port Input 2 Interrupt Enable Bit

Enables or disables the port input 2 interrupt.

D1 EP1: Port Input 1 Interrupt Enable Bit

Enables or disables the port input 1 interrupt.

D0 EP0: Port Input 0 Interrupt Enable Bit

Enables or disables the port input 0 interrupt.

0x300271: DMA Interrupt Enable Register (pINT_EDMA)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
DMA interrupt enable register (pINT_EDMA)	00300271 (B)	D7-5	–	reserved	–		–	–	0 when being read.	
		D4	EIDMA	IDMA	1	Enabled	0	Disabled	0	R/W
		D3	EHDMA3	HSDMA Ch.3					0	R/W
		D2	EHDMA2	HSDMA Ch.2					0	R/W
		D1	EHDMA1	HSDMA Ch.1					0	R/W
		D0	EHDMA0	HSDMA Ch.0					0	R/W

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D[7:5] Reserved

D4 **EIDMA: IDMA Enable Interrupt Bit**

Enables or disables the IDMA interrupt.

D3 **EHDMA3: HSDMA Ch.3 Interrupt Enable Bit**

Enables or disables the HSDMA Ch.3 interrupt.

D2 **EHDMA2: HSDMA Ch.2 Interrupt Enable Bit**

Enables or disables the HSDMA Ch.2 interrupt.

D1 **EHDMA1: HSDMA Ch.1 Interrupt Enable Bit**

Enables or disables the HSDMA Ch.1 interrupt.

D0 **EHDMA0: HSDMA Ch.0 Interrupt Enable Bit**

Enables or disables the HSDMA Ch.0 interrupt.

0x300272: 16-bit Timer 0–1 Interrupt Enable Register (pINT_E16T01)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 0–1 interrupt enable register (pINT_E16T01)	00300272 (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.

Each bit in this register enables or disables an interrupt to the CPU.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D7 E16TC1: 16-bit Timer 1 Comparison A Interrupt Enable Bit
Enables or disables the 16-bit timer 1 comparison A interrupt.

D6 E16TU1: 16-bit Timer 1 Comparison B Interrupt Enable Bit
Enables or disables the 16-bit timer 1 comparison B interrupt.

D[5:4] Reserved

D3 E16TC0: 16-bit Timer 0 Comparison A Interrupt Enable Bit
Enables or disables the 16-bit timer 0 comparison A interrupt.

D2 E16TU0: 16-bit Timer 0 Comparison B Interrupt Enable Bit
Enables or disables the 16-bit timer 0 comparison B interrupt.

D[1:0] Reserved

0x300273: 16-bit Timer 2–3 Interrupt Enable Register (pINT_E16T23)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 2–3 interrupt enable register (pINT_E16T23)	00300273 (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.

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D7 E16TC3: 16-bit Timer 3 Comparison A Interrupt Enable Bit

Enables or disables the 16-bit timer 3 comparison A interrupt.

D6 E16TU3: 16-bit Timer 3 Comparison B Interrupt Enable Bit

Enables or disables the 16-bit timer 3 comparison B interrupt.

D[5:4] Reserved

D3 E16TC2: 16-bit Timer 2 Comparison A Interrupt Enable Bit

Enables or disables the 16-bit timer 2 comparison A interrupt.

D2 E16TU2: 16-bit Timer 2 Comparison B Interrupt Enable Bit

Enables or disables the 16-bit timer 2 comparison B interrupt.

D[1:0] Reserved

0x300274: 16-bit Timer 4–5 Interrupt Enable Register (pINT_E16T45)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 4–5 interrupt enable register (pINT_E16T45)	00300274 (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.

Each bit in this register enables or disables an interrupt to the CPU.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D7 E16TC5: 16-bit Timer 5 Comparison A Interrupt Enable Bit
Enables or disables the 16-bit timer 5 comparison A interrupt.

D6 E16TU5: 16-bit Timer 5 Comparison B Interrupt Enable Bit
Enables or disables the 16-bit timer 5 comparison B interrupt.

D[5:4] Reserved

D3 E16TC4: 16-bit Timer 4 Comparison A Interrupt Enable Bit
Enables or disables the 16-bit timer 4 comparison A interrupt.

D2 E16TU4: 16-bit Timer 4 Comparison B Interrupt Enable Bit
Enables or disables the 16-bit timer 4 comparison B interrupt.

D[1:0] Reserved

0x300276: Serial I/F Ch.0–1 Interrupt Enable Register (pINT_ESIF01)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Serial I/F Ch.0–1 interrupt enable register (pINT_ESIF01)	00300276 (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

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D[7:6] Reserved

D5 **ESTX1: SIF Ch.1 Transmit Buffer Empty Interrupt Enable Bit**

Enables or disables the SIF Ch.1 transmit buffer empty interrupt.

D4 **ESRX1: SIF Ch.1 Receive Buffer Full Interrupt Enable Bit**

Enables or disables the SIF Ch.1 receive buffer full interrupt.

D3 **ESERR1: SIF Ch.1 Receive Error Interrupt Enable Bit**

Enables or disables the SIF Ch.1 receive error interrupt.

D2 **ESTX0: SIF Ch.0 Transmit Buffer Empty Interrupt Enable Bit**

Enables or disables the SIF Ch.0 transmit buffer empty interrupt.

D1 **ESRX0: SIF Ch.0 Receive Buffer Full Interrupt Enable Bit**

Enables or disables the SIF Ch.0 receive buffer full interrupt.

D0 **ESERR0: SIF Ch.0 Receive Error Interrupt Enable Bit**

Enables or disables the SIF Ch.0 receive error interrupt.

0x300277: Port Input 4–7, RTC, A/D Interrupt Enable Register (pINT_EP47_ERTC_EAD)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Port input 4–7, RTC, A/D interrupt enable register (pINT_EP47_ERTC_EAD)	00300277 (B)	D7	–	reserved	1	Enabled	0	Disabled	–	–	0 when being read.
		D6	EP7	Port input 7					0	R/W	
		D5	EP6	Port input 6					0	R/W	
		D4	EP5	Port input 5					0	R/W	
		D3	EP4	Port input 4					0	R/W	
		D2	ERTC	RTC					0	R/W	
		D1	EADE	A/D conversion completion					0	R/W	
		D0	EADC	A/D out-of-range					0	R/W	

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D7 Reserved

D6 EP7: Port Input 7 Interrupt Enable Bit
Enables or disables the port input 7 interrupt.

D5 EP6: Port Input 6 Interrupt Enable Bit
Enables or disables the port input 6 interrupt.

D4 EP5: Port Input 5 Interrupt Enable Bit
Enables or disables the port input 5 interrupt.

D3 EP4: Port Input 4 Interrupt Enable Bit
Enables or disables the port input 4 interrupt.

D2 ERTC: RTC Interrupt Enable Bit
Enables or disables the RTC interrupt.

D1 EADE: A/D Conversion Completion Interrupt Enable Bit
Enables or disables the A/D conversion completion interrupt.

D0 EADC: A/D Out-of-Range Interrupt Enable Bit
Enables or disables the A/D upper/lower limit interrupt.

0x300278: LCDC Interrupt Enable Register (pINT_ELDCD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC interrupt enable register (pINT_ELDCD)	00300278 (B)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	ELCDC	LCDC frame end	1 Enabled 0 Disabled	0	R/W	
		D0	–	reserved	–	–	–	Do not write 1.

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D[7:2] Reserved

D1 ELCD: LCDC Interrupt Enable Bit

Enables or disables interrupt generation of the LCDC interrupt.

D0 Reserved (Do not write 1 to this bit.)

0x300279: Serial I/F Ch.2, SPI Interrupt Enable Register (pINT_ESIF2_ESPI)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Serial I/F Ch.2, SPI interrupt enable register (pINT_ESIF2_ESPI)	00300279 (B)	D7-6	–	reserved		–	–	–	0 when being read.		
		D5	ESPITX	SPI transmit DMA	1	Enabled	0	Disabled	0	R/W	
		D4	ESPIRX	SPI receive DMA					0	R/W	
		D3	–	reserved			–	–	–	–	Do not write 1.
		D2	ESTX2	SIF Ch.2 transmit buffer empty	1	Enabled	0	Disabled	0	R/W	
		D1	ESRX2	SIF Ch.2 receive buffer full					0	R/W	
		D0	ESERR2	SIF Ch.2 receive error					0	R/W	

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D[7:6] Reserved

D5 ESPITX: SPI Transmit DMA Interrupt Enable Bit

Enables or disables the SPI transmit DMA interrupt.

D4 ESPIRX: SPI Receive DMA Interrupt Enable Bit

Enables or disables the SPI receive DMA interrupt.

D3 Reserved (Do not write 1 to this bit.)

D2 ESTX2: SIF Ch.2 Transmit Buffer Empty Interrupt Enable Bit

Enables or disables the SIF Ch.2 transmit buffer empty interrupt.

D1 ESRX2: SIF Ch.2 Receive Buffer Full Interrupt Enable Bit

Enables or disables the SIF Ch.2 receive buffer full interrupt.

D0 ESERR2: SIF Ch.2 Receive Error Interrupt Enable Bit

Enables or disables the SIF Ch.2 receive error interrupt.

0x300280: Key Input, Port Input 0–3 Interrupt Cause Flag Register (pINT_FK01_FP03)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Key input, port input 0–3 interrupt cause flag register (pINT_FK01_FP03)	00300280 (B)	D7–6	–	reserved	1	Occurred	0	Not occurred	–	–	0 when being read.
		D5	FK1	Key input 1					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	

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

The cause-of-interrupt flag is set to 1 when a cause of interrupt 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 a cause of interrupt to request IDMA, note that even when the above conditions are met, no interrupt request to the CPU is generated for the cause of interrupt 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 cause-of-interrupt flag is always set to 1 when a cause of interrupt 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 cause-of-interrupt 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 cause-of-interrupt 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 cause-of-interrupt 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 (D0/0x30029F) = 1) and 0 when using the read/write method (RSTONLY (D0/0x30029F) = 0). Be careful not to confuse these two conditions.

The cause-of-interrupt flag becomes indeterminate when initially reset, so be sure to reset the flag in the software application.

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

D[7:6]	Reserved
D5	FK1: Key Input 1 Cause-of-Interrupt Flag Indicates the key input 1 interrupt cause status.
D4	FK0: Key Input 0 Cause-of-Interrupt Flag Indicates the key input 0 interrupt cause status.
D3	FP3: Port Input 3 Cause-of-Interrupt Flag Indicates the port input 3 interrupt cause status.
D2	FP2: Port Input 2 Cause-of-Interrupt Flag Indicates the port input 2 interrupt cause status.
D1	FP1: Port Input 1 Cause-of-Interrupt Flag Indicates the port input 1 interrupt cause status.
D0	FP0: Port Input 0 Cause-of-Interrupt Flag Indicates the port input 0 interrupt cause status.

0x300281: DMA Interrupt Cause Flag Register (pINT_FDMA)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
DMA interrupt cause flag register (pINT_FDMA)	00300281 (B)	D7-5	–	reserved	–		–	–	0 when being read.	
		D4	FIDMA	IDMA	1	Occurred	0	Not occurred	X	R/W
		D3	FHDM3	HSDMA Ch.3					X	R/W
		D2	FHDM2	HSDMA Ch.2					X	R/W
		D1	FHDM1	HSDMA Ch.1					X	R/W
		D0	FHDM0	HSDMA Ch.0					X	R/W

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D[7:5] Reserved

D4 FIDMA: IDMA Cause-of-Interrupt Flag

Indicates the IDMA interrupt cause status.

D3 FHDM3: HSDMA Ch.3 Cause-of-Interrupt Flag

Indicates the HSDMA Ch.3 interrupt cause status.

D2 EHDM2: HSDMA Ch.2 Cause-of-Interrupt Flag

Indicates the HSDMA Ch.2 interrupt cause status.

D1 FHDM1: HSDMA Ch.1 Cause-of-Interrupt Flag

Indicates the HSDMA Ch.1 interrupt cause status.

D0 FHDM0: HSDMA Ch.0 Cause-of-Interrupt Flag

Indicates the HSDMA Ch.0 interrupt cause status.

0x300282: 16-bit Timer 0–1 Interrupt Cause Flag Register (pINT_F16T01)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 0–1 interrupt cause flag register (pINT_F16T01)	00300282 (B)	D7	F16TC1	16-bit timer 1 comparison A	1	Occurred	0	Not occurred	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	Occurred	0	Not occurred	X	R/W	
		D2	F16TU0	16-bit timer 0 comparison B					X	R/W	
		D1–0	–	reserved					–	–	0 when being read.

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D7 F16TC1: 16-bit Timer 1 Comparison A Cause-of-Interrupt Flag

Indicates the 16-bit timer 1 comparison A interrupt cause status.

D6 F16TU1: 16-bit Timer 1 Comparison B Cause-of-Interrupt Flag

Indicates the 16-bit timer 1 comparison B interrupt cause status.

D[5:4] Reserved**D3 F16TC0: 16-bit Timer 0 Comparison A Cause-of-Interrupt Flag**

Indicates the 16-bit timer 0 comparison A interrupt cause status.

D2 F16TU0: 16-bit Timer 0 Comparison B Cause-of-Interrupt Flag

Indicates the 16-bit timer 0 comparison B interrupt cause status.

D[1:0] Reserved

0x300283: 16-bit Timer 2–3 Interrupt Cause Flag Register (pINT_F16T23)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 2–3 interrupt cause flag register (pINT_F16T23)	00300283 (B)	D7	F16TC3	16-bit timer 3 comparison A	1	Occurred	0	Not occurred	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	Occurred	0	Not occurred	X	R/W	
		D2	F16TU2	16-bit timer 2 comparison B					X	R/W	
		D1–0	–	reserved					–	–	0 when being read.

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D7 F16TC3: 16-bit Timer 3 Comparison A Cause-of-Interrupt Flag
Indicates the 16-bit timer 3 comparison A interrupt cause status.

D6 F16TU3: 16-bit Timer 3 Comparison B Cause-of-Interrupt Flag
Indicates the 16-bit timer 3 comparison B interrupt cause status.

D[5:4] Reserved

D3 F16TC2: 16-bit Timer 2 Comparison A Cause-of-Interrupt Flag
Indicates the 16-bit timer 2 comparison A interrupt cause status.

D2 F16TU2: 16-bit Timer 2 Comparison B Cause-of-Interrupt Flag
Indicates the 16-bit timer 2 comparison B interrupt cause status.

D[1:0] Reserved

0x300284: 16-bit Timer 4–5 Interrupt Cause Flag Register (pINT_F16T45)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 4–5 interrupt cause flag register (pINT_F16T45)	00300284 (B)	D7	F16TC5	16-bit timer 5 comparison A	1	Occurred	0	Not occurred	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	Occurred	0	Not occurred	X	R/W	
		D2	F16TU4	16-bit timer 4 comparison B					X	R/W	
		D1–0	–	reserved					–	–	0 when being read.

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

- D7 F16TC5: 16-bit Timer 5 Comparison A Cause-of-Interrupt Flag**
Indicates the 16-bit timer 5 comparison A interrupt cause status.
- D6 F16TU5: 16-bit Timer 5 Comparison B Cause-of-Interrupt Flag**
Indicates the 16-bit timer 5 comparison B interrupt cause status.
- D[5:4] Reserved**
- D3 F16TC4: 16-bit Timer 4 Comparison A Cause-of-Interrupt Flag**
Indicates the 16-bit timer 4 comparison A interrupt cause status.
- D2 F16TU4: 16-bit Timer 4 Comparison B Cause-of-Interrupt Flag**
Indicates the 16-bit timer 4 comparison B interrupt cause status.
- D[1:0] Reserved**

0x300286: Serial I/F Ch.0–1 Interrupt Cause Flag Register (pINT_FSIF01)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Serial I/F Ch.0–1 interrupt cause flag register (pINT_FSIF01)	00300286 (B)	D7–6	–	reserved	–		–	–	0 when being read.	
		D5	FSTX1	SIF Ch.1 transmit buffer empty	1	Occurred	0	Not occurred	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

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D[7:6] Reserved

D5 FSTX1: SIF Ch.1 Transmit Buffer Empty Cause-of-Interrupt Flag

Indicates the SIF Ch.1 transmit buffer empty interrupt cause status.

D4 FSRX1: SIF Ch.1 Receive Buffer Full Cause-of-Interrupt Flag

Indicates the SIF Ch.1 receive buffer full interrupt cause status.

D3 FSERR1: SIF Ch.1 Receive Error Cause-of-Interrupt Flag

Indicates the SIF Ch.1 receive error interrupt cause status.

D2 FSTX0: SIF Ch.0 Transmit Buffer Empty Cause-of-Interrupt Flag

Indicates the SIF Ch.0 transmit buffer empty interrupt cause status.

D1 FSRX0: SIF Ch.0 Receive Buffer Full Cause-of-Interrupt Flag

Indicates the SIF Ch.0 receive buffer full interrupt cause status.

D0 FSERR0: SIF Ch.0 Receive Error Cause-of-Interrupt Flag

Indicates the SIF Ch.0 receive error interrupt cause status.

0x300287: Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (pINT_FP47_FRTC_FAD)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Port input 4–7, RTC, A/D interrupt cause flag register (pINT_FP47_FRTC _FAD)	00300287 (B)	D7	–	reserved	1	Occurred	0	Not occurred	–	–	0 when being read.
		D6	FP7	Port input 7					X	R/W	
		D5	FP6	Port input 6					X	R/W	
		D4	FP5	Port input 5					X	R/W	
		D3	FP4	Port input 4					X	R/W	
		D2	FRTC	RTC					X	R/W	
		D1	FADE	A/D conversion completion					X	R/W	
		D0	FADC	A/D out-of-range					X	R/W	

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D7 Reserved

D6 FP7: Port Input 7 Cause-of-Interrupt Flag

Indicates the port input 7 interrupt cause status.

D5 FP6: Port Input 6 Cause-of-Interrupt Flag

Indicates the port input 6 interrupt cause status.

D4 FP5: Port Input 5 Cause-of-Interrupt Flag

Indicates the port input 5 interrupt cause status.

D3 FP4: Port Input 4 Cause-of-Interrupt Flag

Indicates the port input 4 interrupt cause status.

D2 FRTC: RTC Cause-of-Interrupt Flag

Indicates the RTC interrupt cause status.

D1 FADE: A/D Conversion Completion Cause-of-Interrupt Flag

Indicates the A/D conversion completion interrupt cause status.

D0 FADC: A/D Out-of-Range Cause-of-Interrupt Flag

Indicates the A/D upper/lower limit interrupt cause status.

0x300288: LCDC Interrupt Cause Flag Register (pINT_FLCDC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
LCDC interrupt cause flag register (pINT_FLCDC)	00300288 (B)	D7-2	-	reserved	-		-	-	0 when being read.	
		D1	FLCDC	LCDC frame end	1	Occurred	0	Not occurred	X	R/W
		D0	-	reserved	-		-	-	-	0 when being read.

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D[7:2] Reserved

D1 FLCDC: LCDC Cause-of-Interrupt Flag

Indicates the LCDC interrupt cause status.

D0 Reserved

0x300289: Serial I/F Ch.2, SPI Interrupt Cause Flag Register (pINT_FSIF2_FSPI)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Serial I/F Ch.2, SPI interrupt cause flag register (pINT_FSIF2_FSPI)	00300289 (B)	D7-6	–	reserved		–	–	–	0 when being read.	
		D5	FSPITX	SPI transmit DMA	1	Occurred	0	Not occurred	X	R/W
		D4	FSPIRX	SPI receive DMA					X	R/W
		D3	–	reserved			–	–	–	0 when being read.
		D2	FSTX2	SIF Ch.2 transmit buffer empty	1	Occurred	0	Not occurred	X	R/W
		D1	FSRX2	SIF Ch.2 receive buffer full					X	R/W
		D0	FSERR2	SIF Ch.2 receive error					X	R/W

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D[7:6] Reserved**D5 FSPITX: SPI Transmit DMA Cause-of-Interrupt Flag**

Indicates the SPI transmit DMA interrupt cause status.

D4 FSPIRX: SPI Receive DMA Cause-of-Interrupt Flag

Indicates the SPI receive DMA interrupt cause status.

D3 Reserved**D2 FSTX2: SIF Ch.2 Transmit Buffer Empty Cause-of-Interrupt Flag**

Indicates the SIF Ch.2 transmit buffer empty interrupt cause status.

D1 FSRX2: SIF Ch.2 Receive Buffer Full Cause-of-Interrupt Flag

Indicates the SIF Ch.2 receive buffer full interrupt cause status.

D0 FSERR2: SIF Ch.2 Receive Error Cause-of-Interrupt Flag

Indicates the SIF Ch.2 receive error interrupt cause status.

0x300290: Port Input 0–3, HSDMA Ch.0–1, 16-bit Timer 0 IDMA Request Register (pIDMAREQ_RP03_RHS_R16T0)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Port input 0–3, HSDMA Ch.0–1, 16-bit timer 0 IDMA request register (pIDMAREQ_RP03_RHS_R16T0)	00300290 (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	HSDMA Ch.1					0	R/W
		D4	RHDM0	HSDMA 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

Each bit in this register specifies whether to invoke IDMA when a cause of interrupt occurs.

When using the set-only method (default)

1 (R/W): IDMA request

0 (R/W): IDMA not invoked (default)

When using the read/write method

1 (R/W): IDMA request

0 (R/W): Interrupt request

If the bit is set to 1, IDMA is invoked when a cause of interrupt 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 Section II.2, “Intelligent DMA (IDMA).”

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 bit is reset to 0 and an interrupt request for the cause of interrupt that enabled IDMA invoking is generated.

D7 R16TC0: 16-bit Timer 0 Comparison A IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 0 comparison A interrupt occurs or not.

D6 R16TU0: 16-bit Timer 0 Comparison B IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 0 comparison B interrupt occurs or not.

D5 RHDM1: HSDMA Ch.1 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the HSDMA Ch.1 interrupt occurs or not.

D4 RHDM0: HSDMA Ch.0 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the HSDMA Ch.0 interrupt occurs or not.

D3 RP3: Port Input 3 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 3 interrupt occurs or not.

D2 RP2: Port Input 2 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 2 interrupt occurs or not.

D1 RP1: Port Input 1 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 1 interrupt occurs or not.

D0 RP0: Port Input 0 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 0 interrupt occurs or not.

0x300291: 16-bit Timer 1–4 IDMA Request Register (pIDMAREQ_R16T14)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
16-bit timer 1–4 IDMA request register (pIDMAREQ_R16T14)	00300291 (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

Each bit in this register specifies whether to invoke IDMA when a cause of interrupt occurs.

When using the set-only method (default)

1 (R/W): IDMA request

0 (R/W): IDMA not invoked (default)

When using the read/write method

1 (R/W): IDMA request

0 (R/W): Interrupt request

If the bit is set to 1, IDMA is invoked when a cause of interrupt 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 Section II.2, “Intelligent DMA (IDMA).”

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 bit is reset to 0 and an interrupt request for the cause of interrupt that enabled IDMA invoking is generated.

D7 R16TC4: 16-bit Timer 4 Comparison A IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 4 comparison A interrupt occurs or not.

D6 R16TU4: 16-bit Timer 4 Comparison B IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 4 comparison B interrupt occurs or not.

D5 R16TC3: 16-bit Timer 3 Comparison A IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 3 comparison A interrupt occurs or not.

D4 R16TU3: 16-bit Timer 3 Comparison B IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 3 comparison B interrupt occurs or not.

D3 R16TC2: 16-bit Timer 2 Comparison A IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 2 comparison A interrupt occurs or not.

D2 R16TU2: 16-bit Timer 2 Comparison B IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 2 comparison B interrupt occurs or not.

D1 R16TC1: 16-bit Timer 1 Comparison A IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 1 comparison A interrupt occurs or not.

D0 R16TU1: 16-bit Timer 1 Comparison B IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 1 comparison B interrupt occurs or not.

0x300292: 16-bit Timer 5, Serial I/F Ch.0 IDMA Request Register (pIDMAREQ_R16T5_RSIF0)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
16-bit timer 5, serial I/F Ch.0 IDMA request register (pIDMAREQ_R16T5_RSIF0)	00300292 (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	
		D5-2	–	reserved			–		–	–	0 when being read.
		D1	R16TC5	16-bit timer 5 comparison A	1	IDMA request	0	Interrupt request	0	R/W	
		D0	R16TU5	16-bit timer 5 comparison B					0	R/W	

Each bit in this register specifies whether to invoke IDMA when a cause of interrupt occurs.

When using the set-only method (default)

1 (R/W): IDMA request

0 (R/W): IDMA not invoked (default)

When using the read/write method

1 (R/W): IDMA request

0 (R/W): Interrupt request

If the bit is set to 1, IDMA is invoked when a cause of interrupt 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 Section II.2, “Intelligent DMA (IDMA).”

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 bit is reset to 0 and an interrupt request for the cause of interrupt that enabled IDMA invoking is generated.

D7 RSTX0: SIF Ch.0 Transmit Buffer Empty IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the SIF Ch.0 transmit buffer empty interrupt occurs or not.

D6 RSRX0: SIF Ch.0 Receive Buffer Full IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the SIF Ch.0 receive buffer full interrupt occurs or not.

D[5:2] Reserved

D1 R16TC5: 16-bit Timer 5 Comparison A IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 5 comparison A interrupt occurs or not.

D0 R16TU5: 16-bit Timer 5 Comparison B IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the 16-bit timer 5 comparison B interrupt occurs or not.

0x300293: Serial I/F Ch.1, A/D, Port Input 4–7 IDMA Request Register (pIDMAREQ_RSIF1_RAD_RP47)

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 (pIDMAREQ_RSIF1 _RAD_RP47)	00300293 (B)	D7	RP7	Port input 7	1	IDMA request	0	Interrupt request	0	R/W	
		D6	RP6	Port input 6							
		D5	RP5	Port input 5							
		D4	RP4	Port input 4							
		D3	–	reserved	–		–	–	0 when being read.		
		D2	RADE	A/D conversion completion	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	

Each bit in this register specifies whether to invoke IDMA when a cause of interrupt occurs.

When using the set-only method (default)

1 (R/W): IDMA request

0 (R/W): IDMA not invoked (default)

When using the read/write method

1 (R/W): IDMA request

0 (R/W): Interrupt request

If the bit is set to 1, IDMA is invoked when a cause of interrupt 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 Section II.2, “Intelligent DMA (IDMA).”

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 bit is reset to 0 and an interrupt request for the cause of interrupt that enabled IDMA invoking is generated.

D7 RP7: Port Input 7 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 7 interrupt occurs or not.

D6 RP6: Port Input 6 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 6 interrupt occurs or not.

D5 RP5: Port Input 5 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 5 interrupt occurs or not.

D4 RP4: Port Input 4 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 4 interrupt occurs or not.

D3 Reserved

D2 RADE: A/D Conversion Completion IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the A/D conversion completion interrupt occurs or not.

D1 RSTX1: SIF Ch.1 Transmit Buffer Empty IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the SIF Ch.1 transmit buffer empty interrupt occurs or not.

D0 RSRX1: SIF Ch.1 Receive Buffer Full IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the SIF Ch.1 receive buffer full interrupt occurs or not.

0x300294: Port Input 0–3, HSDMA Ch.0–1, 16-bit Timer 0 IDMA Enable Register (pIDMAEN_DEP03_DEHS_DE16T0)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Port input 0–3, HSDMA Ch.0–1, 16-bit timer 0 IDMA enable register (pIDMAEN_DEP03_DEHS_DE16T0)	00300294 (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	HSDMA Ch.1					0	R/W	
		D4	DEHDM0	HSDMA 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	

Each bit in this register enables or disables the IDMA request by a cause of interrupt.

When using the set-only method (default)

1 (R/W): IDMA-request enabled

0 (R): IDMA-request disabled (default)

0 (W): Has no effect

When using the read/write method

1 (R/W): IDMA-request enabled

0 (R/W): IDMA-request disabled

If a bit of this register is set to 1, the IDMA request by the cause of interrupt is enabled. If the register bit is set to 0, the IDMA request is disabled.

D7 DE16TC0: 16-bit Timer 0 Comparison A IDMA Enable Bit

Enables or disables the IDMA request by a cause of the 16-bit timer 0 comparison A interrupt.

D6 DE16TU0: 16-bit Timer 0 Comparison B IDMA Enable Bit

Enables or disables the IDMA request by a cause of the 16-bit timer 0 comparison B interrupt.

D5 DEHDM1: HSDMA Ch.1 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the HSDMA Ch.1 interrupt.

D4 DEHDM0: HSDMA Ch.0 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the HSDMA Ch.0 interrupt.

D3 DEP3: Port Input 3 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 3 interrupt.

D2 DEP2: Port Input 2 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 2 interrupt.

D1 DEP1: Port Input 1 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 1 interrupt.

D0 DEP0: Port Input 0 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 0 interrupt.

0x300295: 16-bit Timer 1–4 IDMA Enable Register (pIDMAEN_DE16T14)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
16-bit timer 1–4 IDMA enable register (pIDMAEN _DE16T14)	00300295 (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	

Each bit in this register enables or disables the IDMA request by a cause of interrupt.

When using the set-only method (default)

1 (R/W): IDMA-request enabled

0 (R): IDMA-request disabled (default)

0 (W): Has no effect

When using the read/write method

1 (R/W): IDMA-request enabled

0 (R/W): IDMA-request disabled

If a bit of this register is set to 1, the IDMA request by the cause of interrupt is enabled. If the register bit is set to 0, the IDMA request is disabled.

- D7 DE16TC4: 16-bit Timer 4 Comparison A IDMA Enable Bit**
Enables or disables the IDMA request by a cause of the 16-bit timer 4 comparison A interrupt.
- D6 DE16TU4: 16-bit Timer 4 Comparison B IDMA Enable Bit**
Enables or disables the IDMA request by a cause of the 16-bit timer 4 comparison B interrupt.
- D5 DE16TC3: 16-bit Timer 3 Comparison A IDMA Enable Bit**
Enables or disables the IDMA request by a cause of the 16-bit timer 3 comparison A interrupt.
- D4 DE16TU3: 16-bit Timer 3 Comparison B IDMA Enable Bit**
Enables or disables the IDMA request by a cause of the 16-bit timer 3 comparison B interrupt.
- D3 DE16TC2: 16-bit Timer 2 Comparison A IDMA Enable Bit**
Enables or disables the IDMA request by a cause of the 16-bit timer 2 comparison A interrupt.
- D2 DE16TU2: 16-bit Timer 2 Comparison B IDMA Enable Bit**
Enables or disables the IDMA request by a cause of the 16-bit timer 2 comparison B interrupt.
- D1 DE16TC1: 16-bit Timer 1 Comparison A IDMA Enable Bit**
Enables or disables the IDMA request by a cause of the 16-bit timer 1 comparison A interrupt.
- D0 DE16TU1: 16-bit Timer 1 Comparison B IDMA Enable Bit**
Enables or disables the IDMA request by a cause of the 16-bit timer 1 comparison B interrupt.

0x300296: 16-bit Timer 5, Serial I/F Ch.0 IDMA Enable Register (pIDMAEN_DE16T5_DESIF0)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
16-bit timer 5, serial I/F Ch.0 IDMA enable register (pIDMAEN_DE16T5_DESIF0)	00300296 (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	
		D5-2	–	reserved					–	–	0 when being read.
		D1	DE16TC5	16-bit timer 5 comparison A	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D0	DE16TU5	16-bit timer 5 comparison B					0	R/W	

Each bit in this register enables or disables the IDMA request by a cause of interrupt.

When using the set-only method (default)

1 (R/W): IDMA-request enabled

0 (R): IDMA-request disabled (default)

0 (W): Has no effect

When using the read/write method

1 (R/W): IDMA-request enabled

0 (R/W): IDMA-request disabled

If a bit of this register is set to 1, the IDMA request by the cause of interrupt is enabled. If the register bit is set to 0, the IDMA request is disabled.

D7 DESTX0: SIF Ch.0 Transmit Buffer Empty IDMA Enable Bit

Enables or disables the IDMA request by a cause of the SIF Ch.0 transmit buffer empty interrupt.

D6 DESRX0: SIF Ch.0 Receive Buffer Full IDMA Enable Bit

Enables or disables the IDMA request by a cause of the SIF Ch.0 receive buffer full interrupt.

D[5:2] Reserved

D1 DE16TC5: 16-bit Timer 5 Comparison A IDMA Enable Bit

Enables or disables the IDMA request by a cause of the 16-bit timer 5 comparison A interrupt.

D0 DE16TU5: 16-bit Timer 5 Comparison B IDMA Enable Bit

Enables or disables the IDMA request by a cause of the 16-bit timer 5 comparison B interrupt.

0x300297: Serial I/F Ch.1, A/D, Port Input 4–7 IDMA Enable Register (pIDMAEN_DESIF1_DEAD_DEP47)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Serial I/F Ch.1, A/D, port input 4–7 IDMA enable register (pIDMAEN_DESIF1 _DEAD_DEP47)	00300297 (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 conversion completion	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	

Each bit in this register enables or disables the IDMA request by a cause of interrupt.

When using the set-only method (default)

1 (R/W): IDMA-request enabled

0 (R): IDMA-request disabled (default)

0 (W): Has no effect

When using the read/write method

1 (R/W): IDMA-request enabled

0 (R/W): IDMA-request disabled

If a bit of this register is set to 1, the IDMA request by the cause of interrupt is enabled. If the register bit is set to 0, the IDMA request is disabled.

D7 DEP7: Port Input 7 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 7 interrupt.

D6 DEP6: Port Input 6 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 6 interrupt.

D5 DEP5: Port Input 5 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 5 interrupt.

D4 DEP4: Port Input 4 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 4 interrupt.

D3 Reserved

D2 DEADE: A/D Conversion Completion IDMA Enable Bit

Enables or disables the IDMA request by a cause of the A/D conversion completion interrupt.

D1 DESTX1: SIF Ch.1 Transmit Buffer Empty IDMA Enable Bit

Enables or disables the IDMA request by a cause of the SIF Ch.1 transmit buffer empty interrupt.

D0 DESRX1: SIF Ch.1 Receive Buffer Full IDMA Enable Bit

Enables or disables the IDMA request by a cause of the SIF Ch.1 receive buffer full interrupt.

0x300298: HSDMA Ch.0–1 Trigger Set-up Register (pHSDMA_HTGR1)
0x300299: HSDMA Ch.2–3 Trigger Set-up Register (pHSDMA_HTGR2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.0–1 trigger set-up register (pHSDMA_HTGR1)	00300298 (B)	D7	HSD1S3	HSDMA Ch.1 trigger set-up	0	Software trigger	0	R/W
		D6	HSD1S2		1	#DMAREQ1 input (falling edge)	0	
		D5	HSD1S1		2	#DMAREQ1 input (rising edge)	0	
		D4	HSD1S0		3	Port 1 input	0	
					4	Port 5 input		
					5	(reserved)		
					6	16-bit timer 1 compare B		
					7	16-bit timer 1 compare A		
					8	16-bit timer 5 compare B		
					9	I ² S right		
					A	SI/F Ch.1 Rx buffer full		
					B	SI/F Ch.1 Tx buffer empty		
					C	A/D conversion completion		
					D	Port 9 input (USB PDREQ)		
E	Port 13 input							
D3	HSD0S3			HSDMA Ch.0 trigger set-up	0	Software trigger	0	R/W
D2	HSD0S2	1	#DMAREQ0 input (falling edge)		0			
D1	HSD0S1	2	#DMAREQ0 input (rising edge)		0			
D0	HSD0S0	3	Port 0 input		0			
		4	Port 4 input					
		5	(reserved)					
		6	16-bit timer 0 compare B					
		7	16-bit timer 0 compare A					
		8	16-bit timer 4 compare B					
		9	I ² S left					
		A	SI/F Ch.0 Rx buffer full					
		B	SI/F Ch.0 Tx buffer empty					
		C	A/D conversion completion					
		D	Port 8 input (SPI interrupt)					
		E	Port 12 input					
HSDMA Ch.2–3 trigger set-up register (pHSDMA_HTGR2)	00300299 (B)	D7	HSD3S3	HSDMA Ch.3 trigger set-up	0	Software trigger	0	R/W
		D6	HSD3S2		1	#DMAREQ3 input (falling edge)	0	
		D5	HSD3S1		2	#DMAREQ3 input (rising edge)	0	
		D4	HSD3S0		3	Port 3 input	0	
					4	Port 7 input		
					5	(reserved)		
					6	16-bit timer 3 compare B		
					7	16-bit timer 3 compare A		
					8	(reserved)		
					9	SPI Rx		
					A	(reserved)		
					B	(reserved)		
					C	A/D conversion completion		
					D	Port 11 input (DCSIO interrupt)		
E	Port 15 input							
D3	HSD2S3			HSDMA Ch.2 trigger set-up	0	Software trigger	0	R/W
D2	HSD2S2	1	#DMAREQ2 input (falling edge)		0			
D1	HSD2S1	2	#DMAREQ2 input (rising edge)		0			
D0	HSD2S0	3	Port 2 input		0			
		4	Port 6 input					
		5	(reserved)					
		6	16-bit timer 2 compare B					
		7	16-bit timer 2 compare A					
		8	(reserved)					
		9	SPI Tx					
		A	SI/F Ch.2 Rx buffer full					
		B	SI/F Ch.2 Tx buffer empty					
		C	A/D conversion completion					
		D	Port 10 input (USB interrupt)					
		E	Port 14 input					

These registers are used to select a trigger source for invoking each HSDMA channel.

Table III.2.7.2 HSDMA Trigger Source

Value	Ch.0 trigger source	Ch.1 trigger source	Ch.2 trigger source	Ch.3 trigger source
0000	Software trigger	Software trigger	Software trigger	Software trigger
0001	#DMAREQ0 input (falling edge)	#DMAREQ1 input (falling edge)	#DMAREQ2 input (falling edge)	#DMAREQ3 input (falling edge)
0010	#DMAREQ0 input (rising edge)	#DMAREQ1 input (rising edge)	#DMAREQ2 input (rising edge)	#DMAREQ3 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	(reserved)	(reserved)	(reserved)	(reserved)
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	(reserved)	(reserved)
1001	I ² S left	I ² S right	SPI transmit DMA request	SPI receive DMA request
1010	Serial I/F Ch.0 Rx buffer full	Serial I/F Ch.1 Rx buffer full	Serial I/F Ch.2 Rx buffer full	(reserved)
1011	Serial I/F Ch.0 Tx buffer empty	Serial I/F Ch.1 Tx buffer empty	Serial I/F Ch.2 Tx buffer empty	(reserved)
1100	A/D conversion completion	A/D conversion completion	A/D conversion completion	A/D conversion completion
1101	Port 8 input (SPI interrupt)	Port 9 input (USB PDREQ)	Port 10 input (USB interrupt)	Port 11 input (DCSIO interrupt)
1110	Port 12 input	Port 13 input	Port 14 input	Port 15 input

(Default: 0000)

By selecting a cause of interrupt with the HSDMA trigger set-up bit, the HSDMA channel is invoked when the selected cause of interrupt occurs. The interrupt control bits (cause-of-interrupt flag, interrupt enable register, IDMA request register, interrupt priority register) do not affect this invocation.

The interrupt request to the CPU by the cause of interrupt that invokes HSDMA is output two clocks (MCLK) after the HSDMA request, so the DMA transfer and interrupt handling are performed concurrently when the CPU runs with the instructions in the cache. However, when the interrupt handler contains an instruction that accesses a peripheral circuit, the execution of the instruction is pending until the DMA transfer is completed since the bus is used by the HSDMA.

Before HSDMA can be invoked by the occurrence of a cause of interrupt, 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 Section II.1, “High-Speed DMA (HSDMA).”

0x30029A: HSDMA Software Trigger Register (pHSDMA_HSOFTTGR)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
HSDMA software trigger register (pHSDMA_HSOFTTGR)	0030029A (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

Each control bit in this register is used to start a HSDMA transfer.

- 1 (W): Trigger
- 0 (W): Has no effect
- 0 (R): Always 0 when read (default)

Writing 1 to HST_x generates a trigger pulse that starts a DMA transfer. HST_x is effective only when software trigger is selected as the trigger source of the HSDMA channel using a HSDMA trigger set-up register (0x300298 or 0x300299).

D[7:4] Reserved

- D3 HST3: HSDMA Ch.3 Software Trigger**
Starts a DMA transfer using the HSDMA Ch.3.
- D2 HST2: HSDMA Ch.2 Software Trigger**
Starts a DMA transfer using the HSDMA Ch.2.
- D1 HST1: HSDMA Ch.1 Software Trigger**
Starts a DMA transfer using the HSDMA Ch.1.
- D0 HST0: HSDMA Ch.0 Software Trigger**
Starts a DMA transfer using the HSDMA Ch.0.

0x30029B: LCDC, Serial I/F Ch.2, SPI IDMA Request Register (pIDMAREQ_RLCDC_RSIF2_RSPI)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
LCDC, serial I/F Ch.2, SPI IDMA request register (pIDMAREQ_RLCDC _RSIF2_RSPI)	0030029B (B)	D7-6	–	reserved	–		–	–	0 when being read.	
		D5	RSPITX	SPI transmit DMA	1	IDMA request	0	Interrupt request	0	R/W
		D4	RSPIRX	SPI receive DMA					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	RLCDC	LCDC frame end					0	R/W
		D0	–	reserved	–		–	–	0 when being read.	

Each bit in this register specifies whether to invoke IDMA when a cause of interrupt occurs.

When using the set-only method (default)

1 (R/W): IDMA request

0 (R/W): IDMA not invoked (default)

When using the read/write method

1 (R/W): IDMA request

0 (R/W): Interrupt request

If the bit is set to 1, IDMA is invoked when a cause of interrupt 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 Section II.2, “Intelligent DMA (IDMA).”

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 bit is reset to 0 and an interrupt request for the cause of interrupt that enabled IDMA invoking is generated.

D[7:6] Reserved

D5 RSPITX: SPI Transmit IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the SPI transmit DMA interrupt occurs or not.

D4 RSPIRX: SPI Receive IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the SPI receive DMA interrupt occurs or not.

D3 RSTX2: SIF Ch.2 Transmit Buffer Empty IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the SIF Ch.2 transmit buffer empty interrupt occurs or not.

D2 RSRX2: SIF Ch.2 Receive Buffer Full IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the SIF Ch.2 receive buffer full interrupt occurs or not.

D1 RLCDC: LCDC IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the LCDC interrupt occurs or not.

D0 Reserved

0x30029C: LCDC, Serial I/F Ch.2, SPI IDMA Enable Register (pIDMAEN_DELCDC_DESIF2_DESPI)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC, serial I/F Ch.2, SPI IDMA enable register (pIDMAEN_DELCDC _DESIF2_DESPI)	0030029C (B)	D7-6	-	reserved	-	-	-	0 when being read.
		D5	DESPITX	SPI transmit DMA	1 IDMA enabled	0 IDMA disabled	0	R/W
		D4	DESPIRX	SPI receive DMA			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	DELCDC	LCDC frame end			0	R/W
		D0	-	reserved	-	-	-	0 when being read.

Each bit in this register enables or disables the IDMA request by a cause of interrupt.

When using the set-only method (default)

1 (R/W): IDMA-request enabled

0 (R): IDMA-request disabled (default)

0 (W): Has no effect

When using the read/write method

1 (R/W): IDMA-request enabled

0 (R/W): IDMA-request disabled

If a bit of this register is set to 1, the IDMA request by the cause of interrupt is enabled. If the register bit is set to 0, the IDMA request is disabled.

D[7:6] Reserved

D5 DESPITX: SPI Transmit IDMA Enable Bit

Enables or disables the IDMA request by a cause of the SPI transmit DMA interrupt.

D4 DESPIRX: SPI Receive IDMA Enable Bit

Enables or disables the IDMA request by a cause of the SPI receive DMA interrupt.

D3 DESTX2: SIF Ch.2 Transmit Buffer Empty IDMA Enable Bit

Enables or disables the IDMA request by a cause of the SIF Ch.2 transmit buffer empty interrupt.

D2 DESRX2: SIF Ch.2 Receive Buffer Full IDMA Enable Bit

Enables or disables the IDMA request by a cause of the SIF Ch.2 receive buffer full interrupt.

D1 DELCDC: LCDC IDMA Enable Bit

Enables or disables the IDMA request by a cause of the LCDC interrupt.

D0 Reserved

0x30029F: Flag Set/Reset Method Select Register (pRST_RESET)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Flag set/reset method select register (pRST_RESET)	0030029F (B)	D7-3	–	reserved	–	–	–	0 when being read.
		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	Cause-of-interrupt flag reset method selection	1 Reset only 0 RD/WR	1	R/W	

D[7:3] Reserved

D2 **DENONLY: IDMA Enable Register Set Method Select Bit**

Selects the method for setting the IDMA enable registers.

1 (R/W): Set-only method (default)

0 (R/W): Read/write method

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 this bit. 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 IDMA enable bit can be reset by the hardware between the read and the write, so be careful when using this method.

D1 **IDMAONLY: IDMA Request Register Set Method Select Bit**

Selects the method for setting the IDMA request registers.

1 (R/W): Set-only method (default)

0 (R/W): Read/write method

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 this bit. 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.

D0 **RSTONLY: Cause-of-Interrupt Flag Reset Method Select Bit**

Selects the method for resetting the cause-of-interrupt flags.

1 (R/W): Reset-only method (default)

0 (R/W): Read/write method

With the reset-only method, the cause-of-interrupt flag is reset by writing 1.

The cause-of-interrupt flags for which 0 has been written can neither be set nor reset. Therefore, this method ensures that only a specific cause-of-interrupt flag is reset. However, when using read-modify-write instructions (e.g., bset, bclr, or bnot), note that a cause-of-interrupt flag that has been set to 1 is reset by writing. This method cannot be used to set any cause-of-interrupt flag in the software application.

The read/write method is selected by writing 0 to this bit. When this method is selected, cause-of-interrupt 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 cause-of-interrupt flags for which 0 has been written are reset. Even in a read-modify-write operation, a cause of interrupt can occur between read and write instructions, so be careful when using this method.

0x3002A0: Port Input 8–9 Interrupt Priority Register (pINT_PP89L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input 8–9 interrupt priority register (pINT_PP89L)	003002A0 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP9L2	Port input 9/USB PDREQ interrupt level	0 to 7	X	R/W	
		D5	PP9L1					
		D4	PP9L0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP8L2	Port input 8/SPI interrupt level	0 to 7	X	R/W	
		D1	PP8L1					
D0	PP8L0							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PP9L[2:0]: Port Input 9/USB PDREQ Interrupt Level Bits**

Sets the priority level of the port input 9 interrupt.

Note: These bits function as the USB PDREQ interrupt control bits when SPT9[1:0] (D[3:2]/0x3003C4) = 10.

D3 **Reserved**

D[2:0] **PP8L[2:0]: Port Input 8/SPI Interrupt Level Bits**

Sets the priority level of the port input 8 interrupt.

Note: These bits function as the SPI interrupt control bits when SPT8[1:0] (D[1:0]/0x3003C4) = 10.

0x3002A1: Port Input 10–11 Interrupt Priority Register (pINT_PP1011L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input 10–11 interrupt priority register (pINT_PP1011L)	003002A1 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP11L2	Port input 11/DCSIO interrupt level	0 to 7	X	R/W	
		D5	PP11L1					
		D4	PP11L0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP10L2	Port input 10/USB interrupt level	0 to 7	X	R/W	
		D1	PP10L1					
D0	PP10L0							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PP11L[2:0]: Port Input 11/DCSIO Interrupt Level Bits**

Sets the priority level of the port input 11 interrupt.

Note: These bits function as the DCSIO interrupt control bits when SPTB[1:0] (D[7:6]/0x3003C4) = 10.

D3 **Reserved**

D[2:0] **PP10L[2:0]: Port Input 10/USB Interrupt Level Bits**

Sets the priority level of the port input 10 interrupt.

Note: These bits function as the USB interrupt control bits when SPTA[1:0] (D[5:4]/0x3003C4) = 10.

0x3002A2: Port Input 12–13 Interrupt Priority Register (pINT_PP1213L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input 12–13 interrupt priority register (pINT_PP1213L)	003002A2 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP13L2	Port input 13 interrupt level	0 to 7	X	R/W	
		D5	PP13L1					
		D4	PP13L0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP12L2	Port input 12 interrupt level	0 to 7	X	R/W	
		D1	PP12L1					
D0	PP12L0							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PP13L[2:0]: Port Input 13 Interrupt Level Bits**
Sets the priority level of the port input 13 interrupt.

D3 **Reserved**

D[2:0] **PP12L[2:0]: Port Input 12 Interrupt Level Bits**
Sets the priority level of the port input 12 interrupt.

0x3002A3: Port Input 14–15 Interrupt Priority Register (pINT_PP1415L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input 14–15 interrupt priority register (pINT_PP1415L)	003002A3 (B)	D7	–	reserved	–	–	–	0 when being read.
		D6	PP15L2	Port input 15 interrupt level	0 to 7	X	R/W	
		D5	PP15L1					
		D4	PP15L0					
		D3	–	reserved	–	–	–	0 when being read.
		D2	PP14L2	Port input 14 interrupt level	0 to 7	X	R/W	
		D1	PP14L1					
D0	PP14L0							

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D7 **Reserved**

D[6:4] **PP15L[2:0]: Port Input 15 Interrupt Level Bits**

Sets the priority level of the port input 15 interrupt.

D3 **Reserved**

D[2:0] **PP14L[2:0]: Port Input 14 Interrupt Level Bits**

Sets the priority level of the port input 14 interrupt.

0x3002A4: I²S Interrupt Priority Register (pINT_PI2S)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S interrupt priority register (pINT_PI2S)	003002A4 (B)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	PI2S2	I ² S interrupt level	0 to 7	X	R/W	
		D1	PI2S1					
		D0	PI2S0					

This register is used to set an interrupt priority level. (Default: indeterminate)

The priority level can be set in the range of 0 to 7.

If the level is set below the IL value of the PSR, no interrupt is generated.

D[7:3] Reserved

D[2:0] PI2S[2:0]: I²S Interrupt Level Bits

Sets the priority levels of the I²S interrupt.

0x3002A6: Port Input 8–15 Interrupt Enable Register (pINT_EP815)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
Port input 8–15 interrupt enable register (pINT_EP815)	003002A6 (B)	D7	EP15	Port input 15	1	Enabled	0	Disabled	0	R/W	
		D6	EP14	Port input 14					0	R/W	
		D5	EP13	Port input 13					0	R/W	
		D4	EP12	Port input 12					0	R/W	
		D3	EP11	Port input 11/DCSIO					0	R/W	
		D2	EP10	Port input 10/USB					0	R/W	
		D1	EP9	Port input 9/USB PDREQ					0	R/W	
		D0	EP8	Port input 8/SPI					0	R/W	

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D7 EP15: Port Input 15 Enable Bit

Enables or disables the port input 15 interrupt.

D6 EP14: Port Input 14 Enable Bit

Enables or disables the port input 14 interrupt.

D5 EP13: Port Input 13 Enable Bit

Enables or disables the port input 13 interrupt.

D4 EP12: Port Input 12 Enable Bit

Enables or disables the port input 12 interrupt.

D3 EP11: Port Input 11/DCSIO Enable Bit

Enables or disables the port input 11 interrupt.

Note: This bit functions as the DCSIO interrupt control bit when SPTB[1:0] (D[7:6]/0x3003C4) = 10.

D2 EP10: Port Input 10/USB Enable Bit

Enables or disables the port input 10 interrupt.

Note: This bit functions as the USB interrupt control bit when SPTA[1:0] (D[5:4]/0x3003C4) = 10.

D1 EP9: Port Input 9/USB PDREQ Enable Bit

Enables or disables the port input 9 interrupt.

Note: This bit functions as the USB PDREQ interrupt control bit when SPT9[1:0] (D[3:2]/0x3003C4) = 10.

D0 EP8: Port Input 8/SPI Enable Bit

Enables or disables the port input 8 interrupt.

Note: This bit functions as the SPI interrupt control bit when SPT8[1:0] (D[1:0]/0x3003C4) = 10.

0x3002A7: I²S Interrupt Enable Register (pINT_EI2S)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S interrupt enable register (pINT_EI2S)	003002A7 (B)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	EI2S	I ² S	1 Enabled 0 Disabled	0	R/W	
		D1-0	–	reserved	–	–	–	0 when being read.

Each bit in this register enables or disables an interrupt to the CPU.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

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 causes of interrupt used to request IDMA invocation or clear the standby mode, the corresponding interrupt enable register bit must be set for interrupt enable.

D[7:3] Reserved

D2 EI2S: I²S Enable Bit

Enables or disables the I²S interrupt.

D[1:0] Reserved

0x3002A9: Port Input 8–15 Interrupt Cause Flag Register (pINT_FP815)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Port input 8–15 interrupt cause flag register (pINT_FP815)	003002A9 (B)	D7	FP15	Port input 15	1	Occurred	0	Not occurred	X	R/W
		D6	FP14	Port input 14					X	R/W
		D5	FP13	Port input 13					X	R/W
		D4	FP12	Port input 12					X	R/W
		D3	FP11	Port input 11/DCSIO					X	R/W
		D2	FP10	Port input 10/USB					X	R/W
		D1	FP9	Port input 9/USB PDREQ					X	R/W
		D0	FP8	Port input 8/SPI					X	R/W

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D7 FP15: Port Input 15 Cause-of-Interrupt Flag

Indicates the port input 15 interrupt cause status.

D6 FP14: Port Input 14 Cause-of-Interrupt Flag

Indicates the port input 14 interrupt cause status.

D5 FP13: Port Input 13 Cause-of-Interrupt Flag

Indicates the port input 13 interrupt cause status.

D4 FP12: Port Input 12 Cause-of-Interrupt Flag

Indicates the port input 12 interrupt cause status.

D3 FP11: Port Input 11/DCSIO Cause-of-Interrupt Flag

Indicates the port input 11 interrupt cause status.

Note: This bit functions as the DCSIO interrupt control bit when SPTB[1:0] (D[7:6]/0x3003C4) = 10.

D2 FP10: Port Input 10/USB Cause-of-Interrupt Flag

Indicates the port input 10 interrupt cause status.

Note: This bit functions as the USB interrupt control bit when SPTA[1:0] (D[5:4]/0x3003C4) = 10.

D1 FP9: Port Input 9/USB PDREQ Cause-of-Interrupt Flag

Indicates the port input 9 interrupt cause status.

Note: This bit functions as the USB PDREQ interrupt control bit when SPT9[1:0] (D[3:2]/0x3003C4) = 10.

D0 FP8: Port Input 8/SPI Cause-of-Interrupt Flag

Indicates the port input 8 interrupt cause status.

Note: This bit functions as the SPI interrupt control bit when SPT8[1:0] (D[1:0]/0x3003C4) = 10.

0x3002AA: I²S Interrupt Cause Flag Register (pINT_FI2S)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S interrupt cause flag register (pINT_FI2S)	003002AA (B)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	FI2S	I ² S	1 Occurred 0 Not occurred	X	R/W	
		D1-0	–	reserved	–	–	–	0 when being read.

Each bit in this register is a cause-of-interrupt flag to indicate the interrupt cause occurrence status. The flag that has been set can be reset by writing. (Default: indeterminate)

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred

When written using the reset-only method (default)

- 1 (W): Flag is reset
- 0 (W): Has no effect

When written using the read/write method

- 1 (W): Flag is set
- 0 (W): Flag is reset

See “Key Input, Port Input 0–3 Interrupt Cause Flag Register (0x300280)” for more information.

D[7:3] Reserved

D2 FI2S: I²S Cause-of-Interrupt Flag
Indicates the I²S interrupt cause status.

D[1:0] Reserved

0x3002AC: Port Input 8–15 IDMA Request Register (pIDMAREQ_RP815)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Port input 8–15 IDMA request register (pIDMAREQ_RP815)	003002AC (B)	D7	RP15	Port input 15	1	IDMA request	0	Interrupt request	0	R/W	
		D6	RP14	Port input 14					0	R/W	
		D5	RP13	Port input 13					0	R/W	
		D4	RP12	Port input 12					0	R/W	
		D3	RP11	Port input 11/DCSIO					0	R/W	
		D2	RP10	Port input 10/USB					0	R/W	
		D1	RP9	Port input 9/USB PDREQ					0	R/W	
		D0	RP8	Port input 8/SPI					0	R/W	

Each bit in this register specifies whether to invoke IDMA when a cause of interrupt occurs.

When using the set-only method (default)

1 (R/W): IDMA request

0 (R/W): IDMA not invoked (default)

When using the read/write method

1 (R/W): IDMA request

0 (R/W): Interrupt request

If the bit is set to 1, IDMA is invoked when a cause of interrupt 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 Section II.2, “Intelligent DMA (IDMA).”

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 bit is reset to 0 and an interrupt request for the cause of interrupt that enabled IDMA invoking is generated.

D7 RP15: Port Input 15 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 15 interrupt occurs or not.

D6 RP14: Port Input 14 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 14 interrupt occurs or not.

D5 RP13: Port Input 13 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 13 interrupt occurs or not.

D4 RP12: Port Input 12 IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 12 interrupt occurs or not.

D3 RP11: Port Input 11/DCSIO IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 11 interrupt occurs or not.

Note: This bit functions as the DCSIO interrupt control bit when $SPTB[1:0] (D[7:6]/0x3003C4) = 10$.

D2 RP10: Port Input 10/USB IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 10 interrupt occurs or not.

Note: This bit functions as the USB interrupt control bit when $SPTA[1:0] (D[5:4]/0x3003C4) = 10$.

D1 RP9: Port Input 9/USB PDREQ IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 9 interrupt occurs or not.

Note: This bit functions as the USB PDREQ interrupt control bit when $SPT9[1:0] (D[3:2]/0x3003C4) = 10$.

D0 RP8: Port Input 8/SPI IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the port input 8 interrupt occurs or not.

Note: This bit functions as the SPI interrupt control bit when $SPT8[1:0] (D[1:0]/0x3003C4) = 10$.

0x3002AD: I²S IDMA Request Register (pIDMAREQ_RI2S)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
I ² S IDMA request register (pIDMAREQ_RI2S)	003002AD (B)	D7-1	–	reserved	–			–	–	0 when being read.
		D0	RI2S	I ² S	1 IDMA request	0	Interrupt request	0	R/W	

Each bit in this register specifies whether to invoke IDMA when a cause of interrupt occurs.

When using the set-only method (default)

1 (R/W): IDMA request

0 (R/W): IDMA not invoked (default)

When using the read/write method

1 (R/W): IDMA request

0 (R/W): Interrupt request

If the bit is set to 1, IDMA is invoked when a cause of interrupt 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 Section II.2, “Intelligent DMA (IDMA).”

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 bit is reset to 0 and an interrupt request for the cause of interrupt that enabled IDMA invoking is generated.

D[7:1] Reserved

D0 RI2S: I²S IDMA Request Bit

Specifies whether to invoke IDMA when a cause of the I²S interrupt occurs or not.

0x3002AE: Port Input 8–15 IDMA Enable Register (pIDMAEN_DEP815)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Port input 8–15 IDMA enable register (pIDMAEN_DEP815)	003002AE (B)	D7	DEP15	Port input 15	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DEP14	Port input 14					0	R/W	
		D5	DEP13	Port input 13					0	R/W	
		D4	DEP12	Port input 12					0	R/W	
		D3	DEP11	Port input 11/DCSIO					0	R/W	
		D2	DEP10	Port input 10/USB					0	R/W	
		D1	DEP9	Port input 9/USB PDREQ					0	R/W	
		D0	DEP8	Port input 8/SPI					0	R/W	

Each bit in this register enables or disables the IDMA request by a cause of interrupt.

When using the set-only method (default)

1 (R/W): IDMA-request enabled

0 (R): IDMA-request disabled (default)

0 (W): Has no effect

When using the read/write method

1 (R/W): IDMA-request enabled

0 (R/W): IDMA-request disabled

If a bit of this register is set to 1, the IDMA request by the cause of interrupt is enabled. If the register bit is set to 0, the IDMA request is disabled.

D7 DEP15: Port Input 15 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 15 interrupt.

D6 DEP14: Port Input 14 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 14 interrupt.

D5 DEP13: Port Input 13 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 13 interrupt.

D4 DEP12: Port Input 12 IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 12 interrupt.

D3 DEP11: Port Input 11/DCSIO IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 11 interrupt.

Note: This bit functions as the DCSIO interrupt control bit when $SPTB[1:0] (D[7:6]/0x3003C4) = 10$.

D2 DEP10: Port Input 10/USB IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 10 interrupt.

Note: This bit functions as the USB interrupt control bit when $SPTA[1:0] (D[5:4]/0x3003C4) = 10$.

D1 DEP9: Port Input 9/USB PDREQ IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 9 interrupt.

Note: This bit functions as the USB PDREQ interrupt control bit when $SPT9[1:0] (D[3:2]/0x3003C4) = 10$.

D0 DEP8: Port Input 8/SPI IDMA Enable Bit

Enables or disables the IDMA request by a cause of the port input 8 interrupt.

Note: This bit functions as the SPI interrupt control bit when $SPT8[1:0] (D[1:0]/0x3003C4) = 10$.

0x3002AF: I²S IDMA Enable Register (pIDMAEN_DEI2S)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S IDMA enable register (pIDMAEN_DEI2S)	003002AF (B)	D7-1	–	reserved	–		–	–	0 when being read.
		D0	DEI2S	I ² S	1 IDMA enabled	0 IDMA disabled	0	R/W	

Each bit in this register enables or disables the IDMA request by a cause of interrupt.

When using the set-only method (default)

1 (R/W): IDMA-request enabled

0 (R): IDMA-request disabled (default)

0 (W): Has no effect

When using the read/write method

1 (R/W): IDMA-request enabled

0 (R/W): IDMA-request disabled

If a bit of this register is set to 1, the IDMA request by the cause of interrupt is enabled. If the register bit is set to 0, the IDMA request is disabled.

D[7:1] Reserved

D0 DEI2S: I²S IDMA Enable Bit

Enables or disables the IDMA request by a cause of the I²S interrupt.

0x3003C4: Port Input Interrupt Select Register 3 (pPINTSEL_SPT811)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input interrupt select register 3 (pPINTSEL_SPT811)	003003C4 (B)	D7	SPTB1	FPT11 interrupt input port selection	SPTB[1:0]	Port	0	R/W
		D6	SPTB0		11	P93	0	
					10	INT_DCSIO		
					01	P83		
				00	P73			
		D5	SPTA1	FPT10 interrupt input port selection	SPTA[1:0]	Port	0	R/W
		D4	SPTA0		11	P92	0	
					10	INT_USB		
					01	P82		
				00	P72			
		D3	SPT91	FPT9 interrupt input port selection	SPT9[1:0]	Port	0	R/W
		D2	SPT90		11	P91	0	
					10	USB_PDREQ		
					01	P81		
				00	P71			
		D1	SPT81	FPT8 interrupt input port selection	SPT8[1:0]	Port	0	R/W
D0	SPT80	11	P90		0			
		10	INT_SPI					
		01	P80					
		00	P70					

SPTx[1:0]: FPTx Interrupt Input Port Select Bits

Selects an input pin or interrupt source used to generate the FPTx port input interrupt.

Table III.2.7.3 Selecting Port for Port Input Interrupts

Interrupt system	SPT settings			
	11	10	01	00
FPT11	P93	INT_DCSIO	P83	P73
FPT10	P92	INT_USB	P82	P72
FPT9	P91	USB_PDREQ	P81	P71
FPT8	P90	INT_SPI	P80	P70

(Default: 0b00)

III.2.8 Precautions

- In SLEEP mode, there is a time lag between input of an interrupt signal for wakeup and the start of the clock supply to the ITC, so a delay will occur until the ITC sets the cause-of-interrupt flag. Therefore, no interrupt will occur if the interrupt signal is deasserted before the clock is supplied to the ITC, as the cause-of-interrupt flag in the ITC is not set.

Furthermore, additional time is needed for the CPU to accept the interrupt request from the ITC, the CPU may execute a few instructions that follow the slp instruction before it starts the interrupt processing.

The same problem may occur when the CPU wakes up from SLEEP mode by NMI. No interrupt will occur if the #NMI signal is deasserted before the clock is supplied, as the NMI flag is not set.

- If the cause of interrupt used to restart from the standby mode has been set to invoke the IDMA, the IDMA is started up by that interrupt.

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.

- As the C33 PE Core 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.

- When the reset-only method is used to reset the cause-of-interrupt flag (by writing 1), if a read-modify-write instruction (e.g., bset, bclr, or bnot) is executed, the other cause-of-interrupt 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 cause-of-interrupt flag (by writing 0), all cause-of-interrupt flags for which 0 has been written are reset. When a read-modify-write operation is performed, a cause of interrupt 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.

- After an initial reset, the cause-of-interrupt 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.
- To prevent another interrupt from being generated for the same cause again after generation of an interrupt, be sure to reset the cause-of-interrupt flag before enabling interrupts and setting the PSR again or executing the reti instruction.

- There is a time lag between latching the interrupt signal and latching the interrupt vector and level signals caused by the interface specifications between the CPU and the ITC.

1. The CPU latches the interrupt signal sent from the ITC.

↓

2. The CPU latches the interrupt vector and level signals sent from the ITC.

↓

3. The CPU executes the interrupt handler.

An illegal interrupt exception (vector No. 11) occurs when a register related to the interrupt signal (ITC's interrupt enable and cause-of-interrupt flag registers) is altered before the CPU latches the interrupt vector and level signals (between Steps 1 and 2).

Therefore, it is very rare but an illegal interrupt exception may occur if an interrupt related register is altered when interrupts to the CPU are in enabled status (IE bit in PSR = 1).

However, the illegal interrupt exception that occurs does not affect the program execution if any processing is not performed in the exception handler.

To avoid an illegal interrupt exception occurring, disable interrupts to the CPU (set IE bit in PSR = 0) before altering an interrupt related register.

III.3 Real-Time Clock (RTC)

III.3.1 Overview of the RTC

The S1C33E08 incorporates a real-time clock (RTC) with a perpetual calendar, and an OSC1 oscillator circuit to generate the operating clock for the RTC.

The RTC and OSC1 oscillator circuit operate in SLEEP mode. Moreover, the RTC can periodically generate interrupt requests to the CPU.

The main features of the RTC are outlined below.

- Contains time counters (seconds, minutes, and hours) and calendar counters (days, days of the week, months, and year).
- BCD data can be read from and written to both counters.
- Capable of controlling the starting and stopping of time clocks.
- 24-hour or 12-hour mode can be selected.
- A 30-second correction function can be implemented in software.
- Periodic interrupts are possible.
- Interrupt period can be selected from 1/64 second, 1 second, 1 minute, or 1 hour, with selectable level/edge interrupts.
- Independent power supply, so that the RTC can continue operating even when system power is turned off. (Standby mode)
- A built-in OSC1 oscillator circuit (crystal oscillator or external clock input) that generates a 32.768-kHz (typ.) operating clock.

Figure III.3.1.1 shows a block diagram of the RTC.

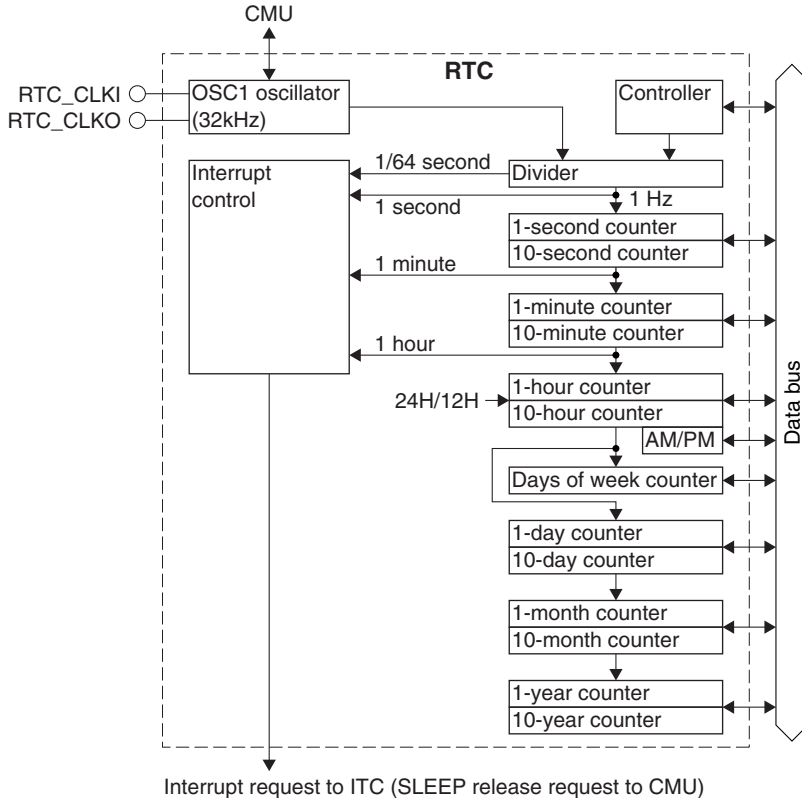


Figure III.3.1.1 RTC Block Diagram

III.3.2 RTC Counters

The RTC contains the following 13 counters, whose count values can be read out as BCD data from the respective registers. Each counter can also be set to any desired date and time by writing data to the respective register.

1-second counter

This 4-bit BCD counter counts in units of seconds. It counts from 0 to 9 synchronously with a 1-second signal derived from the 32.768-kHz OSC1 clock by dividing the clock into smaller frequencies. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-second counter. The count data is read out and written using RTCSL[3:0] (D[3:0]/0x301910).

* **RTCSL[3:0]**: RTC 1-second Counter Bits in the RTC Second Register (D[3:0]/0x301910)

10-second counter

This 3-bit BCD counter counts tens of seconds. It counts from 0 to 5 with 1 carried over from the 1-second counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-minute counter. The count data is read out and written using RTCSH[2:0] (D[6:4]/0x301910).

* **RTCSTH[2:0]**: RTC 10-second Counter Bits in the RTC Second Register (D[6:4]/0x301910)

1-minute counter

This 4-bit BCD counter counts in units of minutes. It counts from 0 to 9 with 1 carried over from the 10-second counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-minute counter. The count data is read out and written using RTCMIL[3:0] (D[3:0]/0x301914).

* **RTCMIL[3:0]**: RTC 1-minute Counter Bits in the RTC Minute Register (D[3:0]/0x301914)

10-minute counter

This 3-bit BCD counter counts tens of minutes. It counts from 0 to 5 with 1 carried over from the 1-minute counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-hour counter. The count data is read out and written using RTCMIH[2:0] (D[6:4]/0x301914).

* **RTCMIH[2:0]**: RTC 10-minute Counter Bits in the RTC Minute Register (D[6:4]/0x301914)

1-hour counter

This 4-bit BCD counter counts in units of hours. It counts from 0 to 9 with 1 carried over from the 10-minute counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-hour counter. Depending whether 12-hour or 24-hour mode is selected, the counter is reset at 12 o'clock or 24 o'clock. The count data is read out and written using RTCHL[3:0] (D[3:0]/0x301918).

* **RTCHL[3:0]**: RTC 1-hour Counter Bits in the RTC Hour Register (D[3:0]/0x301918)

10-hour counter

This 2-bit BCD counter counts tens of hours. With a carry over of 1 from the 1-hour counter, this counter counts from 0 to 1 (when 12-hour mode is selected) or from 0 to 2 (when 24-hour mode is selected). The counter is reset at 12 o'clock or 24 o'clock, and outputs a carry over of 1 to the 1-day counter. The count data is read out and written using RTCHH[1:0] (D[5:4]/0x301918).

* **RTCHH[1:0]**: RTC 10-hour Counter Bits in the RTC Hour Register (D[5:4]/0x301918)

When 12-hour mode is selected, RTCAP (D6/0x301918) that indicates A.M. or P.M. is enabled, with A.M. and P.M. represented by 0 and 1, respectively. For 24-hour mode, RTCAP (D6/0x301918) is fixed to 0.

* **RTCAP**: AM/PM Indicator Bit in the RTC Hour Register (D6/0x301918)

1-day counter

This 4-bit BCD counter counts in units of days. It counts from 0 to 9 with 1 carried over from the hour counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-day counter. The number of days in each month and leap years are taken into account, so that the counter is reset to 1 when months change. The count data is read out and written using RTCDL[3:0] (D[3:0]/0x30191C).

* **RTC DL[3:0]**: RTC 1-day Counter Bits in the RTC Day Register (D[3:0]/0x30191C)

10-day counter

This 2-bit BCD counter counts tens of days. It counts from 0 to 2 or 3 with 1 carried over from the 1-day counter. The number of days in each month and leap years are taken into account, so that when months change the counter is reset to 0 along with the 1-day counter, and outputs a carry over of 1 to the 1-month counter. The count data is read out and written using RTCDH[1:0] (D[5:4]/0x30191C).

* **RTC DH[1:0]**: RTC 10-day Counter Bits in the RTC Day Register (D[5:4]/0x30191C)

1-month counter

This 4-bit BCD counter counts in units of months. It counts from 0 to 9 with 1 carried over from the day counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-month counter. The counter is reset to 1 when years change. The count data is read out and written using RTCMOL[3:0] (D[3:0]/0x301920).

* **RTC MOL[3:0]**: RTC 1-month Counter Bits in the RTC Month Register (D[3:0]/0x301920)

10-month counter

This counter counts in units of 10 months, and is set to 1 with 1 carried over from the 1-month counter. When years change, this counter is reset to 0 along with the 1-month counter, and outputs a carry over of 1 to the 1-year counter. The count data is read out and written using RTCMOH (D4/0x301920).

* **RTC MOH**: RTC 10-month Counter Bit in the RTC Month Register (D4/0x301920)

1-year counter

This 4-bit BCD counter counts in units of years. It counts from 0 to 9 with 1 carried over from the month counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-year counter. The count data is read out and written using RTCYL[3:0] (D[3:0]/0x301924).

* **RTC YL[3:0]**: RTC 1-year Counter Bits in the RTC Year Register (D[3:0]/0x301924)

10-year counter

This 4-bit BCD counter counts tens of years. It counts from 0 to 9 with 1 carried over from the 1-year counter. The count data is read out and written using RTCYH[3:0] (D[7:4]/0x301924).

* **RTC YH[3:0]**: RTC 10-year Counter Bits in the RTC Year Register (D[7:4]/0x301924)

Days of week counter

This is a septenary counter (that counts from 0 to 6) representing the days of the week. It counts with the same timing as the 1-day counter. The count data is read out and written using RTCWK[2:0] (D[2:0]/0x301928).

* **RTC WK[2:0]**: RTC Days of Week Counter Bits in the RTC Days of Week Register (D[2:0]/0x301928)

The correspondence between the counter values and days of the week can be set in a program as desired. Table III.3.2.1 lists the basic correspondence.

Table III.3.2.1 Correspondence between Counter Values and Days of the Week

RTCWK2	RTCWK1	RTCWK0	Days of the week
1	1	0	Saturday
1	0	1	Friday
1	0	0	Thursday
0	1	1	Wednesday
0	1	0	Tuesday
0	0	1	Monday
0	0	0	Sunday

Initial counter values

When initially reset, the counter values are not initialized. After power-on, the counter values are indeterminate. Be sure to initialize the counters by following the procedure described in Section III.3.3.2, “Initial Sequence of the RTC.”

About detection of leap years

The algorithm used in the RTC to detect leap years is for Anno Domini (A.D.) only, and can automatically identify leap years up to the year 2399.

Years (0 to 99) without a remainder when divided by 4 are considered leap years. When the 1-year and 10-year counters both are 0, a common year is assumed.

III.3.3 Control of the RTC

III.3.3.1 Controlling the Operating Clock

Counter clock

The RTC is clocked by the 32.768-kHz (typ.) OSC1 clock. OSC1 oscillation can be turned on or off using SOSC1 (D0/0x301B08) of the CMU.

* **SOSC1**: Low-speed Oscillation (OSC1) On/Off Control Bit in the System Clock Control Register (D0/0x301B08)

To use the RTC, SOSC1 (D0/0x301B08) must be set to 1 (default) to turn the OSC1 oscillator circuit on and keep it running.

Note: If the OSC1 oscillator circuit is turned on while idle, a finite time (of about 3 seconds) is required for its oscillation to stabilize. Do not let the RTC start counting until this time elapses.

The OSC1 clock does not stop regardless of chip standby mode (HALT or SLEEP).

For details of clock control, see Section III.1, “Clock Management Unit (CMU).” For the configuration of the OSC1 oscillator circuit, see Section III.3.5, “OSC1 Oscillator Circuit.”

Register clock

The RTC_CLK clock (= MCLK) is used to operate the RTC control registers. To setup the registers, this clock is required. After the registers are set up, the clock supply can be stopped to reduce current consumption by setting RTCSAPB_CKE (D0/0x301B04) to 0.

* **RTCSAPB_CKE**: RTC SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 1 (D0/0x301B04)

III.3.3.2 Initial Sequence of the RTC

Immediately after power-on, the contents of RTC registers are indeterminate. After powering on, follow the procedure below to let the RTC start ticking the time. Later sections detail the contents of each control.

1. Power-on
2. System initialization processing and waiting for OSC1 stabilization
Although the OSC1 oscillator circuit starts oscillating immediately after power is switched on, a finite time of up to 3 seconds is required before the output clock stabilizes.
3. Disabling RTC interrupts
To prevent the occurrence of unwanted RTC interrupts, the following register settings are required:
Write 0x0 to the RTC Interrupt Mode Register (0x301904) to disable RTC interrupts.
Write 0x1 to the RTC Interrupt Status Register (0x301900) to clear the RTC interrupt status.
For details, see Section III.3.4, “RTC Interrupts.”
4. Starting the count
Write 0x2 (for 12-hour mode) or 0x12 (for 24-hour mode) to the RTC Control Register (0x301908) to start counting by the RTC. This operation initializes the contents of 12-hour/24-hour mode, etc. that affect count data when settings are changed, and is not the standard operation to start counting.
For details, see Section III.3.3.3, “Selecting 12/24-hour Mode and Setting the Counters,” and Section III.3.3.4, “Starting, Stopping, and Resetting Counters.”
5. Confirming accessibility status of the RTC
Use the RTC Access Control Register (0x30190C) to retain the counters intact and read out the busy flag to confirm that the RTC can now be accessed.
For details, see Section III.3.3.5, “Counter Hold and Busy Flag.”
6. Stopping and resetting the count
Write 0x1 to the RTC Control Register (0x301908) to stop the count, then reset the divide-by stage of the count clock.
For details, see Section III.3.3.4, “Starting, Stopping, and Resetting Counters.”
7. Setting the date and time
Use the respective count registers to initialize all counters to the current date and time.
For details, see Section III.3.3.3, “Selecting 12/24-hour Mode and Setting the Counters.”
8. Restarting count
Release the counters from the hold state (set in step 5) and repeat step 4 to restart counting by the RTC.
For details, see Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.4, “Starting, Stopping, and Resetting Counters.”

III.3.3.3 Selecting 12/24-hour Mode and Setting the Counters

Selecting 12-hour/24-hour mode

Whether to use the time clock in 12-hour or 24-hour mode can be selected using RTC24H (D4/0x301908).

RTC24H = 1: 24-hour mode

RTC24H = 0: 12-hour mode

The count range of hour counters changes with this selection.

* **RTC24H**: 24H/12H Mode Select Bit in the RTC Control Register (D4/0x301908)

Basically, this setting should be changed while the counters are idle. RTC24H (D4/0x301908) is allocated to the same address as the control bits that start the counters. Therefore, 12-hour mode or 24-hour mode can be selected at the same time the counters are started.

Note: Rewriting RTC24H (D4/0x301908) may corrupt count data for the hours, days, months, years or days of the week. Therefore, once RTC24H (D4/0x301908) settings are changed, be sure to set data back in these counters again.

Checking A.M./P.M. with 12-hour mode selected

When 12-hour mode is selected, RTCAP (D6/0x301918) that indicates A.M. or P.M. is enabled.

RTCAP = 0: A.M.

RTCAP = 1: P.M.

For 24-hour mode, RTCAP (D6/0x301918) is fixed to 0.

* **RTCAP**: AM/PM Indicator Bit in the RTC Hour Register (D6/0x301918)

When setting the time of day, write either of the values above to this bit to specify A.M. or P.M.

Setting the counters

Idle counters can be accessed for read or write at any time.

However, settings like those shown below should be avoided, since such settings may cause timekeeping errors.

- Settings exceeding the effective range
Do not set count data exceeding 60 seconds, 60 minutes, 12 or 24 hours, 31 days, 12 months, or 99 years.
- Settings nonexistent in the calendar
Do not set such nonexistent dates as April 31 or February 29, 2006. Even if such settings are made, the counters operate normally, so that when 1 is carried over from the hour counter to the 1-day counter, the day counter counts up to the first day of the next month. (For April 31, the day counter counts up to May 1; for February 29, 2006, the day counter counts up to March 1, 2006.)

If any counter must be rewritten while operating, there is a procedure that must be followed to ensure that the counter is rewritten correctly. For details, see Section III.3.3.6, "Reading from and Writing to Counters in Operation."

III.3.3.4 Starting, Stopping, and Resetting Counters

Starting and stopping counters

The RTC starts counting when RTCSTP (D1/0x301908) is set to 0, and stops counting when this bit is set to 1.

* **RTCSTP**: Counter Run/Stop Control Bit in the RTC Control Register (D1/0x301908)

The RTC is stopped by writing 1 to RTCSTP (D1/0x301908) at the 32-kHz input clock divide-by stage of 8,192 Hz or those stages that follow. The RTC does not stop at up to the input clock divide-by-2 stage (16,384 Hz).

If the RTC stops counting when 1 is carried over to the next-digit counter, the count value may be corrupted. Therefore, see the next section to ensure that 1 is not carried over when counters are made to stop. This is unnecessary, however, when the contents of all counters are newly set again.

Resetting the counters

RTCRST (D0/0x301908) is the bit used to reset the 32 kHz to 2 Hz counters.

* **RTCRST**: Software Reset Bit in the RTC Control Register (D0/0x301908)

Setting RTCRST (D0/0x301908) to 1 resets the counters above (cleared to 0), and writing 0 to this bit negates the reset.

III.3.3.5 Counter Hold and Busy Flag

If 1 is carried over when reading the counters, the correct counter value may not be read out. Moreover, if a write or stop operation is attempted, the counter value may be corrupted. Therefore, whether counters are in a carry (busy) state should be checked before reading or writing data from or to the count registers. For this purpose, control bits RTCBSY (D1/0x30190C) and RTCHLD (D0/0x30190C) are provided.

- * **RTCBSY**: Counter Busy Flag in the RTC Access Control Register (D1/0x30190C)
- * **RTCHLD**: Counter Hold Control Bit in the RTC Access Control Register (D0/0x30190C)

RTCBSY (D1/0x30190C) is a read-only flag indicating that 1 is being carried over. RTCBSY (D1/0x30190C) is set to 1 when 1 is being carried over; otherwise, it is 0. RTCBSY (D1/0x30190C) should be confirmed as being 0 before accessing the counters to ensure that the correct value will be read or set.

Note, however, that RTCBSY (D1/0x30190C) is fixed to 1 while counting is in progress. To reflect the current state in the count value, RTCHLD (D0/0x30190C) should be set to 1.

RTCBSY = 0 (RTC accessible)

If the value of RTCBSY (D1/0x30190C) is 0 when this bit is read out after writing 1 to RTCHLD (D0/0x30190C), it means that 1 is not being carried over. In this case, the counter hold function is actuated, with a carry over of 1 to the 1-second counter disabled in hardware. Counters that count less than seconds continue operating.

Data can be read from or written to the count registers in this state. After reading or writing data, reset RTCHLD (D0/0x30190C) to 0.

When 1 must be carried over while data is being read or written with counters in the hold state, 1 second is automatically added at the time, with RTCHLD (D0/0x30190C) reset to 0 to correct the count value. This correction is effective for only 1 second, and the time to carry over 1 on subsequent occasions is ignored. In this case, timekeeping data gets out of order. Therefore, be sure to reset RTCHLD (D0/0x30190C) to 0 as soon as possible after completing the necessary read or write operation.

RTCBSY = 1 (RTC is busy)

If the value of RTCBSY (D1/0x30190C) is 1 when this bit is read after writing 1 to RTCHLD (D0/0x30190C), it means that 1 is being carried over. The period needed for the counters to carry over 1 is 4 ms per second. In this case, reset RTCHLD (D0/0x30190C) to 0 as soon as possible and [A] recheck RTCBSY (D1/0x30190C) by following the same procedure or [B] wait 4 ms before checking RTCBSY (D1/0x30190C).

If RTCBSY (D1/0x30190C) is found to be 1, be sure to immediately reset RTCHLD (D0/0x30190C) to 0. If RTCHLD (D0/0x30190C) is left at 1, the time of day may become incorrect.

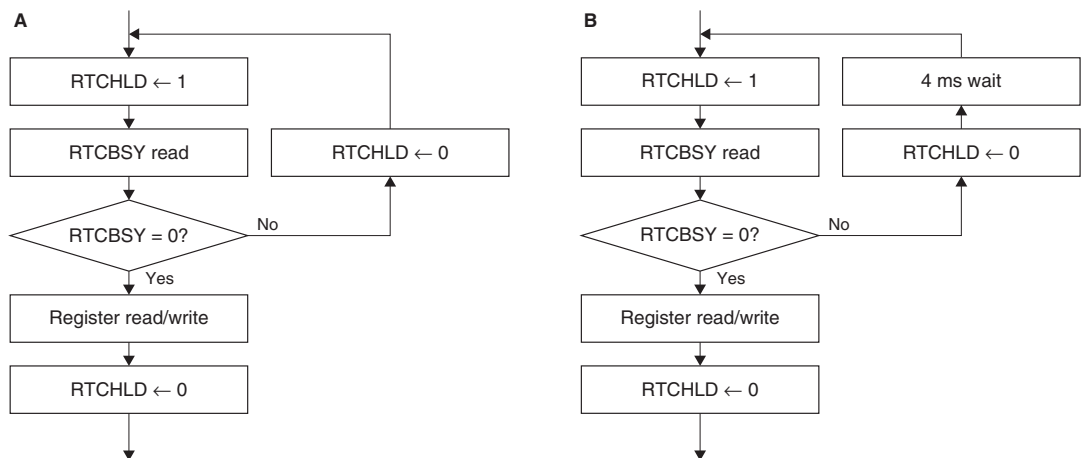


Figure III.3.3.5.1 Procedure for Checking whether the RTC is Busy

There is also a method of reading out data without using RTCHLD (D0/0x30190C) and RTCBSY (D1/0x30190C). (See the next section.)

III.3.3.6 Reading from and Writing to Counters in Operation

As described in the previous section, the counters must be accessed for read/write when 1 is not being carried over. Follow the procedure shown in the flowchart in Figure III.3.3.5.1 to read from or write to the counters.

The counters can be read without using RTCHLD (D0/0x30190C) and RTCBSY (D1/0x30190C), as shown in Figure III.3.3.6.1.

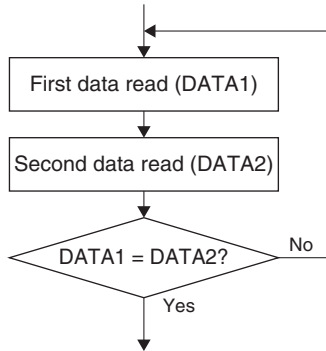


Figure III.3.3.6.1 Procedure for Reading Counters not in the Hold State

III.3.3.7 30-second Correction

The description “30-second correction” means adding 1 to the minutes when seconds of the time clock are in the range of 30 to 59 seconds, and doing nothing when in the range of 0 to 29 seconds. This function may be used to round up seconds to minutes when resetting seconds in an application.

This function can be executed by writing 1 to RTCADJ (D2/0x301908).

* **RTCADJ**: 30-second Adjustment Bit in the RTC Control Register (D2/0x301908)

Writing 1 to RTCADJ (D2/0x301908) causes the RTC to operate as follows:

- When the 10-second counter is 3 or more, the RTC generates a carry over of 1 to start counting by the 1-minute counter.
- When the 10-second counter is 2 or less, the RTC does not generate a carry over of 1.

After RTCADJ (D2/0x301908) is set to 1, it remains set for the 4-ms period required for this processing, then automatically returns to 0.

Accessing the counters while RTCADJ (D2/0x301908) = 1 is prohibited. Writing 0 to RTCADJ (D2/0x301908) is also prohibited, because it would cause the RTC to operate erratically.

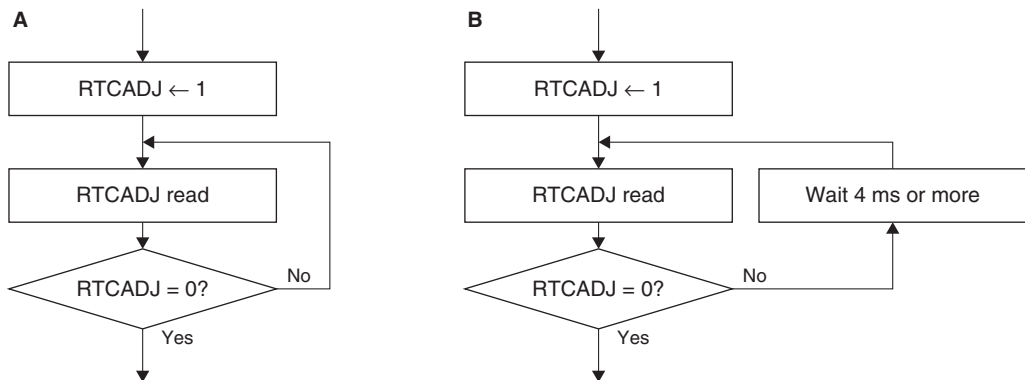


Figure III.3.3.7.1 Procedure for Executing 30-second Correction

III.3.4 RTC Interrupts

The RTC has a function to generate interrupts at given intervals.

Since the RTC is active even in standby mode, interrupts may be used to turn off SLEEP mode.

This section describes the internal interrupt control function of the RTC. To generate interrupts to the CPU, the ITC must also be set up. For details on how to control the ITC, see Section III.2, “Interrupt Controller (ITC).” For details on how to turn off SLEEP mode using an interrupt, see Section III.1, “Clock Management Unit (CMU).”

Setting the interrupt cycle

The interrupt cycle (in which the RTC outputs interrupt requests at specific intervals) can be selected from four choices listed in Table III.3.4.1 by using RTCT[1:0] (D[3:2]/0x301904).

* **RTCT[1:0]**: RTC Interrupt Cycle Setup Bits in the RTC Interrupt Mode Register (D[3:2]/0x301904)

Table III.3.4.1 Interrupt Cycle Settings

RTCT1	RTCT0	Interrupt cycle
1	1	1 hour
1	0	1 minute
0	1	1 second
0	0	1/64 second

RTCT[1:0] (D[3:2]/0x301904) should be set while RTC interrupts are disabled. (See the procedure for enabling and disabling interrupts described below.)

Setting interrupt conditions

The interrupt requests sent to the ITC can be selected as edge-triggered or level-sensed interrupts by setting a register bit. RTCIMD (D1/0x301904) is the bit provided for this purpose.

* **RTCIMD**: RTC Interrupt Mode Select Bit in the RTC Interrupt Mode Register (D1/0x301904)

Setting RTCIMD (D1/0x301904) to 1 selects a level-sensed interrupt; setting it to 0 selects an edge-triggered interrupt.

When an edge-triggered interrupt has been selected, the RTC outputs an interrupt pulse to the ITC using the bus clock supplied from the CMU. If a cause of interrupt occurs when the bus clock has not been supplied such as in SLEEP mode, the RTC switches the interrupt mode to level-sensed and sets the interrupt signal to the active level from occurrence of the interrupt cause until the bus clock supply is started.

Enabling and disabling interrupts

The RTC interrupt requests output to the ITC are enabled by setting RTCIEN (D0/0x301904) to 1 and disabled by setting it to 0.

* **RTCIEN**: RTC Interrupt Enable Bit in the RTC Interrupt Mode Register (D0/0x301904)

Interrupt status

When the RTC is up and running, RTCIRQ (D0/0x301900) is set at the cyclic interrupt intervals set up by RTCT[1:0]. When RTC interrupts are enabled by RTCIEN (D0/0x301904), interrupt requests are sent to the ITC.

* **RTCIRQ**: Interrupt Status Bit in the RTC Interrupt Status Register (D0/0x301900)

Writing 1 to this status bit clears the bit. Because this bit is not cleared in hardware, be sure to clear it in software after an interrupt is generated. If this bit remains set while interrupts are re-enabled or control is returned from the interrupt handler routine by the reti instruction, the same interrupt may be generated again.

Precautions

All RTC interrupt control bits described above are indeterminate when power is turned on. Moreover, these bits are not initialized to specific values by an initial reset.

After power-on, be sure to set RTCIEN (D0/0x301904) to 0 (interrupt disabled) to prevent the occurrence of unwanted RTC interrupts. Also be sure to write 1 to RTCIRQ (D0/0x301900) to reset it.

III.3.5 OSC1 Oscillator Circuit

The S1C33E08 contains an oscillator circuit (OSC1) used to generate a 32.768 kHz (typ.) clock as the clock source for timekeeping operation of the RTC.

The OSC1 clock can also be used as a power-saving operating clock for the core system or peripheral circuits. For details, see Section III.1, “Clock Management Unit (CMU).”

III.3.5.1 Input/Output Pins of the OSC1 Oscillator Circuit

Table III.3.5.1.1 lists the input/output pins of the OSC1 oscillator circuit.

Table III.3.5.1.1 Input/Output Pins of the Low-speed (OSC1) Oscillator Circuit

Pin name	I/O	Function
RTC_CLKI	I	OSC1 input pin: Crystal oscillator or external clock input
RTC_CLKO	O	OSC1 output pin: Crystal oscillator output (left open when external clock is input)

III.3.5.2 Structure of the OSC1 Oscillator Circuit

The OSC1 oscillator circuit accommodates a crystal oscillator and external clock input. As for the RTC, V_{DD} is used to supply power to this circuit.

Figure III.3.5.2.1 shows the structure of the OSC1 oscillator circuit.

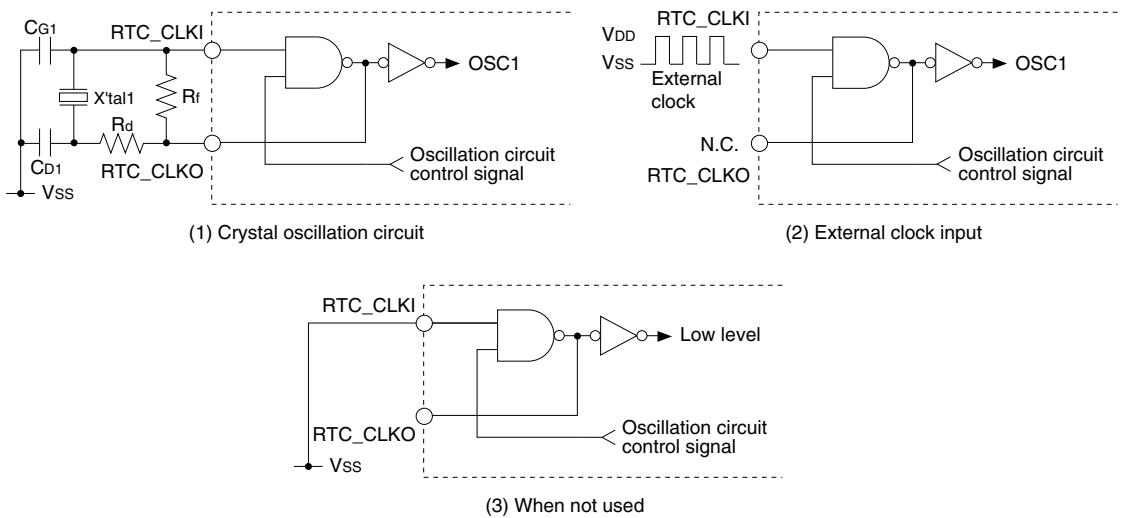


Figure III.3.5.2.1 OSC1 Oscillator Circuit

For use as a crystal oscillator circuit, connect a crystal resonator X'tal1 (32.768 kHz, typ.), feedback resistor (R_f), two capacitors (C_{G1}, C_{D1}), and, if necessary, a drain resistor (R_d) to the RTC_CLKI and RTC_CLKO pins and V_{SS}, as shown in Figure III.3.5.2.1 (1).

To use an external clock, leave the RTC_CLKO pin open and input a V_{DD} level clock (whose duty cycle is 50%) to the RTC_CLKI pin. Do not input V_{DDH} or other I/O level clocks.

The oscillator frequency/input clock frequency is 32.768 kHz (typ.). Make sure the crystal resonator or external clock used in the RTC has this clock frequency. With any other clock frequencies, the RTC cannot be used for timekeeping purposes.

For details of oscillation characteristics and the input characteristics of external clock, see “Electrical Characteristics.”

When not using the OSC1 oscillator circuit, connect the RTC_CLKI pin to V_{SS} and leave the RTC_CLKO pin open.

III.3.5.3 Oscillation Control

Internal control bit SOSC1 (D0/0x301B08) of the CMU register is used to control OSC1 oscillation.

* **SOSC1**: Low-speed Oscillation (OSC1) On/Off Control Bit in the System Clock Control Register (D0/0x301B08)

Setting this control bit to 0 causes the OSC1 oscillator circuit to stop; setting it to 1 causes the OSC1 oscillator circuit to start oscillating, thereby outputting a clock signal waveform. When initially reset, this bit is set to 1, so that the OSC1 oscillator circuit continues oscillating.

- Notes:**
- The System Clock Control Register (0x301B08) is write-protected. The write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96 unless rewriting the said registers.
 - When the oscillator is made to start oscillating by setting SOSC1 (D0/0x301B08) from 0 to 1, a finite time (of up to 3 seconds) is required until oscillation stabilizes. To prevent system malfunction, do not use the oscillator-derived clock until this oscillation stabilization time elapses.

III.3.6 Details of Control Registers

Table III.3.6.1 RTC Register List

Address	Register name	Size	Function
0x00301900	RTC Interrupt Status Register (pRTCINTSTAT)	32	RTC interrupt status
0x00301904	RTC Interrupt Mode Register (pRTCINTMODE)	32	Sets RTC interrupt conditions and enables RTC interrupts.
0x00301908	RTC Control Register (pRTC_CNTL0)	32	Controls RTC operation.
0x0030190C	RTC Access Control Register (pRTC_CNTL1)	32	Controls RTC busy status and counter hold.
0x00301910	RTC Second Register (pRTCSEC)	32	Seconds counter data
0x00301914	RTC Minute Register (pRTCMIN)	32	Minutes counter data
0x00301918	RTC Hour Register (pRTCHOUR)	32	Hours counter data
0x0030191C	RTC Day Register (pRTCDAAY)	32	Days counter data
0x00301920	RTC Month Register (pRTCMONTH)	32	Months counter data
0x00301924	RTC Year Register (pRTCYEAR)	32	Years counter data
0x00301928	RTC Days of Week Register (pRTCDAYWEEK)	32	Days of the week counter data

Each RTC control register is described below.

The RTC control registers are mapped as 32-bit devices to Area 6 at addresses 0x301900 to 0x301928, and can be accessed in units of words, half-words, or bytes.

- Notes:**
- The contents of all RTC control registers are indeterminate when power is turned on, and are not initialized to specific values by initial reset. These registers should be initialized in software.
 - If 1 is being carried over when the counters are accessed for read, the correct counter value may not be read out. Moreover, attempting to write to a counter or other control register may corrupt the counter value. Therefore, do not write to counters while 1 is being carried over. For the correct method of operation, see Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”
 - To access the RTC control registers, the number of wait cycles must be set up in the Misc register. For setting details, see Section III.4, “Misc Registers.”
 - For details of RTC-related registers in the CMU and ITC mentioned here, see the following sections:
 - Section III.1, “Clock Management Unit (CMU)”
 - Section III.2, “Interrupt Controller (ITC)”

0x301900: RTC Interrupt Status Register (pRTCINTSTAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC interrupt status register (pRTCINTSTAT)	00301900 (W)	D31-1	–	reserved	–	–	–	0 when being read.
		D0	RTCIRQ	Interrupt status	1 Occurred 0 Not occurred	X	R/W	Reset by writing 1.

D[31:1] Reserved

D0 RTCIRQ: Interrupt Status Bit

This bit indicates whether a cause of RTC interrupt occurred as follows:

- 1 (R): Cause of interrupt occurred
- 0 (R): No cause of interrupt occurred
- 1 (W): Resets this bit to 0
- 0 (W): Has no effect

This bit is set at cyclic interrupt intervals set up by RTCT[1:0] (D[3:2]/0x301904). When RTC interrupts have been enabled by RTCIEN (D0/0x301904) at this time, an interrupt request is sent to the ITC. This bit is always set, even when RTC interrupts are disabled.

Note: Writing 1 to this status bit clears it. Because this bit is not cleared in hardware, be sure to clear it in software after an interrupt is generated. If this bit remains set while interrupts are re-enabled or control is returned from the interrupt handler routine by the `reti` instruction, the same interrupt may be generated again.

Moreover, the value of this bit is indeterminate after power-on, and is not initialized to 0 by initial reset. To prevent the occurrence of unwanted RTC interrupts, be sure to reset this bit in software after power-on and initial reset.

0x301904: RTC Interrupt Mode Register (pRTCINTMODE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC interrupt mode register (pRTCINTMODE)	00301904 (W)	D31–4	–	reserved	–	–	–	0 when being read.
		D3	RTCT1	RTC interrupt cycle setup	RTCT[1:0]	Cycle	X	R/W
		D2	RTCT0		11	1 hour	X	
					10	1 minute		
					01	1 second		
		00	1/64 second					
		D1	RTCIMD	RTC interrupt mode select	1 Level sense 0 Edge trigger	X	R/W	
		D0	RTCEN	RTC interrupt enable	1 Enabled 0 Disabled	X	R/W	

D[31:4] Reserved

D[3:2] RTCT[1:0]: RTC Interrupt Cycle Setup Bits

These bits select the RTC interrupt cycle.

Table III.3.6.2 Interrupt Cycle Settings

RTCT1	RTCT0	Interrupt cycle
1	1	1 hour
1	0	1 minute
0	1	1 second
0	0	1/64 second

(Default: indeterminate)

RTCIRQ (D0/0x301900) is set by a count-up pulse of the interrupt cycle counter selected. When RTC interrupts are enabled by RTCEN (D0), an interrupt request is sent to the ITC.

RTCT[1:0] should be set while RTC interrupts are disabled. (These bits may also be set simultaneously when RTC interrupts are enabled.)

D1 RTCIMD: RTC Interrupt Mode Select Bit

This bit specifies whether RTC interrupts are to be generated by an edge or level of the interrupt request signal.

1 (R/W): Level sensed

0 (R/W): Edge triggered

When an edge-triggered interrupt is selected and used to turn off SLEEP mode via the CMU, note that no interrupts will be generated because the ITC is inactive. When an RTC interrupt handler routine must be executed after exiting SLEEP mode, select a level-sensed interrupt.

D0 RTCEN: RTC Interrupt-Enable Bit

This bit enables or disables RTC interrupt request output to the ITC.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

To generate an RTC interrupt or use an RTC interrupt request signal to turn off SLEEP mode, set this bit to 1. When this bit is 0, no interrupts are generated even when RTCIRQ (D0/0x301900) is set and SLEEP mode cannot be turned off.

Note: The value of RTCEN is indeterminate after power-on, and not initialized to 0 by initial reset. To prevent the occurrence of unwanted RTC interrupts, be sure to clear this bit in software after power-on and initial reset.

0x301908: RTC Control Register (pRTC_CNTL0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC control register (pRTC_CNTL0)	00301908 (W)	D31-5	–	reserved		–	–	0 when being read.
		D4	RTC24H	24H/12H mode select	1 24H 0 12H	X	R/W	
		D3	–	reserved		–	–	
		D2	RTCADJ	30-second adjustment	1 Adjust 0 –	X	R/W	
		D1	RTCSTP	Counter run/stop control	1 Stop 0 Run	X	R/W	
		D0	RTCST	Software reset	1 Reset 0 –	X	R/W	

D[31:5] Reserved

D4 RTC24H: 24H/12H Mode Select Bit

This bit selects whether to use the hour counter in 24-hour or 12-hour mode.

1 (R/W): 24-hour mode

0 (R/W): 12-hour mode

The count range of hour counters changes with this selection.

Basically, this setting should be changed while the counters are idle. Since this register is assigned a control bit (D1) to start the counters, 12-hour or 24-hour mode may be selected when starting the counters.

Note: Rewriting RTC24H may corrupt the count data for hours, days, months, years, or days of the week. Therefore, after changing the RTC24H setting, be sure to set data back in these counters again.

D3 Reserved

D2 RTCADJ: 30-second Adjustment Bit

This bit executes 30-second correction.

1 (W): Execute 30-second correction

0 (W): Has no effect

1 (R): 30-second correction being executed

0 (R): 30-second correction completed (not being executed)

The description “30-second correction” means adding 1 to the minutes when seconds of the time clock are in the 30-to-59 second range, and doing nothing in the 0-to-29 second range. This function may be used to round up seconds to minutes when resetting seconds in an application.

Writing 1 to this bit causes the RTC to operate as follows:

- When the 10-second counter is 3 or more, the RTC generates a carry over of 1 to start counting by the 1-minute counter.
- When the 10-second counter is 2 or less, the RTC does not generate a carry over of 1.

After being set to 1, this bit remains set for the 4-ms period needed for the processing above, then is automatically reset to 0.

Note: Accessing the counters while RTCADJ = 1 is prohibited. Writing 0 to this bit during such time is also prohibited, because it would cause the RTC to operate erratically.

D1 RTCSTP: Counter Run/Stop Control Bit

This bit starts or stops the counters. It also indicates counter operating status.

1 (R/W): Stops counters/Counters idle

0 (R/W): Starts counters/Counters operating

Setting this bit to 0 starts the counters; setting it to 1 stops the counters.

The value read from this bit is 0 when the counters are operating, and 1 when the counters are idle.

Writing 1 to this bit stops the counters at the 32-kHz input clock divide-by stage of 8,192 Hz or stages that follow. The counters do not stop at up to the input clock divide-by-2 stage (16,384 Hz).

If the counters stop while 1 is being carried over, the count value may be corrupted. Therefore, see Section III.3.3.5 to ensure that 1 is not being carried over when the counters are stopped. This is unnecessary when, for example, the contents of all counters are newly set again.

D0 RTCRST: Software Reset Bit

This bit resets the counters currently at divide-by stages.

1 (R/W): Reset counters

0 (R/W): Negate reset

Setting this bit to 1 resets the 32 kHz to 2 Hz counters (cleared to 0). Writing 0 to this bit negates the reset.

0x30190C: RTC Access Control Register (pRTC_CNTL1)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
RTC access control register (pRTC_CNTL1)	0030190C (W)	D31-2	–	reserved	–			–	–	0 when being read.	
		D1	RTCBSY	Counter busy flag	1	Busy	0	R/W possible	X	R	
		D0	RTCHLD	Counter hold control	1	Hold	0	Running	X	R/W	

D[31:2] Reserved

D1 **RTCBSY: Counter Busy Flag**

This flag indicates whether 1 is being carried over to the next-digit counter.

1 (R): Busy (while 1 is being carried over)

0 (R): Accessible for read/write

1/0 (W): Has no effect

If 1 is being carried over while the counters are being read, correct counter values may not be read. Moreover, attempting a write or stop operation may corrupt the counter values. Therefore, this bit should be checked to confirm that the counters are not in a carry (busy) state before reading or writing data from or to the count registers.

However, because this bit is fixed to 1 while the counters are operating, RTCHLD (D0) should be set to 1 so that the count value reflects the current state.

When a value of 0 is read from this bit after writing 1 to RTCHLD (D0), it means that 1 is not now being carried over. In this case, the counter hold function is also actuated, with a carry over of 1 to the 1-second counter disabled in hardware. Counters for less than seconds continue operating. In this state, data can be read from or written to the count registers. After reading or writing data, reset RTCHLD (D0) to 0.

If 1 is being carried over when data is being read from or written to counters in the hold state, 1 second is automatically added at that time, with RTCHLD (D0) reset to 0 for correcting the count value. This correction is only effective for 1 second, thus ignoring the time needed to carry over 1 on subsequent occasions. In this case, the timekeeping data gets out of order. Therefore, be sure to reset RTCHLD (D0) to 0 as soon as possible after completing the required read or write operation.

When a value of 1 is read from this bit after writing 1 to RTCHLD (D0), it means that 1 is now being carried over. A period of 4 ms per second is required for a carry over of 1 to the counters. In this case, reset RTCHLD (D0) to 0 as soon as possible and check this bit again by following the same procedure, or wait 4 ms before checking this bit. If this bit is set to 1, always reset RTCHLD (D0) to 0 immediately. Leaving RTCHLD (D0) set to 1 may result in an incorrect time of day.

D0 **RTCHLD: Counter Hold Control Bit**

This bit allows the busy state of counters to be checked and the counters held intact.

1 (R/W): Checks for busy state/Holds counters

0 (R/W): Normal operation

For the operation of this bit, see the description of RTCBSY (D1) above.

0x301910: RTC Second Register (pRTCSEC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC second register (pRTCSEC)	00301910 (W)	D31–7	–	reserved	–	–	–	0 when being read.
		D6	RTCSH2	RTC 10-second counter	0 to 5	X	R/W	
		D5	RTCSH1					
		D4	RTCSH0					
		D3	RTCSL3	RTC 1-second counter	0 to 9	X	R/W	
		D2	RTCSL2					
		D1	RTCSL1					
		D0	RTCSL0					

Note: Data should not be read from or written to the counters while 1 is being carried over. (See Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”)

D[31:7] Reserved

D[6:4] RTCSH[2:0]: RTC 10-second Counter Bits

These bits comprise a 3-bit BCD counter used to count tens of seconds.

The counter counts from 0 to 5 with a carry over of 1 from the 1-second counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-minute counter.

D[3:0] RTCSL[3:0]: RTC 1-second Counter Bits

These bits comprise a 4-bit BCD counter used to count units of seconds.

The counter counts from 0 to 9 synchronously with a 1-second signal derived from the 32.768-kHz OSC1 clock. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-second counter.

0x301914: RTC Minute Register (pRTCMin)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC minute register (pRTCMin)	00301914 (W)	D31-7	–	reserved	–	–	–	0 when being read.
		D6	RTCMIH2	RTC 10-minute counter	0 to 5	X	R/W	
		D5	RTCMIH1					
		D4	RTCMIH0					
		D3	RTCMIL3	RTC 1-minute counter	0 to 9	X	R/W	
		D2	RTCMIL2					
		D1	RTCMIL1					
		D0	RTCMIL0					

Note: Data should not be read from or written to the counters while 1 is being carried over. (See Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”)

D[31:7] Reserved

D[6:4] RTCMIH[2:0]: RTC 10-minute Counter Bits

These bits comprise a 3-bit BCD counter used to count tens of minutes. The counter counts from 0 to 5 with a carry over of 1 from the 1-minute counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-hour counter.

D[3:0] RTCMIL[3:0]: RTC 1-minute Counter Bits

These bits comprise a 4-bit BCD counter used to count units of minutes. The counter counts from 0 to 9 with a carry over of 1 from the 10-second counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-minute counter.

0x301918: RTC Hour Register (pRTCHOUR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC hour register (pRTCHOUR)	00301918 (W)	D31–7	–	reserved	–	–	–	0 when being read.
		D6	RTCAP	AM/PM indicator	1 PM 0 AM	X	R/W	
		D5	RTCHH1	RTC 10-hour counter	0 to 2 or 0 to 1	X	R/W	
		D4	RTCHH0			X		
		D3	RTCHL3	RTC 1-hour counter	0 to 9	X	R/W	
		D2	RTCHL2			X		
		D1	RTCHL1			X		
D0	RTCHL0	X						

Notes:

- Data should not be read from or written to the counters while 1 is being carried over. (See Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”)

- Rewriting RTC24H (D4/0x301908) may corrupt the count data in this register. Therefore, after changing the RTC24H (D4/0x301908) setting, be sure to set up this register again.

D[31:7] Reserved**D6 RTCAP: AM/PM Indicator Bit**

When 12-hour mode is selected, this bit indicates A.M. or P.M.

1 (R/W): P.M.

0 (R/W): A.M.

This bit is only effective when RTC24H (D4/0x301908) is set to 0 (12-hour mode).

When 24-hour mode is selected, this bit is fixed to 0. In this case, do not write 1 to RTCAP.

Note: The RTCAP bit keeps the current set value even if RTC24H (D4/0x301908) is changed from 12-hour mode to 24-hour mode, and will be fixed at 0 after the hour counter is updated (or reset in software).

D[5:4] RTCHH[1:0]: RTC 10-hour Counter Bits

These bits comprise a 2-bit BCD counter used to count tens of hours.

With a carry over of 1 from the 1-hour counter, the counter counts from 0 to 1 when 12-hour mode is selected, or from 0 to 2 when 24-hour mode is selected. The counter is reset at 12 0'clock or 24 0'clock, and outputs a carry over of 1 to the 1-day counter.

D[3:0] RTCHL[3:0]: RTC 1-hour Counter Bits

These bits comprise a 4-bit BCD counter used to count units of hours.

The counter counts from 0 to 9 with a carry over of 1 from the 10-minute counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-hour counter. Depending on whether 12-hour mode or 24-hour mode is selected, the counter is reset at 12 0'clock or 24 0'clock.

0x30191C: RTC Day Register (pRTCDA Y)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC day register (pRTCDA Y)	0030191C (W)	D31-6	–	reserved	–	–	–	0 when being read.
		D5	RTCDH1	RTC 10-day counter	0 to 3	X	R/W	
		D4	RTCDH0					
		D3	RTCDL3	RTC 1-day counter	0 to 9	X	R/W	
		D2	RTCDL2					
D1	RTCDL1							
D0	RTCDL0							

Notes:

- Data should not be read from or written to the counters while 1 is being carried over. (See Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”)

- Rewriting RTC24H (D4/0x301908) may corrupt the count data in this register. Therefore, after changing the RTC24H (D4/0x301908) setting, be sure to set up this register again.

D[31:6] Reserved

D[5:4] RTCDH[1:0]: RTC 10-day Counter Bits

These bits comprise a 2-bit BCD counter used to count tens of days. The counter counts from 0 to 2 or 3 with a carry over of 1 from the 1-day counter. The number of days in each month and leap years are taken into account, so that when months change the counter is reset to 0 along with the 1-day counter, and a carry over of 1 is output to the 1-month counter.

D[3:0] RTCDL[3:0]: RTC 1-day Counter Bits

These bits comprise a 4-bit BCD counter used to count units of days. The counter counts from 0 to 9 with a carry over of 1 from the hour counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-day counter. The number of days in each month and leap years are taken into account, so that the counter is reset to 1 when months change.

0x301920: RTC Month Register (pRTCMONTH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC month register (pRTCMONTH)	00301920 (W)	D31–5	–	reserved	–	–	–	0 when being read.	
		D4	RTCMOH	RTC 10-month counter	0 or 1	X	R/W		
		D3	RTCMOL3	RTC 1-month counter		0 to 9	X	R/W	
		D2	RTCMOL2				X		
		D1	RTCMOL1				X		
D0	RTCMOLO	X							

- Notes:**
- Data should not be read from or written to the counters while 1 is being carried over. (See Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”)
 - Rewriting RTC24H (D4/0x301908) may corrupt the count data in this register. Therefore, after changing the RTC24H (D4/0x301908) setting, be sure to set up this register again.

D[31:5] Reserved**D4 RTCMOH: RTC 10-month Counter Bit**

This is a tens of months count bit.

This bit is set to 1 with a carry over of 1 from the 1-month counter. When years change, this bit is reset to 0 along with the 1-month counter, and a carry over of 1 is output to the 1-year counter.

D[3:0] RTCMOL[3:0]: RTC 1-month Counter Bits

These bits comprise a 4-bit BCD counter used to count units of months.

The counter counts from 0 to 9 with a carry over of 1 from the day counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-month counter. The counter is reset to 1 when years change.

0x301924: RTC Year Register (pRTCYEAR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC year register (pRTCYEAR)	00301924 (W)	D31-8	–	reserved		–	–	0 when being read.
		D7	RTCYH3	RTC 10-year counter	0 to 9	X	R/W	
		D6	RTCYH2					
		D5	RTCYH1					
		D4	RTCYH0					
		D3	RTCYL3	RTC 1-year counter	0 to 9	X	R/W	
		D2	RTCYL2			X		
		D1	RTCYL1			X		
		D0	RTCYL0			X		

- Notes:**
- Data should not be read from or written to the counters while 1 is being carried over. (See Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”)
 - Rewriting RTC24H (D4/0x301908) may corrupt the count data in this register. Therefore, after changing the RTC24H (D4/0x301908) setting, be sure to set up this register again.

D[31:8] Reserved

D[7:4] RTCYH[3:0]: RTC 10-year Counter Bits

These bits comprise a 4-bit BCD counter used to count tens of years. The counter counts from 0 to 9 with a carry over of 1 from the 1-year counter.

D[3:0] RTCYL[3:0]: RTC 1-year Counter Bits

These bits comprise a 4-bit BCD counter used to count units of years.

The counter counts from 0 to 9 with a carry over of 1 from the month counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-year counter.

0x301928: RTC Days of Week Register (pRTCDAWEEK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC days of week register (pRTCDAWEEK)	00301928 (W)	D31-3	–	reserved	–	–	–	0 when being read.	
		D2	RTCWK2	RTC days of week counter	RTCWK[2:0]	Days of week	X	R/W	
		D1	RTCWK1		111	–	X		
		D0	RTCWK0		110	Saturday	X		
		101	Friday						
		100	Thursday						
		011	Wednesday						
		010	Tuesday						
		001	Monday						
		000	Sunday						

- Notes:**
- Data should not be read from or written to the counters while 1 is being carried over. (See Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”)
 - Rewriting RTC24H (D4/0x301908) may corrupt the count data in this register. Therefore, after changing the RTC24H (D4/0x301908) setting, be sure to set up this register again.

D[31:3] Reserved

D[2:0] RTCWK[2:0]: RTC Days of Week Counter Bits

This is a septenary counter (that counts from 0 to 6) representing days of the week. This counter counts at the same timing as the 1-day counter.

The correspondence between the counter values and days of the week can be set in a program as desired. Table III.3.6.3 lists the basic correspondence.

Table III.3.6.3 Correspondence between Counter Values and Days of the Week

RTCWK2	RTCWK1	RTCWK0	Days of the week
1	1	0	Saturday
1	0	1	Friday
1	0	0	Thursday
0	1	1	Wednesday
0	1	0	Tuesday
0	0	1	Monday
0	0	0	Sunday

(Default: indeterminate)

III.3.7 Precautions

- The contents of all RTC control registers are indeterminate when power is turned on and are not initialized to specific values by initial reset. Be sure to initialize these registers in software.
- While 1 is being carried over to the next-digit counter, the correct counter value may not be read out. Moreover, attempting to write to the counters or other control registers may corrupt the counter value. Therefore, do not write to the counters while 1 is being carried over. For the correct method of operation, see Section III.3.3.5, “Counter Hold and Busy Flag,” and Section III.3.3.6, “Reading from and Writing to Counters in Operation.”
- Note that rewriting RTC24H (D4/0x301908) to switch between 12-hour mode and 24-hour mode may corrupt the count data for hours, days, months, years, or days of the week. Therefore, after changing the RTC24H (D4/0x301908) setting, be sure to set data in these counters back again.
- Avoid the settings below that may cause timekeeping errors.
 - Settings exceeding the effective range
Do not set count data exceeding 60 seconds, 60 minutes, 12 or 24 hours, 31 days, 12 months, or 99 years.
 - Settings nonexistent in the calendar
Do not set nonexistent dates such as April 31 or February 29, 2006. Even if such settings are made, the counters operate normally, so that when 1 is carried over from the hour counter to the 1-day counter, the day counter counts up to the first day of the next month. (For April 31, the day counter counts up to May 1; for February 29, 2006, the day counter counts up to March 1, 2006.)
- The contents of all RTC interrupt control bits are indeterminate when power is turned on, and are not initialized to specific values by initial reset.
After power-on, be sure to set RTCIEN (D0/0x301904) to 0 (interrupt disabled) for preventing the occurrence of unwanted RTC interrupts. Also be sure to write 1 to RTCIRQ (D0/0x301900) to reset it.
- Immediately after the OSC1 oscillator circuit is activated (as at power-on), a finite time (of about 3 seconds) is required for OSC1 oscillation to stabilize. Do not let the RTC start counting until this time elapses.

III.4 Misc Registers

Note: The Misc registers at addresses 0x300010–0x30001A are write-protected. Before the Misc registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to the Misc Protect Register (0x300020). Note that since unnecessary rewrites to addresses 0x300010–0x30001A could lead to erratic system operation, the Misc Protect Register (0x300020) should be set to other than 0x96 unless said Misc registers must be rewritten.

III.4.1 RTC and USB Wait Control Registers

III.4.1.1 Setting Wait Cycles for Accessing the RTC

The RTC Wait Control Register (0x300010) contains the control bits RTCWT[2:0] (D[2:0]) used to set the number of wait cycles to be inserted when accessing the RTC registers.

* **RTCWT[2:0]:** RTC Register Access Wait Control Bits in the RTC Wait Control Register (D[2:0]/0x300010)

Table III.4.1.1.1 Number of Wait Cycles during RTC Access

RTCWT2	RTCWT1	RTCWT0	Number of wait cycles (in units of MCLK cycles)
1	1	1	7 cycles
1	1	0	6 cycles
1	0	1	5 cycles
1	0	0	4 cycles
0	1	1	3 cycles
0	1	0	2 cycles
0	0	1	1 cycle
0	0	0	0 cycles

(Default: 0b111 = 7 cycles)

The number of wait cycles should be set according to the MCLK clock frequency.

The S1C33E08 is able to operate with $RTCWT[2:0] \geq 1$.

III.4.1.2 Settings for the USB

The USB Wait Control Register (0x300012) contains the control bits USBWT[2:0] (D[2:0]) used to set the number of wait cycles to be inserted when accessing the USB registers.

* **USBWT[2:0]:** USB Register Access Wait Control Bits in the USB Wait Control Register (D[2:0]/0x300012)

Table III.4.1.2.1 Number of Wait Cycles during USB Access

USBWT2	USBWT1	USBWT0	Number of wait cycles (in units of MCLK cycles)	MCLK clock frequency
1	1	1	7 cycles	60 MHz or less
1	1	0	6 cycles	56 MHz or less
1	0	1	5 cycles	45 MHz or less
1	0	0	4 cycles	36 MHz or less
0	1	1	3 cycles	24 MHz or less
0	1	0	2 cycles	16 MHz or less
0	0	1	1 cycle	8 MHz or less
0	0	0	0 cycles	8 MHz or less

(Default: 0b111 = 7 cycles)

The number of wait cycles should be set according to the MCLK clock frequency.

Also the USB Wait Control Register (0x300012) contains the USBSNZ bit (D5) that controls Snooze mode for the USB function controller. Setting USBSNZ (D5/0x300012) to 1 enables Snooze mode.

* **USBSNZ:** USB Snooze Control Bit in the USB Wait Control Register (D5/0x300012)

Refer to Section IX.1, “USB Function Controller,” for details on control of the USB function controller.

III.4.2 Debug Port MUX Register

The P15–P17 and P34–P36 pins are shared with the PC trace debugging function and other peripheral functions. These pins are configured as the debug pins DST0 (P15), DST1 (P16), DPCO (P17), DSIO (P34), DCLK (P35), and DST2 (P36) at initial reset. When using these pins as GPIO or other peripheral function pins, TRCMUX (D0/0x300014) must be set to 0.

* **TRCMUX**: P15–17, P34–36 Debug Function Select Bit in the Debug Port MUX Register (D0/0x300014)

When TRCMUX (D0/0x300014) is set to 0, the port function select bits for P15–P17 and P34–P36 are enabled to select the pin function. Set TRCMUX (D0/0x300014) to 1 to use these pins for debugging.

III.4.3 Boot Register

The Boot Register (0x300018) is used to confirm the boot device and configure #CE10 boot conditions. BOOT[3:0] (D[7:4]/0x300018) indicates the boot device that has been specified by the BOOT1 and BOOT0 pins.

* **BOOT[3:0]**: Boot Mode Indicator Bits in the Boot Register (D[7:4]/0x300018)

Table III.4.3.1 BOOT[3:0] Bits

BOOT3	BOOT2	BOOT1	BOOT0	Boot mode
1	0	0	0	SPI boot
0	1	0	0	NOR Flash/external ROM boot
0	0	1	0	Reserved
0	0	0	1	NAND Flash boot

The Boot Register (0x300018) contains two more control bits, BOOT_ENA (D1) and CE10_SIZE (D0) that are used for the booting process by the internal boot firmware.

* **BOOT_ENA**: #CE10 Area Boot Enable Bit in the Boot Register (D1/0x300018)

* **CE10_SIZE**: #CE10 Area Size Select Bit in the Boot Register (D0/0x300018)

Note: When programming a Flash memory on the target board, BOOT_ENA (D1/0x300018) must be set to 0. Be sure to avoid changing the boot mode when writing the Boot Register (0x300018).

III.4.4 Pin Control Registers

III.4.4.1 Pull-up Control

The S1C33E08 input/output pins have a pull-up resistor that can be connected/disconnected to/from the pin by software control, except some special pins. Each pin has a pull-up control bit to select whether the pull-up resistor is used or not.

Table III.4.4.1.1 lists the correspondence between the register/control bits and pins.

Table III.4.4.1.1 Correspondence between Pull-up Control Bits and Pins

Control register	Control bit	Pin	Init.
P0 Pull-up Control Register (0x300C42)	PUP0[7:0] (D[7:0])	P07–P00	No pull-up
P1 Pull-up Control Register (0x300C43)	PUP1[7:5] (D[7:5])	P17–P15	Pull-up
	PUP1[4:0] (D[4:0])	P14–P10	No pull-up
P2 Pull-up Control Register (0x300C44)	PUP2[7:0] (D[7:0])	P27–P20	Pull-up
P3 Pull-up Control Register (0x300C45)	PUP3[6:4] (D[6:4])	P36–P34	Pull-up
	PUP3[3:0] (D[3:0])	P33–P30	No pull-up
P4 Pull-up Control Register (0x300C46)	PUP4[7:0] (D[7:0])	P47–P40	Pull-up
P5 Pull-up Control Register (0x300C47)	PUP5[7:0] (D[7:0])	P57–P50	Pull-up
P6 Pull-up Control Register (0x300C48)	PUP6[7:0] (D[7:0])	P67–P60	No pull-up
P7 Pull-up Control Register (0x300C49)	PUP7[4:0] (D[4:0])	P74–P70	No pull-up
P8 Pull-up Control Register (0x300C4A)	PUP8[5:0] (D[5:0])	P85–P80	No pull-up
P9 Pull-up Control Register (0x300C4B)	PUP9[7:0] (D[7:0])	P97–P90	No pull-up
PA Pull-up Control Register (0x300C4C)	PUPA[4:0] (D[4:0])	PA4–PA0	Pull-up
PB Pull-up Control Register (0x300C4D)	PUPB[3:0] (D[3:0])	PB3–PB0	Pull-up

When the pull-up control bit is set to 1, the corresponding pin will be pulled up in input mode. When not using pull-up resistors, set the corresponding pull-up control bits to 0.

- Notes:**
- The pull-up control bit is effective in both cases when the pin is used for the external bus and when used for the on-chip peripheral circuit or general-purpose I/O port.
 - When the port is in output mode, the port pin is not pulled up regardless of how the pull-up control bit is set.

III.4.4.2 Driving Bus Signals Low

The S1C33E08 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.4.4.2.1 lists the correspondence between the register/control bits and bus signals.

Table III.4.4.2.1 Correspondence between Low-Drive Control Bits and Bus Signals

Control register	Control bit	Bus signal
Bus Signal Low Drive Control Register (0x300C41)	LDRVDB (D3)	D[15:0]
	LDRVCE (D2)	#CE[11:4]
	LDRVAD (D1)	A[24:0]
	LDRV RW (D0)	#RD, #WRL, #WRH, #BSL

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 SRAMC/SDRAMC.

- 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.4.5 Misc Register Operating Clock

The Misc registers are clocked by the MISC_CLK clock (= MCLK) generated by the CMU.
For details on how to control the clock, see Section III.1, “Clock Management Unit (CMU).”

Controlling the supply of the Misc register operating clock

MISC_CLK is always supplied to the Misc registers and cannot be stopped in normal operation mode. However, the clock supply can be stopped in HALT mode. By setting MISC_HCKE (D24/0x301B04) to 0, MISC_CLK stops when the CPU enters HALT mode and it resumes when the CPU exits HALT mode.

* **MISC_HCKE**: Misc Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D24/0x301B04)

Clock state in standby mode

The supply of the Misc register operating clock stops depending on the type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

It can be stopped by setting MISC_HCKE (D24/0x301B04) to 0.

SLEEP mode: The clock supply stops.

Therefore, the Misc registers also stop operating in SLEEP mode.

III

MISC

III.4.6 Details of Control Registers

Table III.4.6.1 List of Misc Registers

Address	Register name	Size	Function
0x00300010	RTC Wait Control Register (pMISC_RTCWT)	8	Sets the RTC register access wait cycle.
0x00300012	USB Wait Control Register (pMISC_USBWT)	8	Sets the USB register access wait cycle.
0x00300014	Debug Port MUX Register (pMISC_PMUX)	8	Configures the P15–P17 and P34–P36 pins for debugging.
0x00300016	Performance Analyzer Control Register (pMISC_PAC)	8	Test register
0x00300018	Boot Register (pMISC_BOOT)	8	Indicates/sets boot conditions.
0x0030001A	COROM Switch Register (pMISC_COROM)	8	Test register
0x00300020	Misc Protect Register (pMISC_PROT)	8	Enables/disables write protection of Misc registers.
0x00300C41	Bus Signal Low Drive Control Register (pMISC_BUSLOW)	8	Drives the bus signals low.
0x00300C42	P0 Pull-up Control Register (pMISC_PUP0)	8	Controls the P0 port pull-up resistors.
0x00300C43	P1 Pull-up Control Register (pMISC_PUP1)	8	Controls the P1 port pull-up resistors.
0x00300C44	P2 Pull-up Control Register (pMISC_PUP2)	8	Controls the P2 port pull-up resistors.
0x00300C45	P3 Pull-up Control Register (pMISC_PUP3)	8	Controls the P3 port pull-up resistors.
0x00300C46	P4 Pull-up Control Register (pMISC_PUP4)	8	Controls the P4 port pull-up resistors.
0x00300C47	P5 Pull-up Control Register (pMISC_PUP5)	8	Controls the P5 port pull-up resistors.
0x00300C48	P6 Pull-up Control Register (pMISC_PUP6)	8	Controls the P6 port pull-up resistors.
0x00300C49	P7 Pull-up Control Register (pMISC_PUP7)	8	Controls the P7 port pull-up resistors.
0x00300C4A	P8 Pull-up Control Register (pMISC_PUP8)	8	Controls the P8 port pull-up resistors.
0x00300C4B	P9 Pull-up Control Register (pMISC_PUP9)	8	Controls the P9 port pull-up resistors.
0x00300C4C	PA Pull-up Control Register (pMISC_PUPA)	8	Controls the PA port pull-up resistors.
0x00300C4D	PB Pull-up Control Register (pMISC_PUPB)	8	Controls the PB port pull-up resistors.

The following describes the Misc registers.

The Misc registers are mapped as 8-bit devices to Area 6 at addresses 0x300010 to 0x300020 and 0x300C41 to 0x300C4D, and can be accessed in units of bytes.

Note: The Misc registers at addresses 0x300010–0x30001A are write-protected. Before the Misc registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to the Misc Protect Register (0x300020). Note that since unnecessary rewrites to addresses 0x300010–0x30001A could lead to erratic system operation, the Misc Protect Register (0x300020) should be set to other than 0x96 unless said Misc registers must be rewritten.

The registers located from 0x300C41 to 0x300C4D are not protected.

0x300010: RTC Wait Control Register (pMISC_RTCWT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC wait control register (pMISC_RTCWT) Protected	00300010 (B)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	RTCWT2	RTC register access wait control	0 to 7 (cycles)	1	R/W	
		D1	RTCWT1					
		D0	RTCWT0					

D[7:3] Reserved**D[2:0] RTCWT[2:0]: RTC Register Access Wait Control Bits**

These bits set the number of wait cycles to be inserted when accessing the RTC control register.

Table III.4.6.2 Number of Wait Cycles during RTC Access

RTCWT2	RTCWT1	RTCWT0	Number of wait cycles (in units of MCLK cycles)
1	1	1	7 cycles
1	1	0	6 cycles
1	0	1	5 cycles
1	0	0	4 cycles
0	1	1	3 cycles
0	1	0	2 cycles
0	0	1	1 cycle
0	0	0	0 cycles

(Default: 0b111 = 7 cycles)

The number of wait cycles should be set according to the MCLK clock frequency.
The S1C33E08 is able to operate with $RTCWT[2:0] \geq 1$.

0x300012: USB Wait Control Register (pMISC_USBWT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USB wait control register (pMISC_USBWT)	00300012 (B)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	USBSNZ	USB snooze control	1 Enabled 0 Disabled	0	R/W	
		D4-3	–	reserved	–	–	–	0 when being read.
		D2	USBWT2	USB register access wait control	0 to 7 (cycles)	1	R/W	
		D1	USBWT1					
D0	USBWT0							
Protected								

D[7:6] Reserved

D5 USBSNZ: USB Snooze Control Bit

This bit enables/disables the USB snooze control.

1 (R/W): Enable

0 (R/W): Disable (default)

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, see Section IX.1.4.4, “Snooze.”

D[4:3] Reserved

D[2:0] USBWT[2:0]: USB Register Access Wait Control Bits

These bits set the number of wait cycles to be inserted when accessing the USB control register.

Table III.4.6.3 Number of Wait Cycles during USB Access

USBWT2	USBWT1	USBWT0	Number of wait cycles (in units of MCLK cycles)	MCLK clock frequency
1	1	1	7 cycles	60 MHz or less
1	1	0	6 cycles	56 MHz or less
1	0	1	5 cycles	45 MHz or less
1	0	0	4 cycles	36 MHz or less
0	1	1	3 cycles	24 MHz or less
0	1	0	2 cycles	16 MHz or less
0	0	1	1 cycle	8 MHz or less
0	0	0	0 cycles	8 MHz or less

(Default: 0b111 = 7 cycles)

The number of wait cycles should be set according to the MCLK clock frequency.

0x300014: Debug Port MUX Register (pMISC_PMUX)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
Debug port MUX register (pMISC_PMUX) Protected	00300014 (B)	D7-1	–	reserved	–			–	–	0 when being read.
		D0	TRCMUX	P15–17, P34–36 debug function selection	1	Debug	0	GPIO, etc.	1	R/W

D[7:1] Reserved**D0 TRCMUX: P15–17, P34–36 Debug Function Select Bit**

This bit configures the P15–P17 and P34–P36 pins for debugging.

1 (R/W): Debug pin (default)

0 (R/W): GPIO or peripheral function pin

The P15–P17 and P34–P36 pins are shared with the PC trace debugging function and other peripheral functions. These pins are configured as the debug pins DST0 (P15), DST1 (P16), DPCO (P17), DSIO (P34), DCLK (P35), and DST2 (P36) at initial reset. When using these pins as GPIO or other peripheral function pins, TRCMUX must be set to 0.

When TRCMUX is set to 0, the port function select bits for P15–P17 and P34–P36 are enabled to select the pin function. Set TRCMUX to 1 to use these pins for debugging.

0x300016: Performance Analyzer Control Register (pMISC_PAC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Performance analyzer control register (pMISC_PAC) Protected	00300016 (B)	D7-2	-	reserved	-	-	-	Do not access in the user program.
		D1	PARUN	Test bit	-	-	-	
		D0	PACLR	Test bit	-	-	-	

D[7:2] Reserved

D1 PARUN: Test Bit

D0 PACLR: Test Bit

Note: This register is used for factory tests. Do not access from the user program.

0x300018: Boot Register (pMISC_BOOT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Boot register (pMISC_BOOT) Protected	00300018 (B)	D7	BOOT3	Boot mode indicator	BOOT[3:0]		Boot mode	X	R/W	Depend on the BOOT1 and BOOT0 pin status at initial reset
		D6	BOOT2		1000	SPI	X			
		D5	BOOT1		0100	NOR Flash/ROM	X			
		D4	BOOT0		0010	reserved	X			
					0001	NAND Flash	X			
D3-2	–	reserved	–	–	–	0 when being read.				
D1	BOOT_ENA	#CE10 area boot enable	1	Internal	0	External	1	R/W		
D0	CE10_SIZE	#CE10 area size	1	16 bits	0	8 bits	1	R/W		

D[7:4] BOOT[3:0]: Boot Mode Indicator Bits

These bits indicate the boot device that has been specified by the BOOT1 and BOOT0 pins.

Table III.4.6.4 BOOT[3:0] Bits

BOOT3	BOOT2	BOOT1	BOOT0	Boot mode
1	0	0	0	SPI boot
0	1	0	0	NOR Flash/external ROM boot
0	0	1	0	Reserved
0	0	0	1	NAND Flash boot

These bit are set in the system boot program. Do not alter these bit values from the user routine.

D[3:2] Reserved**D1 BOOT_ENA: #CE10 Area Boot Enable Bit**

This bit enables fetching the RESET vector from the #CE10 external area (0xC00000).

1 (R/W): Internal boot (default)

0 (R/W): External boot

This bit is set in the system boot program. When programming a Flash memory on the target board, BOOT_ENA must be set to 0.

D0 CE10_SIZE: #CE10 Area Size Select Bit

This bit sets the #CE10 area size in the booting process.

1 (R/W): 16 bits (default)

0 (R/W): 8 bits

This bit is set in the system boot program. Do not alter this bit value from the user routine.

0x30001A: COROM Switch Register (pMISC_COROM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
COROM switch register (pMISC_COROM) Protected	0030001A (B)	D7-1	-	reserved	-	-	-	Do not access in the user program.
		D0	COROM_SW	Test bit	-	-	-	

D[7:1] **Reserved**

D0 **COROM_SW: Test Bit**

Note: This register is used for factory tests. Do not access from the user program.

0x300020: Misc Protect Register (pMISC_PROT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Misc protect register (pMISC_PROT)	00300020 (B)	D7	PROT7	Misc register protect flag	Writing 10010110 (0x96) removes the write protection of the Misc registers (0x300010–0x30001A). Writing another value set the write protection.	0	R/W	
		D6	PROT6			0		
		D5	PROT5			0		
		D4	PROT4			0		
		D3	PROT3			0		
		D2	PROT2			0		
		D1	PROT1			0		
		D0	PROT0			0		

D[7:0] PROT[7:0]: Misc Register Protect Flag

Enables/disables write protection of the Misc registers (0x300010–0x30001A).

0x96 (R/W): Disable write protection

Other than 0x96 (R/W): Write-protect the register (default: 0x0)

Before altering any Misc register from 0x300010 to 0x30001A, write data 0x96 to this register to disable write protection. If this register is set to other than 0x96, even if an attempt is made to alter any Misc register by executing a write instruction, the content of said register will not be altered even though the instruction may have been executed without a problem. Once this register is set to 0x96, the Misc registers can be rewritten any number of times until being reset to other than 0x96. When rewriting the Misc registers has finished, this register should be set to other than 0x96 to prevent accidental writing to the Misc registers.

0x300C41: Bus Signal Low Drive Control Register (pMISC_BUSLOW)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Bus signal low drive control register (pMISC_BUSLOW)	00300C41 (B)	D7-4	–	reserved	–		–	–	0 when being read.	
		D3	LDRVDB	D15–D0 low drive	1	Low drive	0	Normal output	0	R/W
		D2	LDRVCE	#CE11–#CE4 low drive					0	R/W
		D1	LDRVAD	A24–A0 low drive					0	R/W
		D0	LDRVWRW	#RD,#WRL,#WRH,#BSL low drive					0	R/W

D[7:4] Reserved

D3 LDRVDB: D15–D0 Low Drive Control Bit

Drives the data bus signals forcibly low.

1 (R/W): Low drive

0 (R/W): Normal output (default)

When LDRVDB is set to 1, the D15–D0 signals are forcibly driven low. When it is set to 0, the signals are controlled by the SRAMC/SDRAMC normally.

D2 LDRVCE: #CE11–#CE4 Low Drive Control Bit

Drives the chip enable signals forcibly low.

1 (R/W): Low drive

0 (R/W): Normal output (default)

When LDRVCE is set to 1, the #CE11–#CE4 signals are forcibly driven low. When it is set to 0, the signals are controlled by the SRAMC/SDRAMC normally.

D1 LDRVAD: A24–A0 Low Drive Control Bit

Drives the address bus signals forcibly low.

1 (R/W): Low drive

0 (R/W): Normal output (default)

When LDRVAD is set to 1, the A24–A0 signals are forcibly driven low. When it is set to 0, the signals are controlled by the SRAMC/SDRAMC normally.

D0 LDRVWRW: #RD, #WRL, #WRH, #BSL Low Drive Control Bit

Drives the bus control signals forcibly low.

1 (R/W): Low drive

0 (R/W): Normal output (default)

When LDRVWRW is set to 1, the #RD, #WRL, #WRH, and #BSL signals are forcibly driven low. When it is set to 0, the signals are controlled by the SRAMC normally.

Notes: • When the pins are used as the general-purpose I/O port (Pxx), the low-drive control is not effective.

- If the bus 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.

0x300C42: P0 Pull-up Control Register (pMISC_PUP0)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P0 pull-up control register (pMISC_PUP0)	00300C42 (B)	D7	PUP07	P07 pull-up	1	Pulled up	0	No pull-up	0	R/W	
		D6	PUP06	P06 pull-up					0	R/W	
		D5	PUP05	P05 pull-up					0	R/W	
		D4	PUP04	P04 pull-up					0	R/W	
		D3	PUP03	P03 pull-up					0	R/W	
		D2	PUP02	P02 pull-up					0	R/W	
		D1	PUP01	P01 pull-up					0	R/W	
		D0	PUP00	P00 pull-up					0	R/W	

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D7 PUP07: P07 Pull-up Control Bit

Controls the pull-up resistor at the P07 port. (default: 0, No pull-up)

D6 PUP06: P06 Pull-up Control Bit

Controls the pull-up resistor at the P06 port. (default: 0, No pull-up)

D5 PUP05: P05 Pull-up Control Bit

Controls the pull-up resistor at the P05 port. (default: 0, No pull-up)

D4 PUP04: P04 Pull-up Control Bit

Controls the pull-up resistor at the P04 port. (default: 0, No pull-up)

D3 PUP03: P03 Pull-up Control Bit

Controls the pull-up resistor at the P03 port. (default: 0, No pull-up)

D2 PUP02: P02 Pull-up Control Bit

Controls the pull-up resistor at the P02 port. (default: 0, No pull-up)

D1 PUP01: P01 Pull-up Control Bit

Controls the pull-up resistor at the P01 port. (default: 0, No pull-up)

D0 PUP00: P00 Pull-up Control Bit

Controls the pull-up resistor at the P00 port. (default: 0, No pull-up)

0x300C43: P1 Pull-up Control Register (pMISC_PUP1)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P1 pull-up control register (pMISC_PUP1)	00300C43 (B)	D7	PUP17	P17 pull-up	1	Pulled up	0	No pull-up	1	R/W	
		D6	PUP16	P16 pull-up					1	R/W	
		D5	PUP15	P15 pull-up					1	R/W	
		D4	PUP14	P14 pull-up					0	R/W	
		D3	PUP13	P13 pull-up					0	R/W	
		D2	PUP12	P12 pull-up					0	R/W	
		D1	PUP11	P11 pull-up					0	R/W	
		D0	PUP10	P10 pull-up					0	R/W	

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D7 PUP17: P17 Pull-up Control Bit

Controls the pull-up resistor at the P17 port. (default: 1, Pulled up)

D6 PUP16: P16 Pull-up Control Bit

Controls the pull-up resistor at the P16 port. (default: 1, Pulled up)

D5 PUP15: P15 Pull-up Control Bit

Controls the pull-up resistor at the P15 port. (default: 1, Pulled up)

D4 PUP14: P14 Pull-up Control Bit

Controls the pull-up resistor at the P14 port. (default: 0, No pull-up)

D3 PUP13: P13 Pull-up Control Bit

Controls the pull-up resistor at the P13 port. (default: 0, No pull-up)

D2 PUP12: P12 Pull-up Control Bit

Controls the pull-up resistor at the P12 port. (default: 0, No pull-up)

D1 PUP11: P11 Pull-up Control Bit

Controls the pull-up resistor at the P11 port. (default: 0, No pull-up)

D0 PUP10: P10 Pull-up Control Bit

Controls the pull-up resistor at the P10 port. (default: 0, No pull-up)

0x300C44: P2 Pull-up Control Register (pMISC_PUP2)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P2 pull-up control register (pMISC_PUP2)	00300C44 (B)	D7	PUP27	P27 pull-up	1	Pulled up	0	No pull-up	1	R/W	
		D6	PUP26	P26 pull-up					1	R/W	
		D5	PUP25	P25 pull-up					1	R/W	
		D4	PUP24	P24 pull-up					1	R/W	
		D3	PUP23	P23 pull-up					1	R/W	
		D2	PUP22	P22 pull-up					1	R/W	
		D1	PUP21	P21 pull-up					1	R/W	
		D0	PUP20	P20 pull-up					1	R/W	

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D7 PUP27: P27 Pull-up Control Bit

Controls the pull-up resistor at the P27 port. (default: 1, Pulled up)

D6 PUP26: P26 Pull-up Control Bit

Controls the pull-up resistor at the P26 port. (default: 1, Pulled up)

D5 PUP25: P25 Pull-up Control Bit

Controls the pull-up resistor at the P25 port. (default: 1, Pulled up)

D4 PUP24: P24 Pull-up Control Bit

Controls the pull-up resistor at the P24 port. (default: 1, Pulled up)

D3 PUP23: P23 Pull-up Control Bit

Controls the pull-up resistor at the P23 port. (default: 1, Pulled up)

D2 PUP22: P22 Pull-up Control Bit

Controls the pull-up resistor at the P22 port. (default: 1, Pulled up)

D1 PUP21: P21 Pull-up Control Bit

Controls the pull-up resistor at the P21 port. (default: 1, Pulled up)

D0 PUP20: P20 Pull-up Control Bit

Controls the pull-up resistor at the P20 port. (default: 1, Pulled up)

0x300C45: P3 Pull-up Control Register (pMISC_PUP3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P3 pull-up control register (pMISC_PUP3)	00300C45 (B)	D7	–	reserved	–	–	–	1 when being read.	
		D6	PUP36	P36 pull-up	1 Pulled up	0 No pull-up	1	R/W	
		D5	PUP35	P35 pull-up			1	R/W	
		D4	PUP34	P34 pull-up			1	R/W	
		D3	PUP33	P33 pull-up			0	R/W	
		D2	PUP32	P32 pull-up			0	R/W	
		D1	PUP31	P31 pull-up			0	R/W	
		D0	PUP30	P30 pull-up			0	R/W	

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D7 Reserved

D6 PUP36: P36 Pull-up Control Bit

Controls the pull-up resistor at the P36 port. (default: 1, Pulled up)

D5 PUP35: P35 Pull-up Control Bit

Controls the pull-up resistor at the P35 port. (default: 1, Pulled up)

D4 PUP34: P34 Pull-up Control Bit

Controls the pull-up resistor at the P34 port. (default: 1, Pulled up)

D3 PUP33: P33 Pull-up Control Bit

Controls the pull-up resistor at the P33 port. (default: 0, No pull-up)

D2 PUP32: P32 Pull-up Control Bit

Controls the pull-up resistor at the P32 port. (default: 0, No pull-up)

D1 PUP31: P31 Pull-up Control Bit

Controls the pull-up resistor at the P31 port. (default: 0, No pull-up)

D0 PUP30: P30 Pull-up Control Bit

Controls the pull-up resistor at the P30 port. (default: 0, No pull-up)

0x300C46: P4 Pull-up Control Register (pMISC_PUP4)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P4 pull-up control register (pMISC_PUP4)	00300C46 (B)	D7	PUP47	P47 pull-up	1	Pulled up	0	No pull-up	1	R/W	
		D6	PUP46	P46 pull-up					1	R/W	
		D5	PUP45	P45 pull-up					1	R/W	
		D4	PUP44	P44 pull-up					1	R/W	
		D3	PUP43	P43 pull-up					1	R/W	
		D2	PUP42	P42 pull-up					1	R/W	
		D1	PUP41	P41 pull-up					1	R/W	
		D0	PUP40	P40 pull-up					1	R/W	

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D7 PUP47: P47 Pull-up Control Bit

Controls the pull-up resistor at the P47 port. (default: 1, Pulled up)

D6 PUP46: P46 Pull-up Control Bit

Controls the pull-up resistor at the P46 port. (default: 1, Pulled up)

D5 PUP45: P45 Pull-up Control Bit

Controls the pull-up resistor at the P45 port. (default: 1, Pulled up)

D4 PUP44: P44 Pull-up Control Bit

Controls the pull-up resistor at the P44 port. (default: 1, Pulled up)

D3 PUP43: P43 Pull-up Control Bit

Controls the pull-up resistor at the P43 port. (default: 1, Pulled up)

D2 PUP42: P42 Pull-up Control Bit

Controls the pull-up resistor at the P42 port. (default: 1, Pulled up)

D1 PUP41: P41 Pull-up Control Bit

Controls the pull-up resistor at the P41 port. (default: 1, Pulled up)

D0 PUP40: P40 Pull-up Control Bit

Controls the pull-up resistor at the P40 port. (default: 1, Pulled up)

0x300C47: P5 Pull-up Control Register (pMISC_PUP5)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
P5 pull-up control register (pMISC_PUP5)	00300C47 (B)	D7	PUP57	P57 pull-up	1	Pulled up	0	No pull-up	1	R/W
		D6	PUP56	P56 pull-up					1	R/W
		D5	PUP55	P55 pull-up					1	R/W
		D4	PUP54	P54 pull-up					1	R/W
		D3	PUP53	P53 pull-up					1	R/W
		D2	PUP52	P52 pull-up					1	R/W
		D1	PUP51	P51 pull-up					1	R/W
		D0	PUP50	P50 pull-up					1	R/W

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D7 PUP57: P57 Pull-up Control Bit

Controls the pull-up resistor at the P57 port. (default: 1, Pulled up)

D6 PUP56: P56 Pull-up Control Bit

Controls the pull-up resistor at the P56 port. (default: 1, Pulled up)

D5 PUP55: P55 Pull-up Control Bit

Controls the pull-up resistor at the P55 port. (default: 1, Pulled up)

D4 PUP54: P54 Pull-up Control Bit

Controls the pull-up resistor at the P54 port. (default: 1, Pulled up)

D3 PUP53: P53 Pull-up Control Bit

Controls the pull-up resistor at the P53 port. (default: 1, Pulled up)

D2 PUP52: P52 Pull-up Control Bit

Controls the pull-up resistor at the P52 port. (default: 1, Pulled up)

D1 PUP51: P51 Pull-up Control Bit

Controls the pull-up resistor at the P51 port. (default: 1, Pulled up)

D0 PUP50: P50 Pull-up Control Bit

Controls the pull-up resistor at the P50 port. (default: 1, Pulled up)

0x300C48: P6 Pull-up Control Register (pMISC_PUP6)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P6 pull-up control register (pMISC_PUP6)	00300C48 (B)	D7	PUP67	P67 pull-up	1	Pulled up	0	No pull-up	0	R/W	
		D6	PUP66	P66 pull-up					0	R/W	
		D5	PUP65	P65 pull-up					0	R/W	
		D4	PUP64	P64 pull-up					0	R/W	
		D3	PUP63	P63 pull-up					0	R/W	
		D2	PUP62	P62 pull-up					0	R/W	
		D1	PUP61	P61 pull-up					0	R/W	
		D0	PUP60	P60 pull-up					0	R/W	

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D7 PUP67: P67 Pull-up Control Bit

Controls the pull-up resistor at the P67 port. (default: 0, No pull-up)

D6 PUP66: P66 Pull-up Control Bit

Controls the pull-up resistor at the P66 port. (default: 0, No pull-up)

D5 PUP65: P65 Pull-up Control Bit

Controls the pull-up resistor at the P65 port. (default: 0, No pull-up)

D4 PUP64: P64 Pull-up Control Bit

Controls the pull-up resistor at the P64 port. (default: 0, No pull-up)

D3 PUP63: P63 Pull-up Control Bit

Controls the pull-up resistor at the P63 port. (default: 0, No pull-up)

D2 PUP62: P62 Pull-up Control Bit

Controls the pull-up resistor at the P62 port. (default: 0, No pull-up)

D1 PUP61: P61 Pull-up Control Bit

Controls the pull-up resistor at the P61 port. (default: 0, No pull-up)

D0 PUP60: P60 Pull-up Control Bit

Controls the pull-up resistor at the P60 port. (default: 0, No pull-up)

0x300C49: P7 Pull-up Control Register (pMISC_PUP7)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
P7 pull-up control register (pMISC_PUP7)	00300C49 (B)	D7-5	–	reserved	–		–	–	1 when being read.	
		D4	PUP74	P74 pull-up	1	Pulled up	0	No pull-up	0	R/W
		D3	PUP73	P73 pull-up					0	R/W
		D2	PUP72	P72 pull-up					0	R/W
		D1	PUP71	P71 pull-up					0	R/W
		D0	PUP70	P70 pull-up					0	R/W

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D[7:5] Reserved

D4 **PUP74: P74 Pull-up Control Bit**

Controls the pull-up resistor at the P74 port. (default: 0, No pull-up)

D3 **PUP73: P73 Pull-up Control Bit**

Controls the pull-up resistor at the P73 port. (default: 0, No pull-up)

D2 **PUP72: P72 Pull-up Control Bit**

Controls the pull-up resistor at the P72 port. (default: 0, No pull-up)

D1 **PUP71: P71 Pull-up Control Bit**

Controls the pull-up resistor at the P71 port. (default: 0, No pull-up)

D0 **PUP70: P70 Pull-up Control Bit**

Controls the pull-up resistor at the P70 port. (default: 0, No pull-up)

0x300C4A: P8 Pull-up Control Register (pMISC_PUP8)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
P8 pull-up control register (pMISC_PUP8)	00300C4A (B)	D7-6	–	reserved	–		–	–	1 when being read.	
		D5	PUP85	P85 pull-up	1	Pulled up	0	No pull-up	0	R/W
		D4	PUP84	P84 pull-up					0	R/W
		D3	PUP83	P83 pull-up					0	R/W
		D2	PUP82	P82 pull-up					0	R/W
		D1	PUP81	P81 pull-up					0	R/W
		D0	PUP80	P80 pull-up					0	R/W

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D[7:6] Reserved**D5 PUP85: P85 Pull-up Control Bit**

Controls the pull-up resistor at the P85 port. (default: 0, No pull-up)

D4 PUP84: P84 Pull-up Control Bit

Controls the pull-up resistor at the P84 port. (default: 0, No pull-up)

D3 PUP83: P83 Pull-up Control Bit

Controls the pull-up resistor at the P83 port. (default: 0, No pull-up)

D2 PUP82: P82 Pull-up Control Bit

Controls the pull-up resistor at the P82 port. (default: 0, No pull-up)

D1 PUP81: P81 Pull-up Control Bit

Controls the pull-up resistor at the P81 port. (default: 0, No pull-up)

D0 PUP80: P80 Pull-up Control Bit

Controls the pull-up resistor at the P80 port. (default: 0, No pull-up)

0x300C4B: P9 Pull-up Control Register (pMISC_PUP9)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P9 pull-up control register (pMISC_PUP9)	00300C4B (B)	D7	PUP97	P97 pull-up	1	Pulled up	0	No pull-up	0	R/W	
		D6	PUP96	P96 pull-up					0	R/W	
		D5	PUP95	P95 pull-up					0	R/W	
		D4	PUP94	P94 pull-up					0	R/W	
		D3	PUP93	P93 pull-up					0	R/W	
		D2	PUP92	P92 pull-up					0	R/W	
		D1	PUP91	P91 pull-up					0	R/W	
		D0	PUP90	P90 pull-up					0	R/W	

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

- D7 PUP97: P97 Pull-up Control Bit**
Controls the pull-up resistor at the P97 port. (default: 0, No pull-up)
- D6 PUP96: P96 Pull-up Control Bit**
Controls the pull-up resistor at the P96 port. (default: 0, No pull-up)
- D5 PUP95: P95 Pull-up Control Bit**
Controls the pull-up resistor at the P95 port. (default: 0, No pull-up)
- D4 PUP94: P94 Pull-up Control Bit**
Controls the pull-up resistor at the P94 port. (default: 0, No pull-up)
- D3 PUP93: P93 Pull-up Control Bit**
Controls the pull-up resistor at the P93 port. (default: 0, No pull-up)
- D2 PUP92: P92 Pull-up Control Bit**
Controls the pull-up resistor at the P92 port. (default: 0, No pull-up)
- D1 PUP91: P91 Pull-up Control Bit**
Controls the pull-up resistor at the P91 port. (default: 0, No pull-up)
- D0 PUP90: P90 Pull-up Control Bit**
Controls the pull-up resistor at the P90 port. (default: 0, No pull-up)

0x300C4C: PA Pull-up Control Register (pMISC_PUPA)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
PA pull-up control register (pMISC_PUPA)	00300C4C (B)	D7-5	–	reserved	–		–	–	1 when being read.	
		D4	PUPA4	PA4 pull-up	1	Pulled up	0	No pull-up	1	R/W
		D3	PUPA3	PA3 pull-up					1	R/W
		D2	PUPA2	PA2 pull-up					1	R/W
		D1	PUPA1	PA1 pull-up					1	R/W
		D0	PUPA0	PA0 pull-up					1	R/W

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D[7:5] Reserved

D4 PUPA4: PA4 Pull-up Control Bit

Controls the pull-up resistor at the PA4 port. (default: 1, Pulled up)

D3 PUPA3: PA3 Pull-up Control Bit

Controls the pull-up resistor at the PA3 port. (default: 1, Pulled up)

D2 PUPA2: PA2 Pull-up Control Bit

Controls the pull-up resistor at the PA2 port. (default: 1, Pulled up)

D1 PUPA1: PA1 Pull-up Control Bit

Controls the pull-up resistor at the PA1 port. (default: 1, Pulled up)

D0 PUPA0: PA0 Pull-up Control Bit

Controls the pull-up resistor at the PA0 port. (default: 1, Pulled up)

0x300C4D: PB Pull-up Control Register (pMISC_PUPB)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
PB pull-up control register (pMISC_PUPB)	00300C4D (B)	D7-4	–	reserved	–		–	–	1 when being read.	
		D3	PUPB3	PB3 pull-up	1	Pulled up	0	No pull-up	1	R/W
		D2	PUPB2	PB2 pull-up					1	R/W
		D1	PUPB1	PB1 pull-up					1	R/W
		D0	PUPB0	PB0 pull-up					1	R/W

This register controls the pull-up resistors for the I/O port pins.

1 (R/W): Pulled up

0 (R/W): No pull-up

When the pull-up control bit is set to 1, the corresponding pins are pulled up to high during input mode. When it is set to 0, the pins are not pulled up. When the port is in output mode, the port pin is not pulled up even if the pull-up control bit is set to 1.

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.

D[7:4] Reserved

D3 **PUPB3: PB3 Pull-up Control Bit**

Controls the pull-up resistor at the PB3 port. (default: 1, Pulled up)

D2 **PUPB2: PB2 Pull-up Control Bit**

Controls the pull-up resistor at the PB2 port. (default: 1, Pulled up)

D1 **PUPB1: PB1 Pull-up Control Bit**

Controls the pull-up resistor at the PB1 port. (default: 1, Pulled up)

D0 **PUPB0: PB0 Pull-up Control Bit**

Controls the pull-up resistor at the PB0 port. (default: 1, Pulled up)

III.4.7 Precautions

- The Misc registers at addresses 0x300010–0x30001A are write-protected. Before the Misc registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to the Misc Protect Register (0x300020). Note that since unnecessary rewrites to addresses 0x300010–0x30001A could lead to erratic system operation, the Misc Protect Register (0x300020) should be set to other than 0x96 unless said Misc registers must be rewritten.
- The low-drive control bit is disabled when the pin is used as the general-purpose I/O port (Pxx).
- If the bus 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.

THIS PAGE IS BLANK.

S1C33E08 Technical Manual

IV PERIPHERAL MODULES 2 (TIMERS)

IV.1 16-Bit Timers (T16)

IV.1.1 Configuration of 16-bit Timer

The S1C33E08 contains 6 systems of 16-bit programmable timers (timers 0 to 5). The following lists the main functions of the 16-bit timers.

- Programmable count clocks using the prescaler embedded in each 16-bit timer
- Event counter function using external clocks
- Interrupt generation function with programmable interrupt cycles using the compare data registers
- PWM output using the compare data registers
- Supports fine mode and DA16 mode suitable for high-quality audio output using PWM

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 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 IV.1.1.1 shows the structure of one channel of the 16-bit timer.

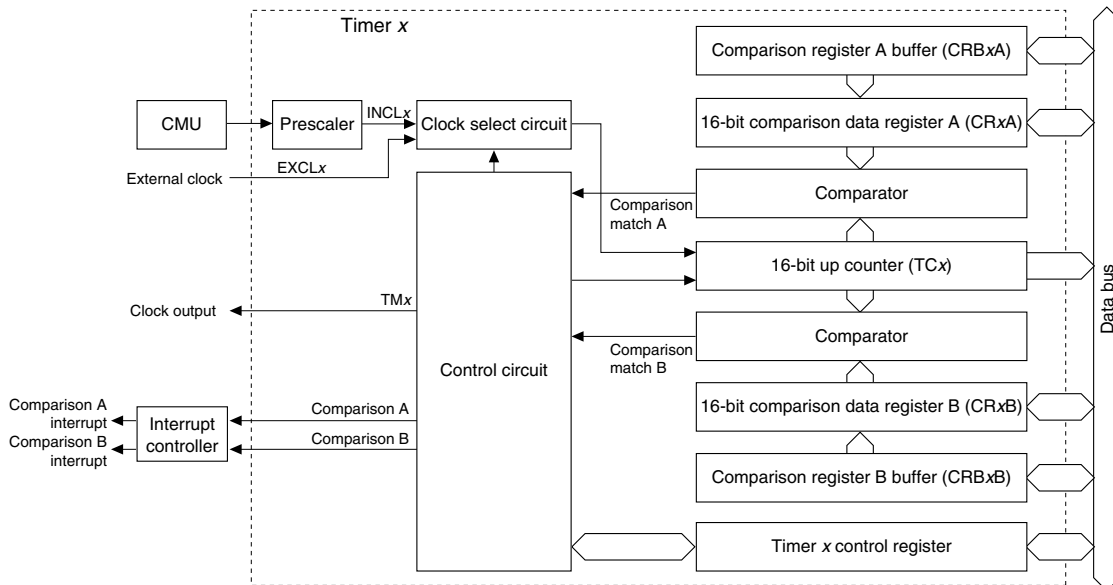


Figure IV.1.1.1 Structure of 16-bit Timer

In each timer, a 16-bit up-counter ($TC_x[15:0]$ ($D[15:0]/0x300784 + 8 \cdot x$)), as well as two 16-bit comparison data registers ($CR_xA[15:0]$ ($D[15:0]/0x300780 + 8 \cdot x$), $CR_xB[15:0]$ ($D[15:0]/0x300782 + 8 \cdot x$)) and their buffers (CRB_xA , CRB_xB), are provided.

- * **TC $_x$ [15:0]**: 16-bit Timer x Counter Data Bits in the 16-bit Timer x Counter Data Register ($D[15:0]/0x300784 + 8 \cdot x$)
- * **CR $_xA$ [15:0]**: 16-bit Timer x Comparison Data A Bits in the 16-bit Timer x Comparison Data A Setup Register ($D[15:0]/0x300780 + 8 \cdot x$)
- * **CR $_xB$ [15:0]**: 16-bit Timer x Comparison Data B Bits in the 16-bit Timer x Comparison Data B Setup Register ($D[15:0]/0x300782 + 8 \cdot x$)

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.

IV PERIPHERAL MODULES 2 (TIMERS): 16-BIT TIMERS (T16)

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 PRESET_x (D1/0x300786 + 8•x)). The software can select whether comparison data is written to the comparison data register or the buffer.

* **PRESET_x**: 16-bit Timer x Reset Bit in the 16-bit Timer x Control Register (D1/0x300786 + 8•x)

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.

IV.1.2 I/O Pins of 16-bit Timers

Table IV.1.2.1 shows the input/output pins used for the 16-bit timers.

Table IV.1.2.1 I/O Pins of 16-bit Timer

Pin name	I/O	Function
EXCL0	I/O	16-bit timer 0 event counter input
EXCL1	I/O	16-bit timer 1 event counter input
EXCL2	I/O	16-bit timer 2 event counter input
EXCL3	I/O	16-bit timer 3 event counter input
EXCL4	I/O	16-bit timer 4 event counter input
EXCL5	I/O	16-bit timer 5 event counter input
TM0	I/O	16-bit timer 0 output
TM1	I/O	16-bit timer 1 output
TM2	I/O	16-bit timer 2 output
TM3	I/O	16-bit timer 3 output
TM4	I/O	16-bit timer 4 output
TM5	I/O	16-bit timer 5 output

TM x (output pin of the 16-bit timer)

This pin outputs a clock generated by the timer x .

EXCL x (event counter input pin)

When using the timer x as an event counter, input count pulses from an external source to this pin.

Note: The 16-bit timer input/output pins are shared with general-purpose I/O ports or other peripheral circuit inputs/outputs, so that functionality in the initial state is set to other than the 16-bit timer input/output. Before the 16-bit timer input/output signals assigned to these pins can be used, the function of these pins must be switched for the 16-bit timer input/output by setting the corresponding Port Function Select Registers.

For details of pin functions and how to switch over, see Section I.3.3, "Switching Over the Multiplexed Pin Functions."

IV.1.3 Uses of 16-bit Timers

The up-counters of the 16-bit 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 (timers 0 to 5)

Each timer's comparison match (matching of counter and comparison data) can be used as a cause of interrupt 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 cause of interrupt can also be used to invoke IDMA or HSDMA.

Clock output to external devices (timers 0 to 5)

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 pin of each timer is described in the preceding section.

A/D converter start trigger (timer 0)

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 timer 0. This makes it possible to perform the A/D conversion at programmable intervals.

To use this function, write 0b01 to the A/D converter control bit TS[1:0] (D[4:3]/0x300542) to select the 16-bit timer 0 as the trigger.

* **TS[1:0]**: A/D Conversion Trigger Select Bits in the A/D Trigger/Channel Select Register (D[4:3]/0x300542)

IV.1.4 16-bit Timer Operating Clock

The 16-bit timers use the TM_x_CLK clock (= MCLK) generated by the CMU as the operating clock. The count clock is generated from the TM_x_CLK by the prescaler embedded in each timer.

Controlling the supply of the operating clock

The TM_x_CLK clock is supplied to the 16-bit timers with default settings. It can be turned off using TM_x_CKE (D13+x/0x301B04) to reduce the amount of power consumed on the chip if 16-bit timers are not used.

* **TM_x_CKE**: 16-bit Timer *x* Clock Control Bit in the Gated Clock Control Register 1 (D13+x/0x301B04)

Setting TM_x_CKE (D13+x/0x301B04) to 0 (1 by default) turns off the TM_x_CLK clock supply to the 16-bit timer *x*. When the clock supply is turned off, the 16-bit timer control registers cannot be accessed and the count operation stops.

For details on how to set and control the operating clock, refer to Section III.1, “Clock Management Unit (CMU).”

Clock state in standby mode

The clock supply to the 16-bit timer stops depending on type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

SLEEP mode: The operating clock supply stops.

Therefore, the 16-bit timer also stops operating in SLEEP mode.

IV.1.5 Control and Operation of 16-bit Timer

The following settings must first be made before the 16-bit timer starts counting:

1. Setting pins for input/output (only when necessary) ... See Sections IV.1.2 and I.3.3.
2. Setting count clock
3. Selecting comparison data register/buffer
4. Setting clock output conditions (signal active level, initial signal level, fine mode) ... See Section IV.1.6.
5. Setting comparison data
6. Setting interrupt/DMA ... See Section IV.1.7.

Standard mode and advanced mode

The 16-bit timer in the S1C33E08 is extended from that of the C33 STD models. This 16-bit timer has two operating modes, standard (STD) mode of which functions are compatible with the existing C33 STD models and an advanced (ADV) mode allowing use of the extended functions. Table IV.1.5.1 shows differences between standard mode and advanced mode.

Table IV.1.5.1 Differences between Standard Mode and Advanced Mode

Function	Standard mode	Advanced mode
Writing to the count data register	Disabled (read only)	Enabled
Setting of the initial timer output level (high or low)	Disabled (depending on the OUTINV x set value)	Enabled (can be specified using INITOL x)
DA16 function (DA16 registers)	Cannot be used	Can be used
Multiple timer full-sync function	Not supported	Supported (can be controlled using PAUSE x)

To configure the 16-bit timer in advanced mode, set T16ADV (D0/0x3007DE) to 1. The control registers/bits for the extended functions are enabled to write after this setting. At initial reset, T16ADV (D0/0x3007DE) is set to 0 and the 16-bit timer enters standard mode.

* **T16ADV**: Standard Mode/Advanced Mode Select Bit in the 16-bit Timer STD/ADV Mode Select Register (D0/0x3007DE)

The following descriptions unless otherwise specified are common contents for both modes. The extended functions in advanced mode are explained assuming that T16ADV (D0/0x3007DE) has been set to 1.

Note: Standard or advanced mode currently set is applied to all the 16-bit timers. It cannot be selected for each timer individually.

Setting the count clock

The count clock for each timer can be selected from between an internal clock and an external clock. Select the input clock using CKSL x (D3/0x300786 + 8• x).

* **CKSL x** : 16-bit Timer x Input Clock Select Bit in the 16-bit Timer x Control Register (D3/0x300786 + 8• x)

An external clock is selected by writing 1 to CKSL x (D3/0x300786 + 8• x), and the internal clock is selected by writing 0. At initial reset, CKSL x (D3/0x300786 + 8• x) 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 count clock is generated from the TM x _CLK (= MCLK) by the prescaler embedded in each timer.

The prescaler's division ratio can be selected from among eight ratios using P16TS x [2:0] (D[2:0]/0x3007E0 + 2• x).

The divided clock is output from the prescaler by writing 1 to P16TON x (D3/0x3007E0 + 2• x).

* **P16TS x [2:0]**: 16-bit Timer x Clock Division Ratio Select Bits in the 16-bit Timer x Clock Control Register (D[2:0]/0x3007E0 + 2• x)

* **P16TON x** : 16-bit Timer x Clock Control Bit in the 16-bit Timer x Clock Control Register (D3/0x3007E0 + 2• x)

Table IV.1.5.2 Division Ratio

P16TSx2	P16TSx1	P16TSx0	Division ratio
1	1	1	MCLK/4096
1	1	0	MCLK/1024
1	0	1	MCLK/256
1	0	0	MCLK/64
0	1	1	MCLK/16
0	1	0	MCLK/4
0	0	1	MCLK/2
0	0	0	MCLK/1

(Default: 0b000 = MCLK/1)

- Notes:**
- When setting a count clock, make sure the 16-bit timer is turned off.
 - P16TONx (D3/0x3007E0 + 2•x) for unused timers should be set to 0 to reduce current consumption.

- **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 two CPU operating clock cycles.

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 (D1/0x300786 + 8•x)).

Select whether comparison data is written to the comparison data register or the buffer using SELCRBx (D5/0x300786 + 8•x).

- * **SELCRBx:** 16-bit Timer x Comparison Register Buffer Enable Bit in the 16-bit Timer x Control Register (D5/0x300786 + 8•x)

When 1 is written to SELCRBx (D5/0x300786 + 8•x), 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 timer contains two data comparators that allows the count data to be compared with given values. CRxA[15:0] (D[15:0]/0x300780 + 8•x) and CRxB[15:0] (D[15:0]/0x300782 + 8•x) are used to set these values.

- * **CRxA[15:0]:** 16-bit Timer x Comparison Data A Bits in the 16-bit Timer x Comparison Data A Setup Register (D[15:0]/0x300780 + 8•x)
- * **CRxB[15:0]:** 16-bit Timer x Comparison Data B Bits in the 16-bit Timer x Comparison Data B Setup Register (D[15:0]/0x300782 + 8•x)

When SELCRBx (D5/0x300786 + 8•x) 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 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.

DA16 function (advanced mode)

Advanced mode supports the DA16 function that divides a 16-bit data into 10 high-order bits and 6 low-order bits and writes them to the comparison data A registers (or buffers) of a two timer pair simultaneously. This makes it possible to reduce software load for using two 16-bit timers as a 16-bit D/A converter.

Three DA16 registers are provided for this function. The following shows the correspondence between these registers and timers:

(timer A and timer B)

DA16 Ch.0 Register (0x3007D0): timer 1 and timer 2

DA16 Ch.1 Register (0x3007D2): timer 3 and timer 4

DA16 Ch.2 Register (0x3007D4): timer 5 and timer 0

When data is written to this register, 10 high-order bits are loaded into the Timer A Comparison Data A Setup Register (buffer) as 10 low-order compare data bits and 6 low-order bits are loaded into the Timer B Comparison Data A Setup Register (buffer) as 6 low-order compare data bits.

In standard mode, data cannot be written to the DA16 registers.

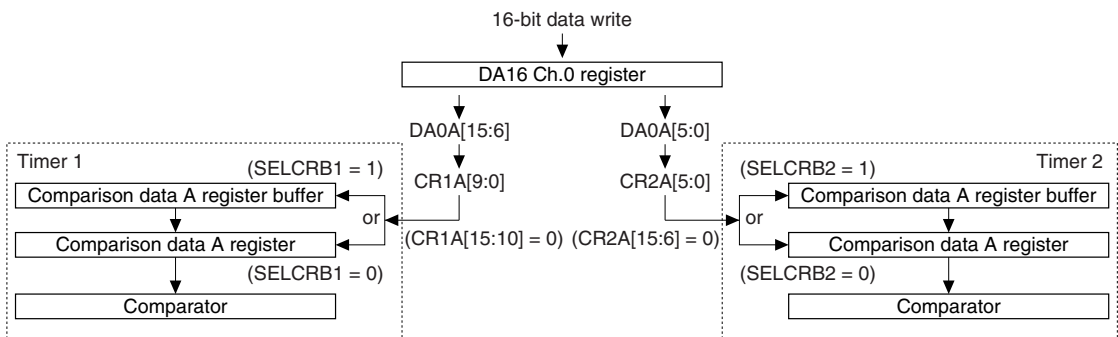


Figure IV.1.5.1 DA16 Function (Ch.0)

Resetting the counter

Each timer includes $PRESET_x$ ($D1/0x300786 + 8 \cdot x$) to reset the counter.

* **PRESET_x**: 16-bit Timer *x* Reset Bit in the 16-bit Timer *x* Control Register ($D1/0x300786 + 8 \cdot x$)

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 $PRUN_x$ ($D0/0x300786 + 8 \cdot x$) to control RUN/STOP.

* **PRUN_x**: 16-bit Timer *x* Run/Stop Control Bit in the 16-bit Timer *x* Control Register ($D0/0x300786 + 8 \cdot x$)

The timer starts counting when 1 is written to $PRUN_x$ ($D0/0x300786 + 8 \cdot x$). The clock input is disabled and the timer stops counting when 0 is written to $PRUN_x$.

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 $SELCRB_x$ ($D5/0x300786 + 8 \cdot x$) 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 $PRUN_x$ ($D0/0x300786 + 8 \cdot x$) and $PRESET_x$ ($D1/0x300786 + 8 \cdot x$) are set to 1 at the same time, the timer starts counting after resetting the counter.

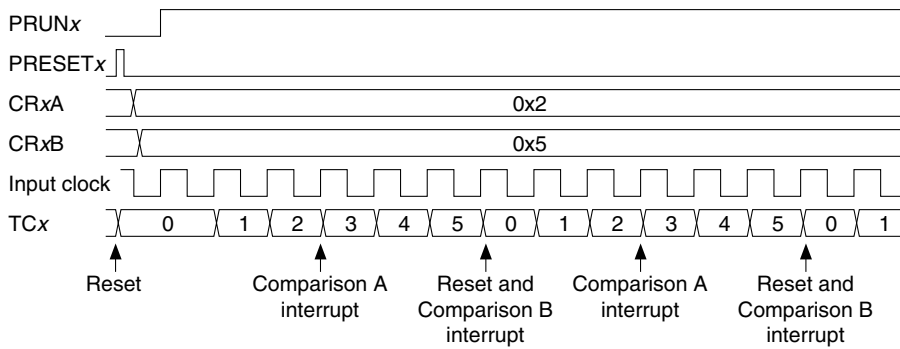


Figure IV.1.5.2 Basic Operation Timing of Counter

To synchronize multiple timers (advanced mode)

Since the timer RUN/STOP control bits are located in different addresses, two or more timers cannot be started at the same time. To synchronize multiple timers, the control bits $PAUSEx$ ($Dx/0x3007DC$) that stop each timer are provided in an address.

* **PAUSEx**: 16-bit Timer x Count Pause Bit in the Count Pause Register ($Dx/0x3007DC$)

When $PAUSEx$ ($Dx/0x3007DC$) is set to 1, timer x is placed in pause state and when set to 0, timer x starts counting or continues stop state according to the set value of $PRUNx$ ($D0/0x300786 + 8 \cdot x$).

However, it is necessary to set the 16-bit timer in advanced mode for using $PAUSEx$ ($Dx/0x3007DC$).

The following shows a procedure to synchronize multiple timers.

1. Set the prescaler clocks for the timers to be synchronized to the same condition.
2. Set $PAUSEx$ ($Dx/0x3007DC$) for the timers to 1 to place the timers in pause state.
3. Set $PRUNx$ ($D0/0x300786 + 8 \cdot x$) for the timers to 1.
The timers do not start counting at this time as $PAUSEx$ ($Dx/0x3007DC$) for the timers have been set to 1.
4. Set all the $PAUSEx$ ($Dx/0x3007DC$) bits for the timers to 0 at the same time.
The corresponding timers start counting simultaneously.

Reading counter data

The counter data can be read out from $TCx[15:0]$ ($D[15:0]/0x300784 + 8 \cdot x$) at any time.

* **TCx[15:0]**: 16-bit Timer x Counter Data Bits in the 16-bit Timer x Counter Data Register ($D[15:0]/0x300784 + 8 \cdot x$)

Writing counter data (advanced mode)

In advanced mode, counter data can be written to $TCx[15:0]$ ($D[15:0]/0x300784 + 8 \cdot x$) at any time. This makes it possible to change the interrupt and/or clock output cycles temporarily.

Standard mode does not allow writing of counter data.

IV.1.6 Controlling Clock Output

The timers can generate a TM_x signal using the comparison match signals from the counter. Figure IV.1.6.1 shows the 16-bit timer clock output circuit.

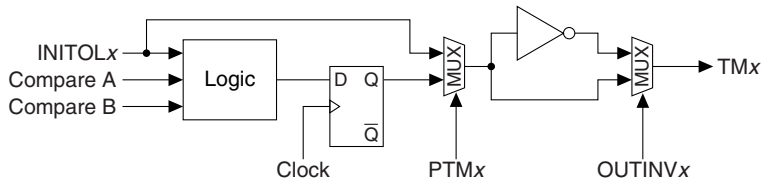


Figure IV.1.6.1 16-bit Timer Clock Output Circuit

Setting the initial output level (advanced mode)

The default output level while the clock output is turned off is 0 (low level). This level can be changed to 1 (high level) using INITOL_x (D8/0x300786 + 8•x). However, this function is available only in advanced mode.

* **INITOL_x**: 16-bit Timer x Initial Output Level Select Bit in the 16-bit Timer x Control Register (D8/0x300786 + 8•x)

When INITOL_x (D8/0x300786 + 8•x) is 0 (default), the initial output level is low. When INITOL_x (D8/0x300786 + 8•x) is set to 1, the initial output level is set to high.

The timer output goes to the initial output level when the timer is reset by writing 1 to PRESET_x (D1/0x300786 + 8•x) as well as when the timer output is turned off.

Setting the signal active level

By default, an active high signal (normal low) is generated. This logic can be inverted using OUTINV_x (D4/0x300786 + 8•x). When 1 is written to OUTINV_x, the timer generates an active low (normal high) signal.

* **OUTINV_x**: 16-bit Timer x Clock Output Inversion Bit in the 16-bit Timer x Control Register (D4/0x300786 + 8•x)

Note that the initial output level set by INITOL_x (D8/0x300786 + 8•x) is inverted when OUTINV_x (D4/0x300786 + 8•x) is set to 1.

See Figure IV.1.6.2 for the waveforms.

Setting the output port

The TM_x signal generated here can be output from the clock output pins (see Table IV.1.2.1), 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 outputs the level according to the set values of INITOL_x (D8/0x300786 + 8•x) and OUTINV_x (D4/0x300786 + 8•x). The output pin holds this level until the output level changes due to the counter value after the timer output is enabled.

Table IV.1.6.1 Initial Output Level

INITOL _x	OUTINV _x	Initial output level
1	1	Low
1	0	High
0	1	High
0	0	Low

Starting clock output

To output the TMx clock, write 1 to the clock output control bit PTMx (D2/0x300786 + 8•x). Clock output is stopped by writing 0 to PTMx and goes to the initial output level according to the set values of INITOLx (D8/0x300786 + 8•x) and OUTINVx (D4/0x300786 + 8•x).

* **PTMx**: 16-bit Timer x Clock Output Control Bit in the 16-bit Timer x Control Register (D2/0x300786 + 8•x)

Figure IV.1.6.2 shows the waveform of the output signal.

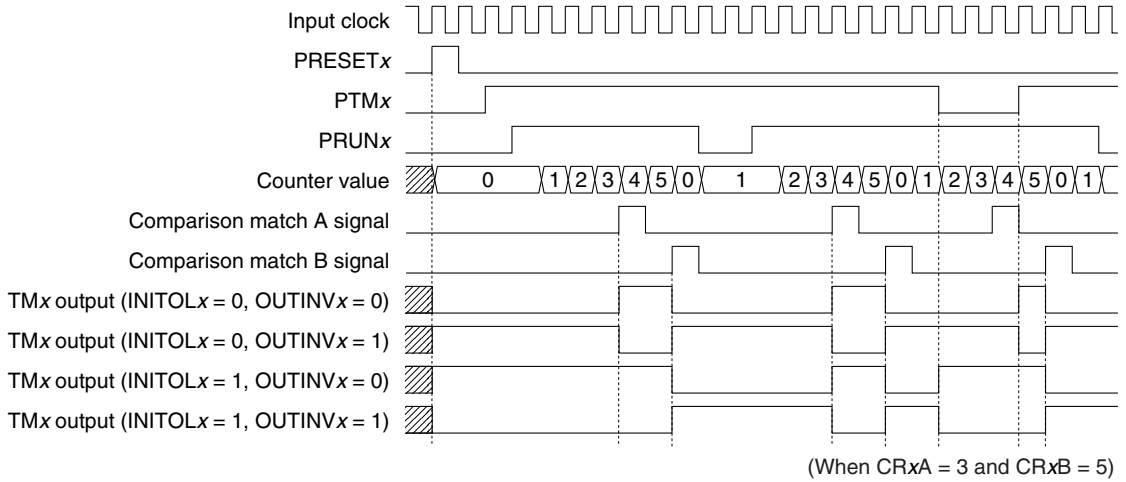


Figure IV.1.6.2 Waveform of 16-bit Timer Output

When OUTINVx (D4/0x300786 + 8•x) = 0 (active high):

The timer outputs a low level (initial output level when output is started) until the counter becomes equal to the comparison data A set in CRxA[15:0] (D[15:0]/0x300780 + 8•x). 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 CRxB[15:0] (D[15:0]/0x300782 + 8•x), the counter is reset and the output pin goes low. At the same time a comparison B interrupt occurs.

* **CRxA[15:0]**: 16-bit Timer x Comparison Data A Bits in the 16-bit Timer x Comparison Data A Setup Register (D[15:0]/0x300780 + 8•x)

* **CRxB[15:0]**: 16-bit Timer x Comparison Data B Bits in the 16-bit Timer x Comparison Data B Setup Register (D[15:0]/0x300782 + 8•x)

When OUTINVx (D4/0x300786 + 8•x) = 1 (active low):

The timer outputs a high level (inverted initial output level when output is started) until the counter becomes equal to the comparison data A set in CRxA[15:0] (D[15:0]/0x300780 + 8•x). 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 CRxB[15:0] (D[15:0]/0x300782 + 8•x), 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] (D[15:0]/0x300780 + 8•x) becomes equal to TCx[15:0] (D[15:0]/0x300784 + 8•x).

* **TCx[15:0]**: 16-bit Timer x Counter Data Bits in the 16-bit Timer x Counter Data Register (D[15:0]/0x300784 + 8•x)

In fine mode, the output signal changes according to CRxA0 (D0/0x300780 + 8•x) when CRxA[15:1] (D[15:1]/0x300780 + 8•x) becomes equal to TCx[14:0] (D[14:0]/0x300784 + 8•x).

When CRxA0 is 0, the output signal changes at the rising edge of the input clock.

When CRxA0 is 1, the output signal changes at the falling edge of the input clock a half cycle from the default setting.

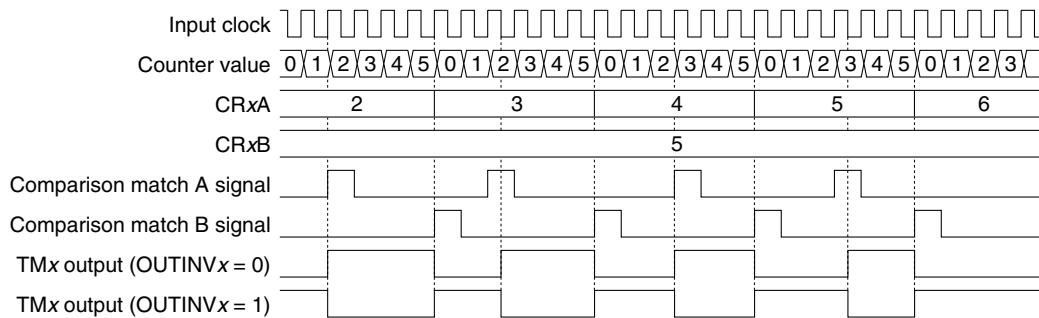


Figure IV.1.6.3 Clock Output in Fine Mode

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 CRxA[15:0] 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[15:0] is $2^{15} - 1 = 32,767$ and the range of CRxA[15:0] that can be set is 0 to $(2 \times CRxB[15:0] - 1)$.

The fine mode is set using SELFMx (D6/0x300786 + 8•x).

* **SELFMx**: 16-bit Timer x Fine Mode Select Bit in the 16-bit Timer x Control Register (D6/0x300786 + 8•x)

When 1 is written to SELFMx (D6/0x300786 + 8•x), fine mode is set. At initial reset, the fine mode is disabled.

Precautions

- (1) If a same value is set to the comparison data A and B registers, a hazard may be generated in the output signal. Therefore, do not set the comparison registers as A = B.
There is no problem when the interrupt function only is used.
- (2) When using the output clock, set the comparison data registers as $A \geq 0$ and $B \geq 1$. The minimum settings are A = 0 and B = 1. In this case, the timer output clock cycle is the input clock $\times 1/2$.
- (3) When the comparison data registers are set as $A > B$, no comparison A signal is generated. In this case, the output signal is fixed at the off level.

IV.1.7 16-bit Timer Interrupts and DMA

The 16-bit 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 IV.1.5.2 in the preceding section.

Control registers of the interrupt controller

Table IV.1.7.1 shows the control registers of the interrupt controller provided for each timer.

Table IV.1.7.1 Control Registers of Interrupt Controller

Cause of interrupt	Cause-of-interrupt flag	Interrupt enable register	Interrupt priority register
Timer 0 comparison A	F16TC0 (D3/0x300282)	E16TC0 (D3/0x300272)	P16T0[2:0] (D[2:0]/0x300266)
Timer 0 comparison B	F16TU0 (D2/0x300282)	E16TU0 (D2/0x300272)	
Timer 1 comparison A	F16TC1 (D7/0x300282)	E16TC1 (D7/0x300272)	P16T1[2:0] (D[6:4]/0x300266)
Timer 1 comparison B	F16TU1 (D6/0x300282)	E16TU1 (D6/0x300272)	
Timer 2 comparison A	F16TC2 (D3/0x300283)	E16TC2 (D3/0x300273)	P16T2[2:0] (D[2:0]/0x300267)
Timer 2 comparison B	F16TU2 (D2/0x300283)	E16TU2 (D2/0x300273)	
Timer 3 comparison A	F16TC3 (D7/0x300283)	E16TC3 (D7/0x300273)	P16T3[2:0] (D[6:4]/0x300267)
Timer 3 comparison B	F16TU3 (D6/0x300283)	E16TU3 (D6/0x300273)	
Timer 4 comparison A	F16TC4 (D3/0x300284)	E16TC4 (D3/0x300274)	P16T4[2:0] (D[2:0]/0x300268)
Timer 4 comparison B	F16TU4 (D2/0x300284)	E16TU4 (D2/0x300274)	
Timer 5 comparison A	F16TC5 (D7/0x300284)	E16TC5 (D7/0x300274)	P16T5[2:0] (D[6:4]/0x300268)
Timer 5 comparison B	F16TU5 (D6/0x300284)	E16TU5 (D6/0x300274)	

When a comparison match state occurs in the timer, the corresponding cause-of-interrupt flag is set to 1.

If the interrupt enable register bit corresponding to that cause-of-interrupt 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 cause-of-interrupt 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 Section III.2, "Interrupt Controller (ITC)."

Intelligent DMA

The cause of interrupt 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 cause of interrupt 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 IV.1.7.2 must be set to 1 in advance. Transfer conditions, etc. must also be set on the IDMA side in advance.

Table IV.1.7.2 Control Bits for IDMA Transfer

Cause of interrupt	IDMA request bit	IDMA enable bit
Timer 0 comparison A	R16TC0(D7/0x300290)	DE16TC0(D7/0x300294)
Timer 0 comparison B	R16TU0(D6/0x300290)	DE16TU0(D6/0x300294)
Timer 1 comparison A	R16TC1(D1/0x300291)	DE16TC1(D1/0x300295)
Timer 1 comparison B	R16TU1(D0/0x300291)	DE16TU1(D0/0x300295)
Timer 2 comparison A	R16TC2(D3/0x300291)	DE16TC2(D3/0x300295)
Timer 2 comparison B	R16TU2(D2/0x300291)	DE16TU2(D2/0x300295)
Timer 3 comparison A	R16TC3(D5/0x300291)	DE16TC3(D5/0x300295)
Timer 3 comparison B	R16TU3(D4/0x300291)	DE16TU3(D4/0x300295)
Timer 4 comparison A	R16TC4(D7/0x300291)	DE16TC4(D7/0x300295)
Timer 4 comparison B	R16TU4(D6/0x300291)	DE16TU4(D6/0x300295)
Timer 5 comparison A	R16TC5(D1/0x300292)	DE16TC5(D1/0x300296)
Timer 5 comparison B	R16TU5(D0/0x300292)	DE16TU5(D0/0x300296)

If the IDMA request and enable bits are set to 1, IDMA is invoked through generation of a cause of interrupt. 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 Section II.2, “Intelligent DMA (IDMA).”

High-speed DMA

The timer 0–5 interrupt causes can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit corresponding to the channel 0 to 5 timers:

Table IV.1.7.3 HSDMA Trigger Set-up Bits

Cause of interrupt	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 (0x300298) = 0111
Timer 0 comparison B	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0–1 trigger set-up register (0x300298) = 0110
Timer 1 comparison A	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0–1 trigger set-up register (0x300298) = 0111
Timer 1 comparison B	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0–1 trigger set-up register (0x300298) = 0110
Timer 2 comparison A	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2–3 trigger set-up register (0x300299) = 0111
Timer 2 comparison B	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2–3 trigger set-up register (0x300299) = 0110
Timer 3 comparison A	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2–3 trigger set-up register (0x300299) = 0111
Timer 3 comparison B	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2–3 trigger set-up register (0x300299) = 0110
Timer 4 comparison B	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0–1 trigger set-up register (0x300298) = 1000
Timer 5 comparison B	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0–1 trigger set-up register (0x300298) = 1000

For HSDMA to be invoked, a 16-bit timer interrupt cause 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 cause of interrupt.

For details on HSDMA transfer, refer to Section II.1, “High-Speed DMA (HSDMA).”

Trap vectors

The trap vector addresses for each default cause of interrupt are set as shown below:

Timer 0 comparison B: 0xC00078
 Timer 0 comparison A: 0xC0007C
 Timer 1 comparison B: 0xC00088
 Timer 1 comparison A: 0xC0008C
 Timer 2 comparison B: 0xC00098
 Timer 2 comparison A: 0xC0009C
 Timer 3 comparison B: 0xC000A8
 Timer 3 comparison A: 0xC000AC
 Timer 4 comparison B: 0xC000B8
 Timer 4 comparison A: 0xC000BC
 Timer 5 comparison B: 0xC000C8
 Timer 5 comparison A: 0xC000CC

The base address of the trap table can be changed using the TTBR register.

IV.1.8 Details of Control Registers

Table IV.1.8.1 List of 16-bit Timer Registers

Address	Register name	Size	Function
0x00300780	16-bit Timer 0 Comparison Data A Setup Register (pT16_CR0A)	16	Sets 16-bit timer 0 comparison data A.
0x00300782	16-bit Timer 0 Comparison Data B Setup Register (pT16_CR0B)	16	Sets 16-bit timer 0 comparison data B.
0x00300784	16-bit Timer 0 Counter Data Register (pT16_TC0)	16	16-bit timer 0 counter data
0x00300786	16-bit Timer 0 Control Register (pT16_CTL0)	16	Controls 16-bit timer 0.
0x00300788	16-bit Timer 1 Comparison Data A Setup Register (pT16_CR1A)	16	Sets 16-bit timer 1 comparison data A.
0x0030078A	16-bit Timer 1 Comparison Data B Setup Register (pT16_CR1B)	16	Sets 16-bit timer 1 comparison data B.
0x0030078C	16-bit Timer 1 Counter Data Register (pT16_TC1)	16	16-bit timer 1 counter data
0x0030078E	16-bit Timer 1 Control Register (pT16_CTL1)	16	Controls 16-bit timer 1.
0x00300790	16-bit Timer 2 Comparison Data A Setup Register (pT16_CR2A)	16	Sets 16-bit timer 2 comparison data A.
0x00300792	16-bit Timer 2 Comparison Data B Setup Register (pT16_CR2B)	16	Sets 16-bit timer 2 comparison data B.
0x00300794	16-bit Timer 2 Counter Data Register (pT16_TC2)	16	16-bit timer 2 counter data
0x00300796	16-bit Timer 2 Control Register (pT16_CTL2)	16	Controls 16-bit timer 2.
0x00300798	16-bit Timer 3 Comparison Data A Setup Register (pT16_CR3A)	16	Sets 16-bit timer 3 comparison data A.
0x0030079A	16-bit Timer 3 Comparison Data B Setup Register (pT16_CR3B)	16	Sets 16-bit timer 3 comparison data B.
0x0030079C	16-bit Timer 3 Counter Data Register (pT16_TC3)	16	16-bit timer 3 counter data
0x0030079E	16-bit Timer 3 Control Register (pT16_CTL3)	16	Controls 16-bit timer 3.
0x003007A0	16-bit Timer 4 Comparison Data A Setup Register (pT16_CR4A)	16	Sets 16-bit timer 4 comparison data A.
0x003007A2	16-bit Timer 4 Comparison Data B Setup Register (pT16_CR4B)	16	Sets 16-bit timer 4 comparison data B.
0x003007A4	16-bit Timer 4 Counter Data Register (pT16_TC4)	16	16-bit timer 4 counter data
0x003007A6	16-bit Timer 4 Control Register (pT16_CTL4)	16	Controls 16-bit timer 4.
0x003007A8	16-bit Timer 5 Comparison Data A Setup Register (pT16_CR5A)	16	Sets 16-bit timer 5 comparison data A.
0x003007AA	16-bit Timer 5 Comparison Data B Setup Register (pT16_CR5B)	16	Sets 16-bit timer 5 comparison data B.
0x003007AC	16-bit Timer 5 Counter Data Register (pT16_TC5)	16	16-bit timer 5 counter data
0x003007AE	16-bit Timer 5 Control Register (pT16_CTL5)	16	Controls 16-bit timer 5.
0x003007D0	DA16 Ch.0 Register (pDA16_CR0A)	16	Sets DA16 Ch.0 comparison data A.
0x003007D2	DA16 Ch.1 Register (pDA16_CR1A)	16	Sets DA16 Ch.1 comparison data A.
0x003007D4	DA16 Ch.2 Register (pDA16_CR2A)	16	Sets DA16 Ch.2 comparison data A.
0x003007DC	Count Pause Register (pT16_CNT_PAUSE)	16	Stops multiple timers simultaneously.
0x003007DE	16-bit Timer STD/ADV Mode Select Register (pT16_ADVMODE)	16	Selects standard or advanced mode.
0x003007E0	16-bit Timer 0 Clock Control Register (pT16_CLKCTL_0)	16	Controls 16-bit timer 0 clock and selects division ratio.
0x003007E2	16-bit Timer 1 Clock Control Register (pT16_CLKCTL_1)	16	Controls 16-bit timer 1 clock and selects division ratio.
0x003007E4	16-bit Timer 2 Clock Control Register (pT16_CLKCTL_2)	16	Controls 16-bit timer 2 clock and selects division ratio.
0x003007E6	16-bit Timer 3 Clock Control Register (pT16_CLKCTL_3)	16	Controls 16-bit timer 3 clock and selects division ratio.
0x003007E8	16-bit Timer 4 Clock Control Register (pT16_CLKCTL_4)	16	Controls 16-bit timer 4 clock and selects division ratio.
0x003007EA	16-bit Timer 5 Clock Control Register (pT16_CLKCTL_5)	16	Controls 16-bit timer 5 clock and selects division ratio.

The following describes each 16-bit timer control register.

The 16-bit timer control registers are mapped in the 16-bit device area from 0x300780 to 0x3007EA, and can be accessed in units of half-words or bytes.

Note: When setting the 16-bit timer control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x300780–0x3007A8: 16-bit Timer *x* Comparison Data A Setup Registers (pT16_CRxA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit timer <i>x</i> comparison data A setup register (pT16_CRxA)	00300780 003007A8 (HW)	D15	CRxA15	16-bit timer <i>x</i> comparison data A	0 to 65535	X	R/W	
		D14	CRxA14	CRxA15 = MSB				
		D13	CRxA13	CRxA0 = LSB				
		D12	CRxA12					
		D11	CRxA11					
		D10	CRxA10					
		D9	CRxA9					
		D8	CRxA8					
		D7	CRxA7					
		D6	CRxA6					
		D5	CRxA5					
		D4	CRxA4					
		D3	CRxA3					
		D2	CRxA2					
		D1	CRxA1					
		D0	CRxA0					

Note: The letter 'x' in bit names, etc., denotes a timer number from 0 to 5.

0x300780 16-bit Timer 0 Comparison Data A Setup Register (pT16_CR0A)

0x300788 16-bit Timer 1 Comparison Data A Setup Register (pT16_CR1A)

0x300790 16-bit Timer 2 Comparison Data A Setup Register (pT16_CR2A)

0x300798 16-bit Timer 3 Comparison Data A Setup Register (pT16_CR3A)

0x3007A0 16-bit Timer 4 Comparison Data A Setup Register (pT16_CR4A)

0x3007A8 16-bit Timer 5 Comparison Data A Setup Register (pT16_CR5A)

D[15:0] CRxA[15:0]: 16-bit Timer *x* Comparison Data A Bits

Sets the comparison data A for each timer. (Default: indeterminate)

When SELCRB_{*x*} (D5/0x300786 + 8•*x*) is set to 0, comparison data is directly read or writing from/to the comparison data register A.

When SELCRB_{*x*} 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 (OUTINV_{*x*} (D4/0x300786 + 8•*x*) = 0) or falls (OUTINV_{*x*} = 1). This does not affect the counter value and count-up operation.

0x300782–0x3007AA: 16-bit Timer *x* Comparison Data B Setup Registers (pT16_CRxB)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit timer <i>x</i> comparison data B setup register (pT16_CRxB)	00300782	D15	CRxB15	16-bit timer <i>x</i> comparison data B CRxB15 = MSB CRxB0 = LSB	0 to 65535	X	R/W	
		D14	CRxB14					
	003007AA (HW)	D13	CRxB13					
		D12	CRxB12					
		D11	CRxB11					
		D10	CRxB10					
		D9	CRxB9					
		D8	CRxB8					
		D7	CRxB7					
		D6	CRxB6					
		D5	CRxB5					
		D4	CRxB4					
		D3	CRxB3					
		D2	CRxB2					
		D1	CRxB1					
		D0	CRxB0					

Note: The letter ‘*x*’ in bit names, etc., denotes a timer number from 0 to 5.

- 0x300782 16-bit Timer 0 Comparison Data B Setup Register (pT16_CR0B)
- 0x30078A 16-bit Timer 1 Comparison Data B Setup Register (pT16_CR1B)
- 0x300792 16-bit Timer 2 Comparison Data B Setup Register (pT16_CR2B)
- 0x30079A 16-bit Timer 3 Comparison Data B Setup Register (pT16_CR3B)
- 0x3007A2 16-bit Timer 4 Comparison Data B Setup Register (pT16_CR4B)
- 0x3007AA 16-bit Timer 5 Comparison Data B Setup Register (pT16_CR5B)

D[15:0] CRxB[15:0]: 16-bit Timer *x* Comparison Data B Bits

Sets the comparison data B for each timer. (Default: indeterminate)

When SELCRB_{*x*} (D5/0x300786 + 8•*x*) is set to 0, comparison data is directly read or writing from/to the comparison data register B.

When SELCRB_{*x*} 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 (OUTINV_{*x*} (D4/0x300786 + 8•*x*) = 0) or rises (OUTINV_{*x*} = 1). Furthermore, the counter is reset to 0.

0x300784–0x3007AC: 16-bit Timer x Counter Data Registers (pT16_TCx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit timer x counter data register (pT16_TCx)	00300784 003007AC (HW)	D15	TCx15	16-bit timer x counter data	0 to 65535	X	R/W	Data can be written only in advanced mode.
		D14	TCx14	TCx15 = MSB				
		D13	TCx13	TCx0 = LSB				
		D12	TCx12					
		D11	TCx11					
		D10	TCx10					
		D9	TCx9					
		D8	TCx8					
		D7	TCx7					
		D6	TCx6					
		D5	TCx5					
		D4	TCx4					
		D3	TCx3					
		D2	TCx2					
		D1	TCx1					
		D0	TCx0					

Note: The letter 'x' in bit names, etc., denotes a timer number from 0 to 5.

0x300784 16-bit Timer 0 Counter Data Register (pT16_TC0)

0x30078C 16-bit Timer 1 Counter Data Register (pT16_TC1)

0x300794 16-bit Timer 2 Counter Data Register (pT16_TC2)

0x30079C 16-bit Timer 3 Counter Data Register (pT16_TC3)

0x3007A4 16-bit Timer 4 Counter Data Register (pT16_TC4)

0x3007AC 16-bit Timer 5 Counter Data Register (pT16_TC5)

IV

T16

D[15:0] TCx[15:0]: 16-bit Timer x Counter Data Bits

The counter data of each timer can be read from this register. (Default: indeterminate)

The data can be read out at any time.

In advanced mode, counter data can be written at any time. This makes it possible to change the interrupt and/or clock output cycles temporarily. Standard mode does not allow writing of counter data.

0x300786–0x3007AE: 16-bit Timer x Control Registers (pT16_CTLx)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer x control register (pT16_CTLx)	00300786	D15–9	–	reserved	–			–	–	0 when being read.	
		D8	INITOLx	16-bit timer x initial output level	1	High	0	Low	0	R/W	Advanced mode
	003007AE (HW)	D7	(TMODEx)	(reserved for 16-bit timer x test)	1	Test mode	0	Normal	0	R	Do not write 1.
		D6	SELFMx	16-bit timer x fine mode selection	1	Fine mode	0	Normal	0	R/W	
		D5	SELCRBx	16-bit timer x comparison buffer	1	Enabled	0	Disabled	0	R/W	
		D4	OUTINVx	16-bit timer x output inversion	1	Invert	0	Normal	0	R/W	
		D3	CKSLx	16-bit timer x input clock selection	1	External clock	0	Internal clock	0	R/W	
		D2	PTMx	16-bit timer x clock output control	1	On	0	Off	0	R/W	
		D1	PRESETx	16-bit timer x reset	1	Reset	0	Invalid	0	W	0 when being read.
		D0	PRUNx	16-bit timer x Run/Stop control	1	Run	0	Stop	0	R/W	

Note: The letter ‘x’ in bit names, etc., denotes a timer number from 0 to 5.

- 0x300786 16-bit Timer 0 Control Register (pT16_CTL0)
- 0x30078E 16-bit Timer 1 Control Register (pT16_CTL1)
- 0x300796 16-bit Timer 2 Control Register (pT16_CTL2)
- 0x30079E 16-bit Timer 3 Control Register (pT16_CTL3)
- 0x3007A6 16-bit Timer 4 Control Register (pT16_CTL4)
- 0x3007AE 16-bit Timer 5 Control Register (pT16_CTL5)

D[15:9] Reserved

D8 **INITOLx: 16-bit Timer x Initial Output Level Select Bit (Advanced Mode)**

Selects an initial output level for timer output.

1 (R/W): High

0 (R/W): Low (default)

The timer output pin goes to the initial output level set using this bit when the timer output is turned off by writing 0 to PTMx (D2) or when the timer is reset by writing 1 to PRESETx (D1). However, this level is inverted if OUTINVx (D4) is set to 1.

Note that writing to this bit is enabled only in advanced mode.

D7 **(TMODEx): Reserved**

Do not set this bit to 1.

D6 **SELFMx: 16-bit Timer x Fine Mode Select Bit**

Sets fine mode for clock output.

1 (R/W): Fine mode

0 (R/W): Normal output (default)

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.

D5 **SELCRBx: 16-bit Timer x Comparison Buffer Enable Bit**

Enables or disables writing to the comparison register buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

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.

D4 OUTINV_x: 16-bit Timer x Output Inversion Bit

Selects a logic of the output signal.

1 (R/W): Inverted (active low)

0 (R/W): Normal (active high) (default)

By writing 1 to OUTINV_x, an active-low signal (off level = high) is generated for the TM_x output.

When OUTINV_x is set to 0, an active-high signal (off level = low) is generated.

Writing 1 to this bit inverts the initial output level set using INITOL_x (D8) as well.

D3 CKSL_x: 16-bit Timer x Input Clock Select Bit

Selects the input clock for each timer.

1 (R/W): External clock

0 (R/W): Internal clock (default)

The internal clock (prescaler output) is selected for the input clock of each timer by writing 0 to CKSL_x. 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 the corresponding port function select register before an external clock is selected here.

D2 PTM_x: 16-bit Timer x Clock Output Control Bit

Controls the output of the TM_x signal (timer output clock).

1 (R/W): On

0 (R/W): Off (default)

The TM_x signal is output from the clock output pin by writing 1 to PTM_x. Clock output is stopped by writing 0 to PTM_x and goes to the off level according to the set values of OUTINV_x (D4) and INITOL_x (D8). In this case, the clock output pin must be set using the corresponding port function select register before outputting the TM_x signal here.

D1 PRESET_x: 16-bit Timer x Reset Bit

Preset the reload data in the counter.

1 (W): Reset

0 (W): Has no effect

0 (R): Always 0 when read (default)

The counter of timer *x* is reset by writing 1 to PRESET_x.

D0 PRUN_x: 16-bit Timer x Run/Stop Control Bit

Controls the timer's RUN/STOP state.

1 (R/W): Run

0 (R/W): Stop (default)

Each timer is made to start counting up by writing 1 to PRUN_x 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.

0x3007D0–0x3007D4: DA16 Ch.x Registers (pDA16_CRxA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
DA16 Ch.x register (pDA16_CRxA)	003007D0 003007D4 (HW)	D15	DAxA15	DA16 Ch.x comparison data A DAxA15 = MSB DAxA0 = LSB	0 to 65535	X	R/W	Advanced mode
		D14	DAxA14					
		D13	DAxA13					
		D12	DAxA12					
		D11	DAxA11					
		D10	DAxA10					
		D9	DAxA9					
		D8	DAxA8					
		D7	DAxA7					
		D6	DAxA6					
		D5	DAxA5					
		D4	DAxA4					
		D3	DAxA3					
		D2	DAxA2					
		D1	DAxA1					
		D0	DAxA0					

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 2.

- 0x3007D0 DA16 Ch.0 Register (pDA16_CR0A)
- 0x3007D2 DA16 Ch.1 Register (pDA16_CR1A)
- 0x3007D4 DA16 Ch.2 Register (pDA16_CR2A)

D[15:0] DAxA[15:0]: DA16 Ch.x Comparison Data A Bits (Advanced Mode)

Sets the comparison data A for each channel in DA16 mode. (Default: indeterminate)

The following shows the correspondence between these registers and timers:

(timer A and timer B)

- DA16 Ch.0 Register: timer 1 and timer 2
- DA16 Ch.1 Register: timer 3 and timer 4
- DA16 Ch.2 Register: timer 5 and timer 0

When data is written to this register, 10 high-order bits (DAxA[15:6]) are loaded into the Timer A Comparison Data A Setup Register (buffer) as 10 low-order compare data bits and 6 low-order bits (DAxA[5:0]) are loaded into the Timer B Comparison Data A Setup Register (buffer) as 6 low-order compare data bits.

This makes it possible to reduce software load for using two 16-bit timers as a 16-bit D/A converter.

Note that writing to this register is enabled only in advanced mode.

0x3007DC: Count Pause Register (pT16_CNT_PAUSE)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Count pause register (pT16_CNT_PAUSE)	003007DC (HW)	D15-6	–	reserved	–		–	–	0 when being read.
		D5	PAUSE5	16-bit timer 5 count pause	1	Pause	0	R/W	Advanced mode
		D4	PAUSE4	16-bit timer 4 count pause	0	Count	0	R/W	
		D3	PAUSE3	16-bit timer 3 count pause	0		R/W		
		D2	PAUSE2	16-bit timer 2 count pause	0		R/W		
		D1	PAUSE1	16-bit timer 1 count pause	0		R/W		
		D0	PAUSE0	16-bit timer 0 count pause	0		R/W		

Since the timer RUN/STOP control bits are located in different addresses, two or more timers cannot be started at the same time. To synchronize multiple timers, the control bits PAUSE_x that stop each timer are provided in this register.

1 (R/W): Pause

0 (R/W): Count (default)

When PAUSE_x is set to 1, timer *x* is placed in pause state and when set to 0, timer *x* starts counting or continues stop state according to the set value of PRUN_x (D0/0x300786 + 8•*x*).

Note that writing to this bit is enabled only in advanced mode.

The following shows a procedure to synchronize multiple timers.

1. Set the prescaler clocks for the timers to be synchronized to the same condition.
2. Set PAUSE_x for the timers to 1 to place the timers in pause state.
3. Set PRUN_x (D0/0x300786 + 8•*x*) for the timers to 1.
4. Set all the PAUSE_x bits for the timers to 0 at the same time.

D[15:6] Reserved**D5 PAUSE5: 16-bit Timer 5 Count Pause Bit (Advanced Mode)**

Stops the counter in 16-bit timer 5.

D4 PAUSE4: 16-bit Timer 4 Count Pause Bit (Advanced Mode)

Stops the counter in 16-bit timer 4.

D3 PAUSE3: 16-bit Timer 3 Count Pause Bit (Advanced Mode)

Stops the counter in 16-bit timer 3.

D2 PAUSE2: 16-bit Timer 2 Count Pause Bit (Advanced Mode)

Stops the counter in 16-bit timer 2.

D1 PAUSE1: 16-bit Timer 1 Count Pause Bit (Advanced Mode)

Stops the counter in 16-bit timer 1.

D0 PAUSE0: 16-bit Timer 0 Count Pause Bit (Advanced Mode)

Stops the counter in 16-bit timer 0.

0x3007DE: 16-bit Timer STD/ADV Mode Select Register (pT16_ADVMODE)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
16-bit timer STD/ADV mode select register (pT16_ADVMODE)	003007DE (HW)	D15-1	–	reserved	–			–	–	Writing 1 not allowed.
		D0	T16ADV	Standard mode/advanced mode select	1	Advanced mode	0	Standard mode	0	R/W

D[15:1] Reserved

D0 T16ADV: Standard/Advanced Mode Select Bit

Selects standard or advanced mode.

1 (R/W): Advanced mode

0 (R/W): Standard mode (default)

The 16-bit timer in the S1C33E08 is extended from that of the C33 STD models. This 16-bit timer has two operating modes, standard (STD) mode of which functions are compatible with the existing C33 STD models and an advanced (ADV) mode allowing use of the extended functions. Table IV.1.8.2 shows differences between standard mode and advanced mode.

Table IV.1.8.2 Differences between Standard Mode and Advanced Mode

Function	Standard mode	Advanced mode
Writing to the count data register	Disabled (read only)	Enabled
Setting of the initial timer output level (high or low)	Disabled (depending on the OUTINVx set value)	Enabled (can be specified using INITOLx)
DA16 function (DA16 registers)	Cannot be used	Can be used
Multiple timer full-sync function	Not supported	Supported (can be controlled using PAUSEx)

To configure the 16-bit timer in advanced mode, set this bit to 1. The control registers/bits for the extended functions are enabled to write after this setting.

Note: Standard or advanced mode currently set is applied to all the 16-bit timers. It cannot be selected for each timer individually.

0x3007E0–0x3007EA: 16-bit Timer x Clock Control Registers (pT16_CLKCTL_x)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
16-bit timer x clock control register (pT16_CLKCTL_x)	003007E0 003007EA (HW)	D15–4	–	reserved	–	–	–	0 when being read.		
		D3	P16TONx	16-bit timer x clock control	1 On 0 Off	0	R/W			
		D2	P16TSx2	16-bit timer x clock division ratio selection	P16TSx[2:0]	Division ratio	0	R/W		
		D1	P16TSx1				111	MCLK/4096	0	
		D0	P16TSx0				110	MCLK/1024	0	
							101	MCLK/256		
							100	MCLK/64		
		011	MCLK/16							
		010	MCLK/4							
		001	MCLK/2							
		000	MCLK/1							

Note: The letter ‘x’ in bit names, etc., denotes a timer number from 0 to 5.

- 0x3007E0 16-bit Timer 0 Clock Control Register (pT16_CLKCTL_0)
- 0x3007E2 16-bit Timer 1 Clock Control Register (pT16_CLKCTL_1)
- 0x3007E4 16-bit Timer 2 Clock Control Register (pT16_CLKCTL_2)
- 0x3007E6 16-bit Timer 3 Clock Control Register (pT16_CLKCTL_3)
- 0x3007E8 16-bit Timer 4 Clock Control Register (pT16_CLKCTL_4)
- 0x3007EA 16-bit Timer 5 Clock Control Register (pT16_CLKCTL_5)

D[15:4] Reserved

D3 P16TONx: 16-bit Timer x Clock Control Bit

Controls the clock supply to 16-bit timer x.

1 (R/W): On

0 (R/W): Off (default)

D[2:0] P16TSx[2:0]: 16-bit Timer x Clock Division Ratio Setup Bits

Selects a division ratio to generate the 16-bit timer x clock.

Table IV.1.8.3 Selecting Division Ratio

P16TSx2	P16TSx1	P16TSx0	Division ratio
1	1	1	MCLK/4096
1	1	0	MCLK/1024
1	0	1	MCLK/256
1	0	0	MCLK/64
0	1	1	MCLK/16
0	1	0	MCLK/4
0	0	1	MCLK/2
0	0	0	MCLK/1

(Default: 0b000)

IV.1.9 Precautions

- When setting the count clock or operation mode, make sure the 16-bit timer is turned off.
- 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.
- 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 $\times 1/2$.
- 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 $A > 2 \times B + 1$.
- After an initial reset, the cause-of-interrupt flag becomes indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in the software.
- To prevent another interrupt from being generated by the same cause of interrupt after an interrupt has occurred, be sure to reset the cause-of-interrupt flag before setting the PSR again or executing the reti instruction.

IV.2 Watchdog Timer (WDT)

IV.2.1 Configuration of the Watchdog Timer

The S1C33E08 incorporates a watchdog timer to detect the CPU running uncontrollably. The watchdog timer consists of a 30-bit up counter and comparison data register for generating an NMI or internal reset signal at programmable cycles.

By resetting the watchdog timer within such a cycle in software so as not to generate NMI or internal reset signals, it is possible to detect a program running uncontrollably that does not execute that processing routine.

The WDT clock (= MCLK) or external clock input for 16-bit timer 0 (EXCL0) can be selected as the count clock for the watchdog timer.

Moreover, a clock can be generated synchronously with NMI/reset generation cycles (set by the comparison data register) and output from the watchdog timer to external devices.

Figure IV.2.1.1 shows a block diagram of the watchdog timer.

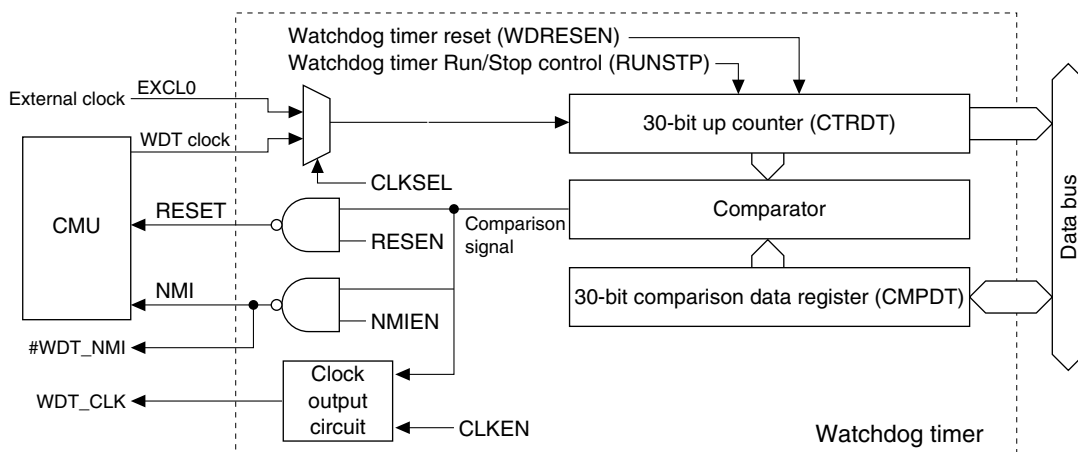


Figure IV.2.1.1 Block Diagram of Watchdog Timer

IV.2.2 Input/Output Pins of the Watchdog Timer

Table IV.2.2.1 Input/Output Pins of Watchdog Timer

Pin name	I/O	Function
EXCL0	I	External clock input pin (external clock input for 16-bit timer 0)
WDT_CLK	O	Watchdog timer clock output pin
#WDT_NMI	O	Watchdog timer NMI output pin

The EXCL0 pin is used to clock the counter of the watchdog timer with an external clock.

The WDT_CLK pin is used to output the clock generated in the watchdog timer to external devices.

The #WDT_NMI pin is used to output the NMI signal generated in the watchdog timer to external devices.

Note: These pins are shared with general-purpose input/output ports or other peripheral circuit input/output pins, and set for other than the watchdog timer function by default. Therefore, before these pins can be used as input/output ports for the watchdog timer clock, the corresponding Port Function Select Register must be set to switch over the pin functions.

For details about pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

IV.2.3 Operating Clock of the Watchdog Timer

The watchdog timer module is clocked by the WDT clock (= MCLK) supplied from the CMU. At initial reset, this clock is selected as the operating clock for the watchdog timer. While the watchdog timer remains idle or is not being used, the clock supplied from the CMU can be turned off to reduce the amount of current consumed on the chip. Use WDT_CKE (D9/0x301B04) for this control.

* **WDT_CKE**: Watchdog Timer Clock Control Bit in the Gated Clock Control Register 1 (D9/0x301B04)

Setting WDT_CKE (D9/0x301B04) to 0 turns off the clock supplied from the CMU to the watchdog timer.

For details about clock generation and control, see Section III.1, “Clock Management Unit (CMU).”

- Notes:**
- Even when using an external clock as the count clock for the watchdog timer, the WDT clock is required for watchdog timer operation and access to its control register.
 - The Gated Clock Control Register 1 (0x301B04) is write-protected. To rewrite this register and other CMU control registers at addresses 0x301B00 to 0x301B14, write protection must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 may cause the system to operate erratically, make sure that data set in the Clock Control Protect Register (0x301B24) is other than 0x96 unless rewriting said registers.

IV.2.4 Control of the Watchdog Timer

IV.2.4.1 Setting Up the Watchdog Timer

Selecting the count clock

The internal clock (MCLK) or external clock (EXCL0) can be selected as the count clock for the 30-bit up-counter by using CLKSEL (D6/0x300662).

* **CLKSEL**: Watchdog Timer Input Clock Select Bit in Watchdog Timer Enable Register (D6/0x300662)

Setting CLKSEL (D6/0x300662) to 0 (default) selects the internal clock (MCLK); setting it to 1 selects the external clock (EXCL0). Therefore, before an external clock can be used, the function of the pin set as an I/O port by default must be switched to EXCL0 (external clock input for 16-bit timer 0) by using the Port Function Select Register. For details about pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

For details about MCLK generation and control, see Section III.1, “Clock Management Unit (CMU).”

Setting the NMI/reset generation cycle

The watchdog timer has a 30-bit comparison data register that can be used to set a cycle in which to generate an NMI or reset signal.

- * **CMPDT[15:0]**: 16 Low-order Comparison Data Bits in the Watchdog Timer Comparison Data Setup Register 0 (D[15:0]/0x300664)
- * **CMPDT[29:16]**: 14 High-order Comparison Data Bits in the Watchdog Timer Comparison Data Setup Register 1 (D[13:0]/0x300666)

The data set in these register bits is compared with the up-counter value. When both match, a specified NMI or reset signal is output. The up-counter is reset to 0 at this time.

The NMI/reset generation cycle can be calculated from the equation below.

$$\text{NMI generating cycle} = \frac{\text{CMPDT} + 1}{f_{\text{WDTIN}}} \text{ [sec]}$$

where

CMPDT = value set in CMPDT[29:0] (D[13:0]/0x300666, D[15:0]/0x300664)

f_{WDTIN} = MCLK or EXCL0 input clock frequency [Hz]

For example, the specifiable maximum NMI/reset generation cycle is about 21.47 seconds at 50-MHz MCLK input.

Note: Do not set a value equal to or less than 0x0000001F in the comparison data register.

Selecting the NMI/reset generation function

To output an NMI signal when the watchdog timer is not reset within a specified cycle, set NMIEN (D1/0x300662) to 1. To output a reset signal instead, set RESEN (D0/0x300662) to 1.

- * **NMIEN**: Watchdog Timer NMI Enable Bit in the Watchdog Timer Enable Register (D1/0x300662)
- * **RESEN**: Watchdog Timer RESET Enable Bit in the Watchdog Timer Enable Register (D0/0x300662)

Setting both bits to 0 (default) generates neither an NMI signal nor a reset signal, although the up-counter remains active and can output a clock.

Setting both bits to 1 outputs both an NMI signal and a reset signal. In this case, however, reset exception handling is executed since it has priority over the NMI exception handling.

The NMI and reset signals are both output as pulses of 32 MCLK clocks in width.

Note: Depending on the counter and comparison register values, an NMI or reset signal may be generated after the NMI or reset function is enabled here (or even when the watchdog timer has not yet been started). Always be sure to set comparison data and reset the watchdog timer before writing 1 to NMIEN (D1/0x300662) or RESEN (D0/0x300662).

Write protection of watchdog timer registers

The Watchdog Timer Enable Register (0x300662) and Watchdog Timer Comparison Data Registers (0x300664, 0x300666) are write-protected to prevent NMI or reset signals from being inadvertently generated by unnecessary write operations. To rewrite these registers, write protection must be removed by writing 0x96 to the Watchdog Timer Write-Protect Register (0x300660). Once the registers are rewritten, be sure to write other than 0x96 to the Watchdog Timer Write-Protect Register (0x300660) to reapply write protection.

IV.2.4.2 Starting/Stopping the Watchdog Timer

Writing 1 to RUNSTP (D4/0x300662) starts counting by the watchdog timer; writing 0 stops the watchdog timer.

* **RUNSTP**: Watchdog Timer Run/Stop Control Bit in the Watchdog Timer Enable Register (D4/0x300662)

Since RUNSTP (D4/0x300662) exists in the write-protected Watchdog Timer Enable Register (0x300662), write protection must be removed by writing 0x96 to the Watchdog Timer Write-Protect Register (0x300660) before the content of RUNSTP can be altered.

IV.2.4.3 Resetting the Watchdog Timer

Before the NMI/reset generation function of the watchdog timer can be used, a routine to reset the watchdog timer before NMI or reset generation must be prepared in a location for periodic processing. Make sure that this routine is processed within the NMI/reset generation cycle described earlier.

Writing 1 to WDRESEN (D0/0x30066C) resets the watchdog timer. The up-counter is reset to 0 at this time, then starts counting NMI/reset generation cycles all over again.

* **WDRESEN**: Watchdog Timer Reset Bit in the Watchdog Timer Control Register (D0/0x30066C)

If the watchdog timer is not reset within the set cycle for some reason, the CPU is placed into trap handling by an NMI or reset signal to execute the processing routine.

The reset and NMI trap vector addresses are set by default to 0xC00000 and 0xC0001C, respectively. The trap table base address can be altered by using TTBR.

The count value of the up-counter can be read out from the Watchdog Timer Count Registers (0x300668, 0x30066A) at any time.

* **CTRDT[15:0]**: 16 Low-order Counter Data Bits in the Watchdog Timer Count Register 0 (D[15:0]/0x300668)

* **CTRDT[29:16]**: 14 High-order Counter Data Bits in the Watchdog Timer Count Register 1 (D[13:0]/0x30066A)

IV.2.4.4 Operation in Standby Mode

In HALT mode

In HALT mode, the watchdog timer remains active as it is supplied with a clock. Therefore, if HALT mode remains active beyond the NMI/reset generation cycle, an NMI or reset signal deactivates HALT mode.

To disable the watchdog timer in HALT mode, set NMIEN (D1/0x300662) or RESEN (D0/0x300662) to 0. Otherwise, write 0 to RUNSTP (D4/0x300662) to stop the watchdog timer before executing the halt instruction. When NMIEN (D1/0x300662) or RESEN (D0/0x300662) disables NMI or reset generation, the watchdog timer continues counting even in HALT mode. To reenble NMI or reset generation after exiting HALT mode, be sure to reset the watchdog timer beforehand.

When HALT mode is entered after stopping the watchdog timer, be sure to reset the watchdog timer before re-starting it.

In SLEEP mode

The supply of MCLK from the CMU stops in SLEEP mode. Therefore, the watchdog timer also stops operating. To prevent an unnecessary NMI or reset signal from being generated after exiting SLEEP mode, be sure to reset the watchdog timer before executing the slp instruction. Moreover, disable NMI/reset generation by setting NMIEN (D1/0x300662) or RESEN (D0/0x300662) as required.

IV.2.4.5 Clock Output of the Watchdog Timer

The watchdog timer can output an NMI/reset generation cycle-synchronous clock from the IC to external devices. For this clock output, set CLKEN (D5/0x300662) to 1 after setting up the WDT_CLK pin.

* **CLKEN**: Watchdog Timer Clock Output Control Bit in the Watchdog Timer Enable Register (D5/0x300662)

Since CLKEN (D5/0x300662) also exists in the write-protected Watchdog Timer Enable Register (0x300662), write protection must be removed by writing 0x96 to the Watchdog Timer Write-Protect Register (0x300660) before the content of CLKEN can be altered.

If the watchdog timer is not reset in software, the level of clock output from the IC is reversed synchronously with the NMI generation cycles. (This applies when reset generation is disabled.)

When the watchdog timer is reset in software, clock output from the IC goes low at that time and remains low.

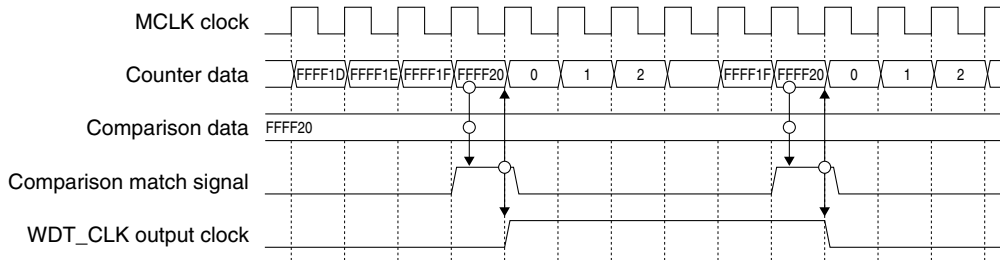


Figure IV.2.4.5.1 Clock Output of Watchdog Timer

IV.2.4.6 External NMI Output

The watchdog timer can output the NMI signal generated to external devices. The watchdog timer uses the #WDT_NMI pin for this output. This pin is configured as a general-purpose I/O pin at initial reset, therefore, the pin function must be set as #WDT_NMI (see Section I.3.3).

Setting NMIEN (D1/0x300662) to 1 enables the external NMI signal output as well as the internal NMI signal output. When the watchdog timer counter reaches the comparison data, the #WDT_NMI pin outputs a low pulse with 32 MCLK clock cycles.

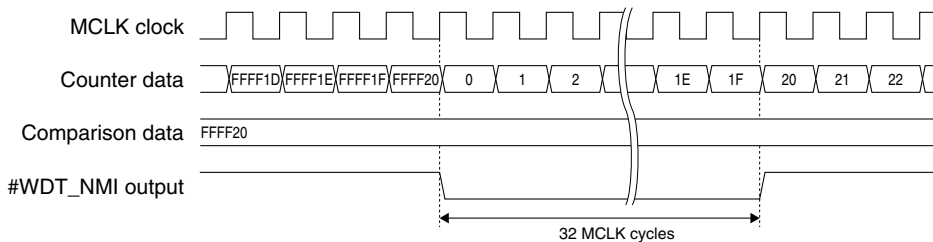


Figure IV.2.4.6.1 External NMI Output

IV.2.5 Details of Control Registers

Table IV.2.5.1 List of Watchdog Timer Control Registers

Address	Register name	Size	Function
0x00300660	Watchdog Timer Write-Protect Register (pWD_WP)	16	Enables/disables WDT control registers for writing.
0x00300662	Watchdog Timer Enable Register (pWD_EN)	16	Configures and starts watchdog timer.
0x00300664	Watchdog Timer Comparison Data Setup Register 0 (pWD_COMP_LOW)	16	Sets comparison data (16 low-order bits).
0x00300666	Watchdog Timer Comparison Data Setup Register 1 (pWD_COMP_HIGH)	16	Sets comparison data (14 high-order bits).
0x00300668	Watchdog Timer Count Register 0 (pWD_CNT_LOW)	16	Watchdog timer counter data (16 low-order bits)
0x0030066A	Watchdog Timer Count Register 1 (pWD_CNT_HIGH)	16	Watchdog timer counter data (14 high-order bits)
0x0030066C	Watchdog Timer Control Register (pWD_CNTL)	16	Resets watchdog timer.

The following describes the watchdog timer control registers.

The watchdog timer control registers are mapped to the 16-bit device area at addresses 0x300660 to 0x30066C, and can be accessed in units of half-words.

- Notes:**
- The watchdog timer control registers allow accessing in half-word size only (except for 0x300662 and 0x30066C that allow byte access as well as half-word access). Do not read/write the registers in byte size.
 - When setting the watchdog timer control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x300660: Watchdog Timer Write-Protect Register (pWD_WP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog timer write-protect register (pWD_WP)	00300660 (HW)	D15	WDPTC15	Watchdog timer register write-protect	Writing 0x96 removes the write protection of the watchdog timer enable and comparison data registers (0x300662–0x300666). Writing another value set the write protection.	X	W	0 when being read.
		D14	WDPTC14					
		D13	WDPTC13					
		D12	WDPTC12					
		D11	WDPTC11					
		D10	WDPTC10					
		D9	WDPTC9					
		D8	WDPTC8					
		D7	WDPTC7					
		D6	WDPTC6					
		D5	WDPTC5					
		D4	WDPTC4					
		D3	WDPTC3					
		D2	WDPTC2					
		D1	WDPTC1					
		D0	WDPTC0					

D[15:0] WDPTC[15:0]: Watchdog Timer Write-Protect Bits

These bits set or clear write protection at addresses 0x300662 to 0x300666.

0x96 (W): Clears write protection

Other than 0x96 (W): Applies write protection (default, indeterminate value)

0x0 (R): Always 0x0 when read

Before altering the Watchdog Timer Enable Register (0x300662) or Watchdog Timer Comparison Data Registers (0x300664, 0x300666), write 0x96 to this register to remove write protection. Setting this register to other than 0x96 will result in the contents of the registers above not being altered even when executing the write instruction without any problem. Once write protection is removed by writing 0x96 to this register, said registers can be rewritten any number of times until this register is set to other than 0x96. When the clock control registers have been rewritten, be sure to write other than 0x96 to this register to prevent erroneous writing to said registers.

0x300662: Watchdog Timer Enable Register (pWD_EN)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
Watchdog timer enable register (pWD_EN)	00300662 (HW)	D15-7	–	reserved	–			–	–	0 when being read.	
		D6	CLKSEL	Watchdog timer input clock select	1	External clock	0	Internal clock	0	R/W	
		D5	CLKEN	Watchdog timer clock output control	1	On	0	Off	0	R/W	
		D4	RUNSTP	Watchdog timer Run/Stop control	1	Run	0	Stop	0	R/W	
		D3-2	–	reserved	–			–	–	0 when being read.	
		D1	NMIEN	Watchdog timer NMI enable	1	Enabled	0	Disabled	0	R/W	
		D0	RESEN	Watchdog timer RESET enable	1	Enabled	0	Disabled	0	R/W	

Note: This register is write-protected to prevent NMI or reset signals from being inadvertently generated by unnecessary write operations. To rewrite this register, write protection must be removed by writing 0x96 to the Watchdog Timer Write-Protect Register (0x300660). Once the register has been rewritten, be sure to write other than 0x96 to the Watchdog Timer Write-Protect Register (0x300660) to reapply write protection.

D[15:7] Reserved**D6 CLKSEL: Watchdog Timer Input Clock Select Bit**

This bit selects the count clock for the watchdog timer.

1 (R/W): External clock (EXCL0)

0 (R/W): Internal clock (MCLK) (default)

Setting this bit to 0 (default) selects the internal clock (MCLK); setting it to 1 selects the external clock (EXCL0). Before an external clock can be used, the function of the pin set by default as an I/O port must be switched to EXCL0 (external clock input for 16-bit timer 0) by using the Port Function Select Register. For details about pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

D5 CLKEN: Watchdog Timer Clock Output Control Bit

This bit controls the clock output of the watchdog timer.

1 (R/W): On

0 (R/W): Off (default)

Setting this bit to 1 outputs an NMI/reset generation cycle-synchronous clock from the IC. Before this clock output can be used, however, the function of the pin set by default as an I/O port must be switched to WDT_CLK (watchdog timer clock output) by using the Port Function Select Register. For details about pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

D4 RUNSTP: Watchdog Timer Run/Stop Control Bit

This bit starts or stops the watchdog timer.

1 (R/W): Start

0 (R/W): Stop (default)

When the NMI or reset generation function is enabled, be sure to set comparison data and reset the watchdog timer before starting the watchdog timer, thus preventing the generation of unnecessary NMI or reset signals.

D[3:2] Reserved

D1 **NMIEN: Watchdog Timer NMI Enable Bit**

This bit enables NMI signal output by the watchdog timer.

1 (R/W): Enable

0 (R/W): Disable (default)

Setting this bit to 1 outputs an NMI signal (a pulse 32 MCLK clocks in width) to the CMU and the #WDT_NMI pin when the count of the up-counter matches the value set in the comparison data register. Setting this bit to 0 outputs no NMI signals.

Regardless of how this bit is set, the up-counter is reset to 0 when the up-counter and set value of the comparison data register match, then starts counting all over again.

D0 **RESEN: Watchdog Timer RESET Enable Bit**

This bit enables internal reset signal output by the watchdog timer.

1 (R/W): Enable

0 (R/W): Disable (default)

Setting this bit to 1 outputs a reset signal (a pulse 32 MCLK clocks in width) to the CMU when the count of the up-counter matches the value set in the comparison data register. Setting this bit to 0 outputs no reset signals.

0x300664: Watchdog Timer Comparison Data Setup Register 0 (pWD_COMP_LOW)**0x300666: Watchdog Timer Comparison Data Setup Register 1 (pWD_COMP_HIGH)**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog timer comparison data setup register 0 (pWD_COMP_LOW)	00300664 (HW)	D15	CMPDT15	Watchdog timer comparison data CMPDT0 = LSB	0x0 to 0x3FFFFFFF (low-order 16 bits)	0	R/W	
		D14	CMPDT14					
		D13	CMPDT13					
		D12	CMPDT12					
		D11	CMPDT11					
		D10	CMPDT10					
		D9	CMPDT9					
		D8	CMPDT8					
		D7	CMPDT7					
		D6	CMPDT6					
		D5	CMPDT5					
		D4	CMPDT4					
		D3	CMPDT3					
		D2	CMPDT2					
		D1	CMPDT1					
		D0	CMPDT0					
Watchdog timer comparison data setup register 1 (pWD_COMP_HIGH)	00300666 (HW)	D15-14	–	reserved	–	–	–	0 when being read.
		D13	CMPDT29	Watchdog timer comparison data CMPDT29 = MSB	0x0 to 0x3FFFFFFF (high-order 14 bits)	0	R/W	
		D12	CMPDT28					
		D11	CMPDT27					
		D10	CMPDT26					
		D9	CMPDT25					
		D8	CMPDT24					
		D7	CMPDT23					
		D6	CMPDT22					
		D5	CMPDT21					
		D4	CMPDT20					
		D3	CMPDT19					
		D2	CMPDT18					
		D1	CMPDT17					
		D0	CMPDT16					

Note: These registers are write-protected to prevent NMI or reset signals from being inadvertently generated by unnecessary write operations. To rewrite these registers, write protection must be removed by writing 0x96 to the Watchdog Timer Write-Protect Register (0x300660). Once the registers have been rewritten, be sure to write other than 0x96 to the Watchdog Timer Write-Protect Register (0x300660) to reapply write protection.

Use these registers to set the NMI/reset generation cycle.

With NMI or reset generation enabled, an NMI or reset signal is output when the up-counter matches the comparison data set in these registers.

When a clock is output from the watchdog timer, these registers also set the output clock cycle.

D[15:0]/0x300664 CMPDT[15:0]: Watchdog Timer Comparison Data (16 low-order bits)

The 16 low-order bits of comparison data are set in these bits. (Default: 0x0000)

D[13:0]/0x300666 CMPDT[29:16]: Watchdog Timer Comparison Data (14 high-order bits)

The 14 high-order bits of comparison data are set in these bits. (Default: 0x0000)

Note: Do not set a value equal to or less than 0x0000001F as comparison data.

0x300668: Watchdog Timer Count Register 0 (pWD_CNT_LOW)

0x30066A: Watchdog Timer Count Register 1 (pWD_CNT_HIGH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog timer count register 0 (pWD_CNT_LOW)	00300668 (HW)	D15	CTRDT15	Watchdog timer counter data CTRDT0 = LSB	0x0 to 0x3FFFFFFF (low-order 16 bits)	X	R	
		D14	CTRDT14					
		D13	CTRDT13					
		D12	CTRDT12					
		D11	CTRDT11					
		D10	CTRDT10					
		D9	CTRDT9					
		D8	CTRDT8					
		D7	CTRDT7					
		D6	CTRDT6					
		D5	CTRDT5					
		D4	CTRDT4					
		D3	CTRDT3					
		D2	CTRDT2					
		D1	CTRDT1					
		D0	CTRDT0					
Watchdog timer count register 1 (pWD_CNT_HIGH)	0030066A (HW)	D15-14	–	reserved	–	–	–	0 when being read.
		D13	CTRDT29	Watchdog timer counter data CTRDT29 = MSB	0x0 to 0x3FFFFFFF (high-order 14 bits)	X	R	
		D12	CTRDT28					
		D11	CTRDT27					
		D10	CTRDT26					
		D9	CTRDT25					
		D8	CTRDT24					
		D7	CTRDT23					
		D6	CTRDT22					
		D5	CTRDT21					
		D4	CTRDT20					
		D3	CTRDT19					
		D2	CTRDT18					
		D1	CTRDT17					
		D0	CTRDT16					

The current count value of the up-counter can be read out from these registers.

D[15:0]/0x300668 CTRDT[15:0]: Watchdog Timer Count Data (16 low-order bits)

The 16 low-order bits of the 30-bit up-counter are read out from these bits. (Default: indeterminate)

D[13:0]/0x30066A CTRDT[29:16]: Watchdog Timer Count Data (14 high-order bits)

The 14 high-order bits of the 30-bit up-counter are read out from these bits. (Default: indeterminate)

0x30066C: Watchdog Timer Control Register (pWD_CNTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog timer control register (pWD_CNTL)	0030066C (HW)	D15-1	–	reserved	–	–	–	0 when being read.
		D0	WDRESEN	Watchdog timer reset	1 Reset 0 Invalid	0	W	

D[15:1] Reserved**D0 WDRESEN: Watchdog Timer Reset Bit**

This bit resets the watchdog timer.

1 (W): Reset

0 (W): Has no effect

0 (R): Always 0 when read (default)

With NMI or reset signal output enabled, the watchdog timer must be reset by writing 1 to this bit within the set NMI/reset generation cycle. The up-counter is thereby reset to 0, then starts counting NMI/reset generation cycles all over again.

IV.2.6 Precautions

- When NMI or reset signal output by the watchdog timer is enabled, the watchdog timer must be reset within the set NMI/reset generation cycle.
- Do not set a value equal to or less than 0x0000001F in the comparison data register.
- Depending on the counter and comparison register values, an NMI or reset signal may be generated after the NMI or reset function is enabled, or immediately after the watchdog timer starts. Always be sure to set comparison data and reset the watchdog timer before writing 1 to NMIEN (D1/0x300662), RESEN (D0/0x300662), or RUNSTP (D4/0x300662).
 - * **NMIEN**: Watchdog Timer NMI Enable Bit in the Watchdog Timer Enable Register (D1/0x300662)
 - * **RESEN**: Watchdog Timer RESET Enable Bit in the Watchdog Timer Enable Register (D0/0x300662)
 - * **RUNSTP**: Watchdog Timer Run/Stop Control Bit in the Watchdog Timer Enable Register (D4/0x300662)

S1C33E08 Technical Manual

V PERIPHERAL MODULES 3 (INTERFACE)

V.1 General-Purpose Serial Interface (EFSIO)

V.1.1 Configuration of Serial Interfaces

V.1.1.1 Features of Serial Interfaces

The S1C33E08 contains three channels (Ch.0, Ch.1 and Ch.2) of serial interfaces, the features of which are described below.

- A clock-synchronized, asynchronous, or ISO7816 mode can be selected for the transfer method.
 - Clock-synchronized mode (Ch.0, Ch.1 and Ch.2)
 - Data length: 8 bits, fixed (No start, stop, and parity bits)
 - Receive error: An overrun error can be detected.
 - Asynchronous mode (Ch.0, Ch.1 and Ch.2)
 - Data length: 7 or 8 bits, selectable
 - Receive error: Overrun, framing, or parity errors can be detected.
 - Start bit: 1 bit, fixed
 - Stop bit: 1 or 2 bits, selectable
 - Parity bit: Even, odd, or none, selectable
- Since the transmit and receive units are independent, full-duplex communication is possible. Supports IrDA interface. Internal clock or external clock is selectable.
- ISO7816 mode (Ch.1)
 - Data length: 8 bits, fixed (start, stop, and parity bits are not included)
 - Receive error: Overrun, framing, or parity errors can be detected.
 - Start bit: 1 bit, fixed
 - Stop bit: 2 bits (T = 0) or 1 bit (T = 1)
 - Parity bit: Even, fixed
 - Half-duplex communication using one data signal line
 - Transmit time guard function for low-speed serial device
- Baud-rate setting: Any desired baud rate can be set by selecting the baud-rate timer, or using external clock input (asynchronous mode only). Up to 8 Mbps transfer in clock-synchronized mode or up to 1 Mbps transfer in asynchronous mode are possible.
- 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 buffer empty, receive buffer full, and receive error) can be generated.

Figure V.1.1.1.1 shows the configuration of the serial interface (one channel).

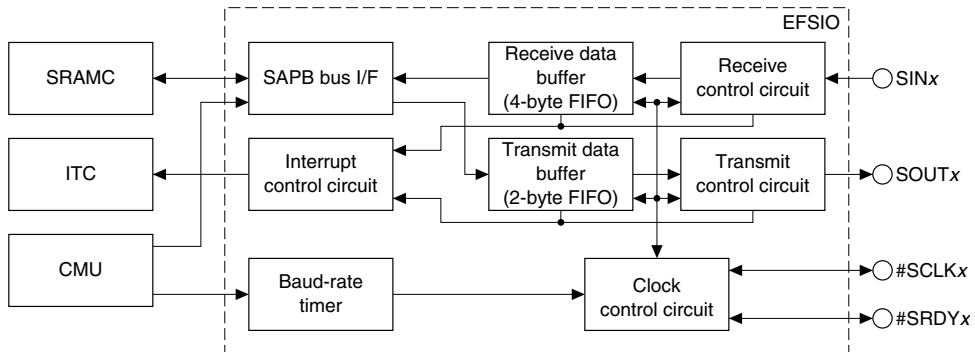


Figure V.1.1.1.1 Configuration of Serial Interface

Note: Ch.0 to Ch.2 have the same configuration and the same function (except for ISO7816 mode in Ch.1). The signal and control bit names are suffixed by a 0, 1, or 2 to indicate the channel number, enabling discrimination between channels 0 to 2. In this manual, however, channel numbers 0 to 2 are replaced with 'x' unless discrimination is necessary, because explanations are common to all three channels.

V.1.1.2 I/O Pins of Serial Interface

Table V.1.1.2.1 lists the I/O pins used by the serial interface.

Table V.1.1.2.1 Serial-Interface Pin Configuration

Pin name	I/O	Function
SIN0	I	Serial I/F Ch.0 data input
SOUT0	O	Serial I/F Ch.0 data output
#SCLK0	I/O	Serial I/F Ch.0 clock input/output
#SRDY0	I/O	Serial I/F Ch.0 ready input/output
SIN1	I	Serial I/F Ch.1 data input
SOUT1	O	Serial I/F Ch.1 data output
#SCLK1	I/O	Serial I/F Ch.1 clock input/output
#SRDY1	I/O	Serial I/F Ch.1 ready input/output
SIN2	I	Serial I/F Ch.2 data input
SOUT2	O	Serial I/F Ch.2 data output
#SCLK2	I/O	Serial I/F Ch.2 clock input/output
#SRDY2	I/O	Serial I/F Ch.2 ready input/output

SINx (serial-data input pin)

This pin is used to input serial data to the device.

SOUTx (serial-data output pin)

This pin is used to output serial data from the device.

#SCLKx (clock input/output pin)

This pin is used to input or output a clock.

In clock-synchronized slave mode, it is used as a clock input pin; in clock-synchronized master mode and ISO7816 mode (Ch.1), it is used as a clock output pin.

In 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 clock-synchronized mode.

In clock-synchronized slave mode, it is used as a ready-signal output pin; in clock-synchronized master mode, it is used as a ready-signal input pin.

This pin is not used in asynchronous mode and ISO7816 mode (Ch.1), so it can be used as an I/O port.

Note: The serial interface input/output pins are shared with general-purpose I/O ports or other peripheral circuit inputs/outputs, so that functionality in the initial state is set to other than the serial interface input/output. Before the serial interface input/output signals assigned to these pins can be used, the function of these pins must be switched for the serial interface input/output by setting the corresponding Port Function Select Registers.

For details of pin functions and how to switch over, see Section I.3.3, "Switching Over the Multiplexed Pin Functions."

V.1.1.3 Setting Interface Mode and Transfer Mode

The interface type and transfer mode of the serial interface can be set using SMDx[1:0] (D[1:0]/0x300Bx3), IRMDx[1:0] (D[1:0]/0x300Bx4), and 7816MD1[1:0] (D[1:0]/0x300B1A) individually for each channel as shown in Table V.1.1.3.1 below.

- * **SMDx[1:0]**: Serial I/F Ch.x Transfer Mode Select Bits in the Serial I/F Ch.x Control Register (D[1:0]/0x300Bx3)
- * **IRMDx[1:0]**: Serial I/F Ch.x Interface Mode Select Bits in the Serial I/F Ch.x IrDA Register (D[1:0]/0x300Bx4)
- * **7816MD1[1:0]**: Serial I/F Ch.1 ISO7816 Mode Select Bits in the Serial I/F Ch.1 ISO7816 Mode Control Register (D[1:0]/0x300B1A)

Table V.1.1.3.1 Mode Settings

7816MD1[1:0]	IRMDx[1:0]	Interface mode	SMDx[1:0]	Transfer mode	Support Ch.
10	00	ISO7816 mode (Ch.1 only)	11	ISO7816 (T = 1) mode, asynchronous	Ch.1
			00	ISO7816 (T = 1) mode, clock-sync	
01			ISO7816 (T = 0) mode, asynchronous		
00				ISO7816 (T = 0) mode, clock-sync	
00	10	IrDA mode	11	8-bit asynchronous mode (IrDA I/F)	Ch.0
	00	Normal mode	10	7-bit asynchronous mode (IrDA I/F)	Ch.1
			11	8-bit asynchronous mode (normal I/F)	Ch.2
			10	7-bit asynchronous mode (normal I/F)	
			01	Clock-synchronized slave mode	
00	Clock-synchronized master mode				
Other				Reserved	-

At initial reset, 7816MD1[1:0] (D[1:0]/0x300B1A) is set to 0b00, SMDx[1:0] (D[1:0]/0x300Bx3) and IRMDx[1:0] (D[1:0]/0x300Bx4) become 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 V.1.1.3.2.

Table V.1.1.3.2 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
ISO7816	Data input	Data output	Clock output	P port

All four pins are used in the clock-synchronized mode.

In the asynchronous mode, since #SRDYx is unused, the #SRDYx pin can be used as an I/O (P) port. In addition, when an external clock is not used, the #SCLKx pin can also be used as an I/O port.

The ISO7816 mode (Ch.1) does not use #SRDY1, so this pin can be used as an I/O (P) 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.

V.1.1.4 Serial Interface Operating Clock

The serial interface use the clocks generated by the CMU as the operating clock. Furthermore, each channel uses the data transfer clock generated by the baud-rate timer embedded in the EFSIO module.

Controlling the supply of the clock for accessing EFSIO

The SAPB bus interface clock (= MCLK) is supplied to the serial interface with default settings. It can be turned off using EFSIOSAPB_CKE (D5/0x301B04) to reduce the amount of power consumed on the chip if all the serial interface functions are not used.

* **EFSIOSAPB_CKE**: EFSIO SAPB I/F Clock Control Bit in the Gated Clock Control Register 1 (D5/0x301B04)

Setting EFSIOSAPB_CKE (D5/0x301B04) to 0 (1 by default) turns off the clock supply to the serial interface. When the clock supply is turned off, the serial interface control registers cannot be accessed.

For details on how to set and control clocks, see Section III.1, “Clock Management Unit (CMU).”

Clock for baud-rate timer and interface

The data transfer clock is generated by the baud-rate timer embedded in each channel. The baud-rate timer operating clock (= MCLK) is supplied separately with the SAPB bus interface clock shown above. Although this clock cannot be turned off while the S1C33E08 is running, this clock can be automatically turned off in HALT mode (see Section III.1.9.2) by setting EFSIOBR_HCKE (D25/0x301B04) to 0 (default: on).

* **EFSIOBR_HCKE**: EFSIO Baud Rate Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D25/0x301B04)

Refer to Section V.1.2 for the baud-rate timer.

Clock state in standby mode

The clock supply to the serial interface stops depending on type of standby mode.

HALT mode: The SAPB bus interface clock is supplied the same way as in normal mode (it can be stopped before entering HALT mode).

The baud-rate timer clock can be automatically stopped in HALT mode.

SLEEP mode: The SAPB bus interface clock and baud-rate timer clock stop.

Therefore, the serial interface also stops operating in SLEEP mode.

Note: The Gated Clock Control Register 1 (0x301B04) is write-protected. Write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless re-writing said registers.

V.1.1.5 Standard Mode and Advanced Mode

The serial interface in the S1C33E08 is extended from that of the C33 STD models. This serial interface has two operating modes, standard (STD) mode of which functions are compatible with the existing C33 STD models and an advanced (ADV) mode allowing use of the extended functions. Table V.1.1.5.1 shows differences between standard mode and advanced mode.

Table V.1.1.5.1 Differences between Standard Mode and Advanced Mode

Function	Standard mode	Advanced mode
#SRDY mask control	Disabled	Enabled
Number of received data in the buffer to generate a receive-buffer full interrupt	One	One to four can be specified.

To configure the serial interface in advanced mode, set SIOADV (D0/0x300B4F) to 1. The control bits for the extended functions are enabled to write after this setting. At initial reset, SIOADV (D0/0x300B4F) is set to 0 and the serial interface enters standard mode.

* **SIOADV**: Standard Mode/Advanced Mode Select Bit in the Serial I/F STD/ADV Mode Select Register (D0/0x300B4F)

The following descriptions unless otherwise specified are common contents for both modes. The extended functions in advanced mode are explained assuming that SIOADV (D0/0x300B4F) has been set to 1.

Note: Standard or advanced mode currently set is applied to all the serial interface channels. It cannot be selected for each channel individually.

V.1.2 Baud-Rate Timer (Setting Baud Rate)

The clock-synchronized master mode and ISO7816 mode use the internal clock for data transfer. Also in the asynchronous mode, the internal clock can be selected as the operating clock. Each channel has a dedicated baud-rate timer (12-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.

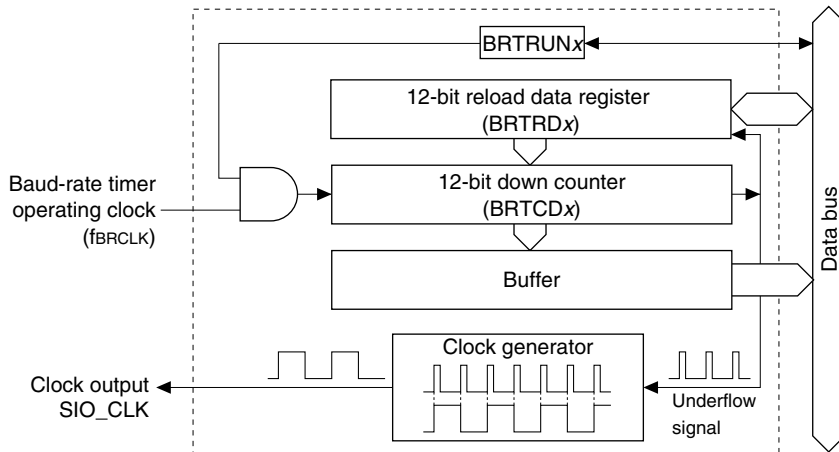


Figure V.1.2.1 Transfer Clock Generation by the Baud-Rate Timer

The baud-rate timer is configured with a 12-bit presettable down counter $BRTCDx[11:0]$ ($D[3:0]/0x300Bx9$, $D[7:0]/0x300Bx8$) and a 12-bit reload data register $BRTRDx[11:0]$ ($D[3:0]/0x300Bx7$, $D[7:0]/0x300Bx6$) for setting an initial value to the counter.

- * **$BRTCDx[11:8]$** : Serial I/F Ch.x Baud-rate Timer Counter Data [11:8] Bits in the Serial I/F Ch.x Baud-rate Timer Counter Data Register (MSB) ($D[3:0]/0x300Bx9$)
- $BRTCDx[7:0]$** : Serial I/F Ch.x Baud-rate Timer Counter Data [7:0] Bits in the Serial I/F Ch.x Baud-rate Timer Counter Data Register (LSB) ($D[7:0]/0x300Bx8$)
- * **$BRTRDx[11:8]$** : Serial I/F Ch.x Baud-rate Timer Reload Data [11:8] Bits in the Serial I/F Ch.x Baud-rate Timer Reload Data Register (MSB) ($D[3:0]/0x300Bx7$)
- $BRTRDx[7:0]$** : Serial I/F Ch.x Baud-rate Timer Reload Data [7:0] Bits in the Serial I/F Ch.x Baud-rate Timer Reload Data Register (LSB) ($D[7:0]/0x300Bx6$)

The baud-rate timer uses the MCLK clock supplied from the CMU as the count clock (BRCLK). For details on how to set and control the MCLK clock, see Section III.1, “Clock Management Unit (CMU).”

This clock can be automatically turned off in HALT mode (see Section V.1.1.4).

The following procedure generates the clock by the baud-rate timer.

1. Set an initial value to the reload data register $BRTRDx[11:0]$ ($D[3:0]/0x300Bx7$, $D[7:0]/0x300Bx6$).
2. Set $BRTRUNx$ ($D0/0x300Bx5$) to 1.

- * **$BRTRUNx$** : Serial I/F Ch.x Baud-rate Timer Run/Stop Control Bit in the Serial I/F Ch.x Baud-rate Timer Control Register ($D0/0x300Bx5$)

The baud-rate timer loads the initial value set in the reload data register to the counter when 1 is written to $BRTRUNx$ ($D0/0x300Bx5$), 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 $BRTRUNx$ 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 depends on the transfer mode.

Clock-synchronized master mode, ISO7816 mode (clock-synchronized)

$$\text{BRTRD} = \frac{f_{\text{BRCLK}}}{2 \times \text{bps}} - 1$$

BRTRD: Reload data register setup value of the baud-rate timer
 f_{BRCLK}: Baud-rate timer operating clock frequency (= MCLK Hz)
 bps: Transfer rate (bits/second)

Asynchronous mode

$$\text{BRTRD} = \frac{f_{\text{BRCLK}} \times \text{DIVMD}}{2 \times \text{bps}} - 1$$

BRTRD: Reload data register setup value of the baud-rate timer
 f_{BRCLK}: Baud-rate timer operating clock frequency (= MCLK Hz)
 bps: Transfer rate (bits/second)
 DIVMD: Internal division ratio of the serial interface (1/16 or 1/8 selected by DIVMD_x)

ISO7816 mode

$$\text{BRTRD} = \frac{f_{\text{BRCLK}}}{2 \times \text{bps}} \times \frac{D}{F} - 1$$

$$\text{BRTRD} = \frac{f_{\text{BRCLK}}}{2 \times \text{fSIO_CLK}} - 1$$

BRTRD: Reload data register setup value of the baud-rate timer
 f_{BRCLK}: Baud-rate timer operating clock frequency (= MCLK Hz)
 bps: Transfer rate (bits/second)
 D: Bit rate adjustment value (1, 2, 4, 8, 16, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64)
 F: Clock frequency divide value (372, 558, 744, 1116, 1488, 1860, 512, 768, 1024, 1536, 2048)
 fSIO_CLK: ISO7816 clock frequency (baud-rate timer output and #SCLK1 output)

V.1.3 Clock-Synchronized Interface

V.1.3.1 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 $SMDx[1:0]$ ($D[1:0]/0x300Bx3$).

* $SMDx[1:0]$: Serial I/F Ch.x Transfer Mode Select Bits in the Serial I/F Ch.x Control Register ($D[1:0]/0x300Bx3$)

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 V.1.3.1.1 shows an example of how the input/output pins are connected in the clock-synchronized mode.

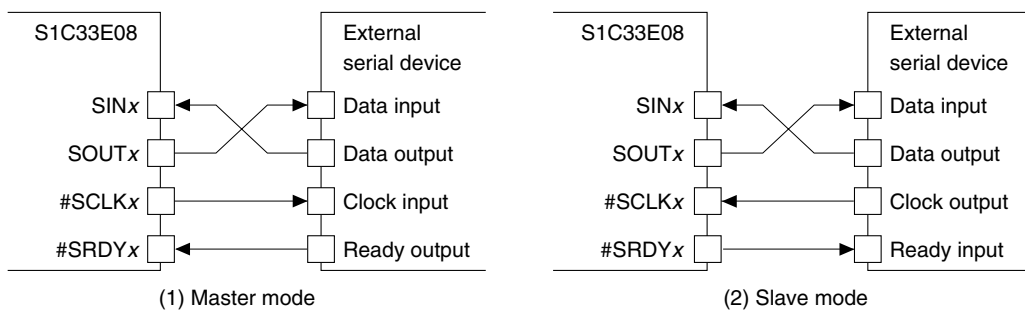


Figure V.1.3.1.1 Example of Connection in Clock-Synchronized Mode

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

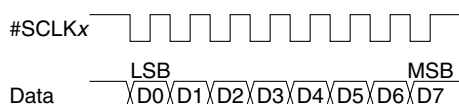


Figure V.1.3.1.2 Clock-Synchronized Transfer Data Format

Serial data is transmitted and received starting with the LSB.

V.1.3.2 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 clocks
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 Section V.1.7, “Serial Interface Interrupts and DMA.”

Note: Always make sure the serial interface is inactive (TXEN_x (D7/0x300Bx3) and RXEN_x (D6/0x300Bx3) = 0) before these settings are made. A change of settings during operation may cause a malfunction.

* **TXEN_x:** Serial I/F Ch.x Transmit Enable Bit in the Serial I/F Ch.x Control Register (D7/0x300Bx3)

* **RXEN_x:** Serial I/F Ch.x Receive Enable Bit in the Serial I/F Ch.x Control Register (D6/0x300Bx3)

Setting input/output pins

All four pins—SIN_x, SOUT_x, #SCLK_x, and #SRDY_x—are used in the clock-synchronized mode. Configure the Port Function Select Registers to enable these pin functions according to the channel to be used (two or more channel can be used simultaneously). For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Setting the interface mode

Write 0b00 to IRMD_x[1:0] (D[1:0]/0x300Bx4) to choose the ordinary interface. Since IRMD_x[1:0] (D[1:0]/0x300Bx4) becomes indeterminate at initial reset, it must be initialized.

* **IRMD_x[1:0]:** Serial I/F Ch.x Interface Mode Select Bits in the Serial I/F Ch.x IrDA Register (D[1:0]/0x300Bx4)

Also 7816MD1[1:0] (D[1:0]/0x300B1A) must be set to 0b00 in Ch.1.

* **7816MD1[1:0]:** Serial I/F Ch.1 ISO7816 Mode Select Bits in the Serial I/F Ch.1 ISO7816 Mode Control Register (D[1:0]/0x300B1A)

Setting the transfer mode

Use SMD_x[1:0] (D[1:0]/0x300Bx3) 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 SMD_x[1:0] to 0b00; when using the serial interface as a slave, set SMD_x[1:0] to 0b01.

* **SMD_x[1:0]:** Serial I/F Ch.x Transfer Mode Select Bits in the Serial I/F Ch.x Control Register (D[1:0]/0x300Bx3)

Setting the input clock

• Clock-synchronized master mode

This mode operates using the internal clock generated by the baud-rate timer. Setup the baud-rate timer according to the transfer rate for each channel. For how to control the baud-rate timer, see Section V.1.2, “Baud-Rate Timer (Setting Baud Rate).”

The serial-interface control register contains SSCK_x (D2/0x300Bx3) 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.

* **SSCK_x:** Serial I/F Ch.x Input Clock Select Bit in the Serial I/F Ch.x Control Register (D2/0x300Bx3)

• Clock-synchronized slave mode

This mode operates using the clock that is output by the external master. This clock is input from the #SCLK_x pin. Therefore, there is no need to control the baud-rate timer.

Initialize SSCK_x by writing 1 (#SCLK_x).

Setting the receive FIFO level (advanced mode)

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 FIFOINTx[1:0] (D[6:5]/0x300Bx4) to set this number of data. Writing 0–3 to FIFOINTx[1:0] (D[6:5]/0x300Bx4) 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.

- * **FIFOINTx[1:0]**: Serial I/F Ch.x Receive Buffer Full Interrupt Timing Select Bits in the Serial I/F Ch.x IrDA Register (D[6:5]/0x300Bx4)

V.1.3.3 Control and Operation of Clock-Synchronized Transfer

Transmit control

(1) Enabling transmit operation

Use the transmit-enable bit TXENx (D7/0x300Bx3) 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 #SCLKx pin is also enabled (ready for input/output).

Transmit is disabled and the transmit data buffer (FIFO) is cleared by writing 0 to TXENx (D7/0x300Bx3).

- * **TXENx**: Serial I/F Ch.x Transmit Enable Bit in the Serial I/F Ch.x Control Register (D7/0x300Bx3)

After the port function select register is set for the serial input/output, 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 (D7/0x300Bx3) and receive-enable bit RXENx (D6/0x300Bx3) cannot be enabled simultaneously. When transmitting data, fix RXENx (D6/0x300Bx3) at 0 and do not change it during a transmit operation.

In addition, make sure TXENx (D7/0x300Bx3) is not set to 0 during a transmit operation.

- * **RXENx**: Serial I/F Ch.x Receive Enable Bit in the Serial I/F Ch.x Control Register (D6/0x300Bx3)

(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 TXDx[7:0] (D[7:0]/0x300Bx0). The data written to TXDx[7:0] (D[7:0]/0x300Bx0) enters the transmit data buffer and waits for transmission.

- * **TXDx[7:0]**: Serial I/F Ch.x Transmit Data Bits in the Serial I/F Ch.x Transmit Data Register (D[7:0]/0x300Bx0)

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 TDBEx (D1/0x300Bx2) 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.

- * **TDBEx**: Serial I/F Ch.x Transmit Data Buffer Empty Flag in the Serial I/F Ch.x Status Register (D1/0x300Bx2)

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 TEND_x (D5/0x300Bx2). This flag goes 1 when data is being transmitted and goes 0 when the transmission has completed.

* **TEND_x**: Serial I/F Ch.*x* Transmit-Completion Flag in the Serial I/F Ch.*x* Status Register (D5/0x300Bx2)

When data is transmitted successively in clock-synchronized master mode, TEND_x (D5/0x300Bx2) maintains 1 until all data is transmitted (Figure V.1.3.3.1). In slave mode, TEND_x (D5/0x300Bx2) goes 0 every time 1-byte data is transmitted (Figure V.1.3.3.2).

When all the data in the transmit data buffer are transferred, a cause of the transmit-data empty interrupt 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 cause of interrupt 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 Section V.1.7, “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 #SRDY_x 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 #SRDY_x is on a low level.

Figure V.1.3.3.1 shows a transmit timing chart in the clock-synchronized master mode.

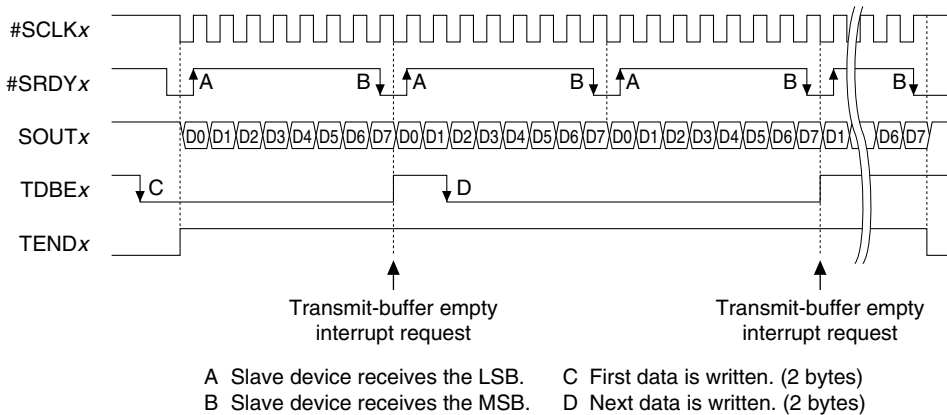


Figure V.1.3.3.1 Transmit Timing Chart in Clock-Synchronized Master Mode

1. If the #SRDY_x signal from the slave is on a high level, the master waits until it is on a low level (ready to receive).
2. If #SRDY_x is on a low level, the synchronizing clock input to the serial interface begins. The synchronizing clock is also output from the #SCLK_x 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 SOUT_x 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 SOUT_x. 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 V.1.3.3.2 shows a transmit timing chart in the clock-synchronized slave mode.

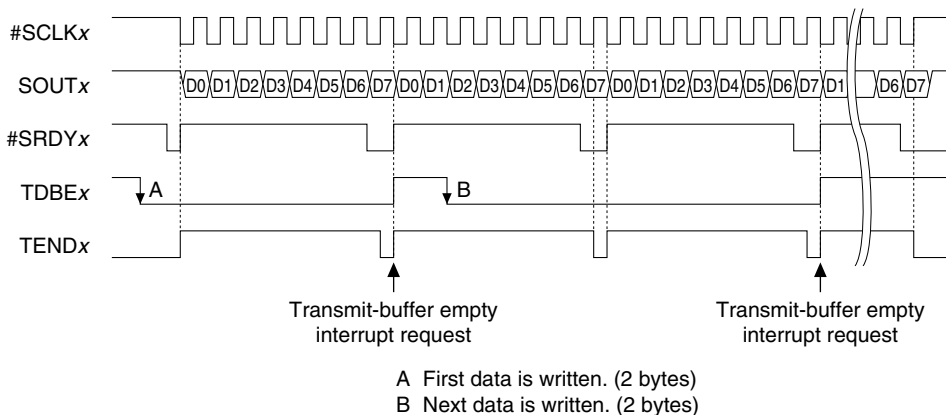


Figure V.1.3.3.2 Transmit Timing Chart in Clock-Synchronized Slave Mode

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. If the transmit data buffer becomes empty at this point, a transmit-buffer empty interrupt request occurs. 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 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 TXENx (D7/0x300Bx3) 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 TXENx (D7/0x300Bx3).

Receive control

(1) Enabling receive operation

Use the receive-enable bit $RXEN_x$ (D6/0x300Bx3) 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 $\#SCLK_x$ pin also is enabled (ready for input/output). Receive operations are disabled and the receive data buffer (FIFO) is cleared by writing 0 to $RXEN_x$ (D6/0x300Bx3).

* **$RXEN_x$** : Serial I/F Ch.x Receive Enable Bit in the Serial I/F Ch.x Control Register (D6/0x300Bx3)

After the port function select register is set for the serial input/output, the I/O direction of the $\#SRDY_x$ and $\#SCLK_x$ pins are changed at follows:

$\#SRDY_x$: When slave mode is set, a switch is made to output mode.

Otherwise, input mode is maintained.

$\#SCLK_x$: 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, $RXEN_x$ (D6/0x300Bx3) and transmit-enable bit $TXEN_x$ (D7/0x300Bx3) cannot be enabled simultaneously. When receiving data, fix $TXEN_x$ (D7/0x300Bx3) at 0 and do not change it during a receive operation. In addition, make sure $RXEN_x$ (D6/0x300Bx3) is not set to 0 during a receive operation.

* **$TXEN_x$** : Serial I/F Ch.x Transmit Enable Bit in the Serial I/F Ch.x Control Register (D7/0x300Bx3)

(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 $RXD_x[7:0]$ (D[7:0]/0x300Bx1). The older data is output first and cleared by reading.

* **$RXD_x[7:0]$** : Serial I/F Ch.x Receive Data Bits in the Serial I/F Ch.x Receive Data Register (D[7:0]/0x300Bx1)

The number of data in the receive data buffer can be checked by reading $RXD_xNUM[1:0]$ (D[7:6]/0x300Bx2). When $RXD_xNUM[1:0]$ (D[7:6]/0x300Bx2) is 0, the buffer contains 0 or 1 data. When $RXD_xNUM[1:0]$ (D[7:6]/0x300Bx2) is 1–3, the buffer contains 2–4 data.

* **$RXD_xNUM[1:0]$** : Number of Ch.x Receive Data in FIFO in the Serial I/F Ch.x Status Register (D[7:6]/0x300Bx2)

Furthermore, $RDBF_x$ (D0/0x300Bx2) is provided for indicating whether the receive 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.

* **$RDBF_x$** : Serial I/F Ch.x Receive Data Buffer Full Flag in the Serial I/F Ch.x Status Register (D0/0x300Bx2)

When the receive data buffer has received the specified number or more data (one in standard mode or one to four in advanced mode), a cause of the receive-buffer full interrupt 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 cause of interrupt 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 Section V.1.7, “Serial Interface Interrupts and DMA.”

The following describes a receive operation in the master and slave modes.

• **Clock-synchronized master mode**

Figure V.1.3.3.3 shows a receive timing chart in the clock-synchronized master mode.

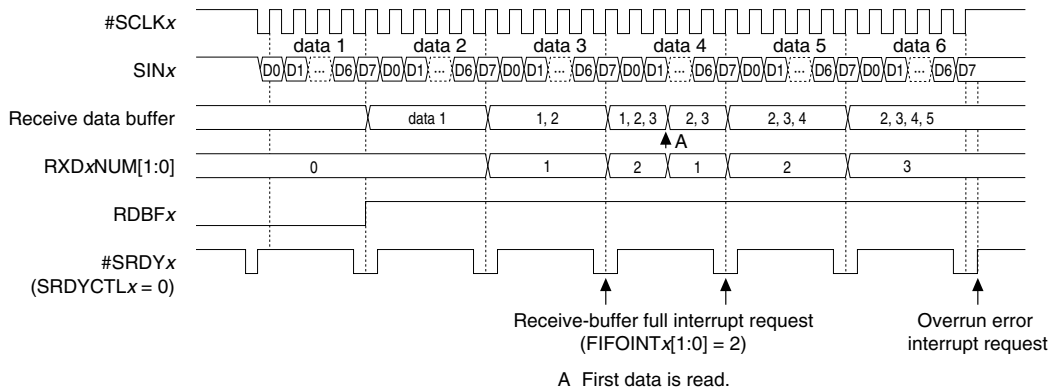


Figure V.1.3.3.3 Receive Timing Chart in Clock-Synchronized Master Mode

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 buffer, enabling the data to be read out.

• **Clock-synchronized slave mode**

Figure V.1.3.3.4 shows a receive timing chart in the clock-synchronized slave mode.

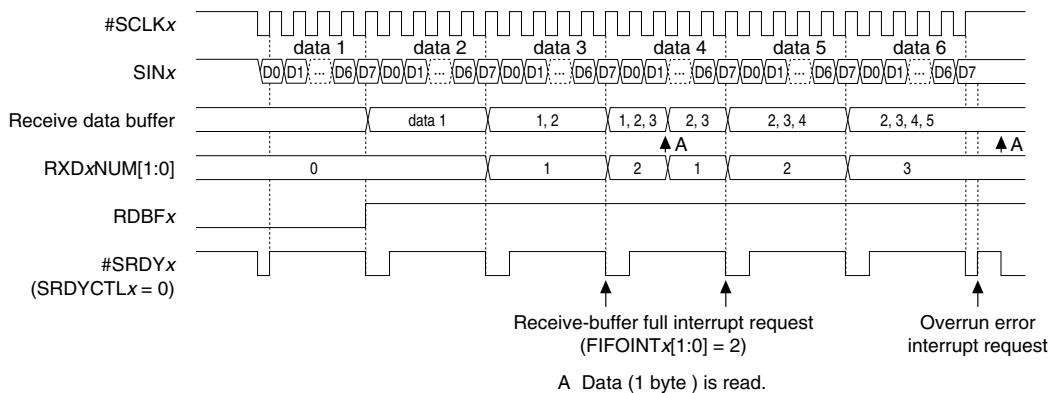


Figure V.1.3.3.4 Receive Timing Chart in Clock-Synchronized Slave Mode

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 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 OER_x (D2/0x300Bx2) 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.

* **OER_x**: Serial I/F Ch.x Overrun Error Flag in the Serial I/F Ch.x Status Register (D2/0x300Bx2)

The overrun error is one of the receive-error interrupt causes 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 $\#SRDY_x$ as shown below.

(4) Controlling the $\#SRDY_x$ signal (advanced mode)

When the slave device is in receive mode, the $\#SRDY_x$ 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 $\#SRDY_x$ signal is turned to a low level by writing 1 to $RXEN_x$ (D6/0x300Bx3) to enable receive operations, thereby indicating to the master device that the slave is ready to receive. When the LSB of data is received, $\#SRDY_x$ is turned to a high level; when the MSB is received, $\#SRDY_x$ is returned to a low level, in preparation for the next receive operation.

If an overrun error occurs, $\#SRDY_x$ is turned to a high level (unable to receive) at that point, so receive operations for the following data are suspended. In this case, $\#SRDY_x$ 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 $\#SRDY_x$ 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 provides $\#SRDY_x$ high mask mode. In this mode, if the receive data buffer is full, the $\#SRDY_x$ 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 $SRDYCTL_x$ (D7/0x300Bx4) to 1.

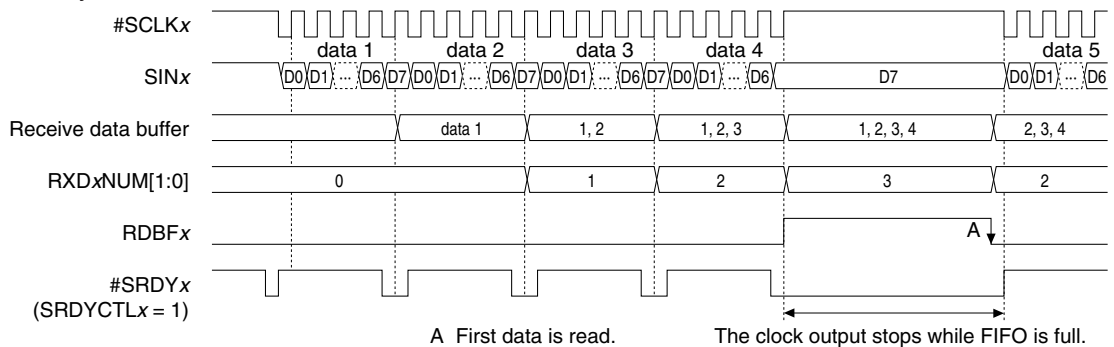
* **SRDYCTL_x**: Serial I/F Ch.x $\#SRDY$ Control Bit in the Serial I/F Ch.x IrDA Register (D7/0x300Bx4)

This function is effective in the clock-synchronized master mode as well. In this case, the $\#SRDY_x$ signal (low) from the slave device is ignored when the receive data buffer is full and the serial interface stops outputting the $\#SCLK_x$ 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.

V PERIPHERAL MODULES 3 (INTERFACE): GENERAL-PURPOSE SERIAL INTERFACE (EFSIO)

Clock-synchronized master mode



Clock-synchronized slave mode

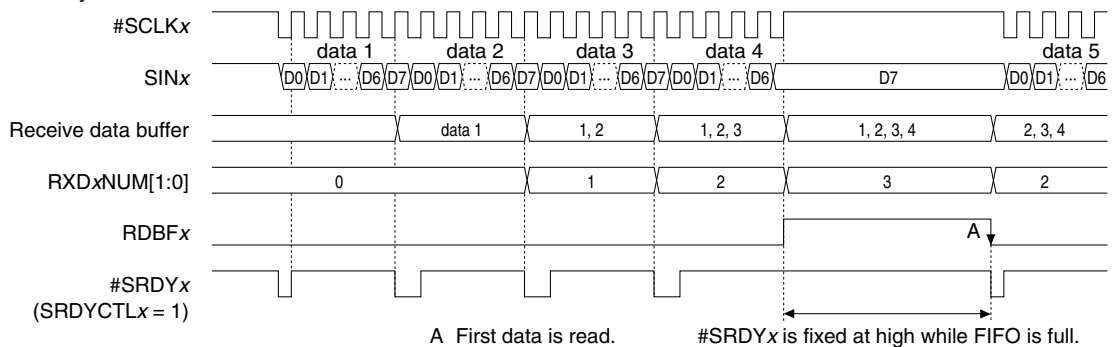


Figure V.1.3.3.5 #SRDYx High Mask Mode

(5) Terminating receive operation

Upon completion of a data receive operation, write 0 to the receive-enable bit $RXEN_x$ (D6/0x300B.x3) 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 $RXEN_x$ (D6/0x300B.x3) to 0.

V.1.4 Asynchronous Interface

V.1.4.1 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] = 0b11), 8 bits of data can be transferred; in the 7-bit asynchronous mode (SMDx[1:0] = 0b10), 7 bits of data can be transferred.

* **SMDx[1:0]**: Serial I/F Ch.x Transfer Mode Select Bits in the Serial I/F Ch.x Control Register (D[1:0]/0x300Bx3)

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 timer or an external clock that is input from the #SCLKx 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 V.1.4.1.1 shows an example of how input/output pins are connected for transfers in the asynchronous mode.

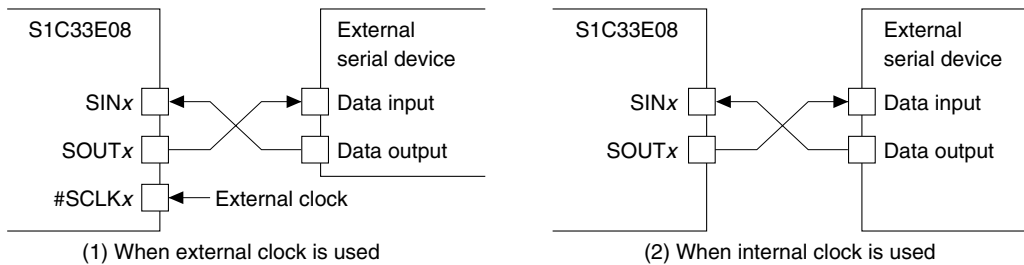


Figure V.1.4.1.1 Example of Connection in Asynchronous Mode

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

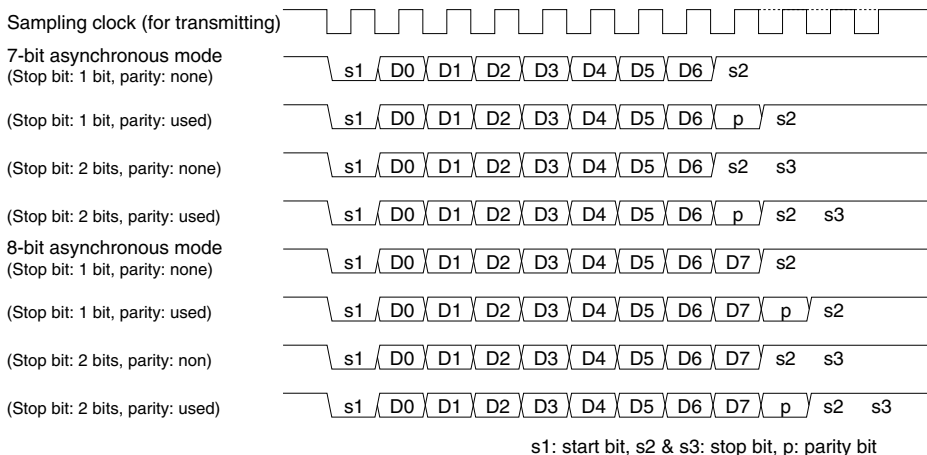


Figure V.1.4.1.2 Data Format for Asynchronous Transfer

Serial data is transmitted and received, starting with the LSB.

V.1.4.2 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 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 Section V.1.7, “Serial Interface Interrupts and DMA.”

Note: Always make sure the serial interface is inactive (TXEN_x (D7/0x300B_x3) and RXEN_x (D6/0x300B_x3) = 0) before making these settings. A change in settings during operation may result in a malfunction.

- * **TXEN_x**: Serial I/F Ch._x Transmit Enable Bit in the Serial I/F Ch._x Control Register (D7/0x300B_x3)
- * **RXEN_x**: Serial I/F Ch._x Receive Enable Bit in the Serial I/F Ch._x Control Register (D6/0x300B_x3)

Setting input/output pins

In the asynchronous mode, two pins—SIN_x and SOUT_x—are used. When external clock input is used, one more pin, #SCLK_x, is also used. Configure the Port Function Select Registers to enable these pin functions according to the channel to be used (two or more channel can be used simultaneously). For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Setting the interface mode

Initialize IRMD_x[1:0] (D[1:0]/0x300B_x4) by writing 0b00 when using the serial interface as a normal interface, or 0b10 when using the serial interface as an IrDA interface. This setting must be made before a transfer mode is set.

- * **IRMD_x[1:0]**: Serial I/F Ch._x Interface Mode Select Bits in the Serial I/F Ch._x IrDA Register (D[1:0]/0x300B_x4)

Also 7816MD1[1:0] (D[1:0]/0x300B1A) must be set to 0b00 in Ch.1.

- * **7816MD1[1:0]**: Serial I/F Ch.1 ISO7816 Mode Select Bits in the Serial I/F Ch.1 ISO7816 Mode Control Register (D[1:0]/0x300B1A)

Setting the transfer mode

Use SMD_x[1:0] (D[1:0]/0x300B_x3) to set the transfer mode of the serial interface as described earlier. When using the serial interface in the 8-bit asynchronous mode, set SMD_x[1:0] (D[1:0]/0x300B_x3) to 0b11, when using the serial interface in the 7-bit asynchronous mode, set SMD_x[1:0] (D[1:0]/0x300B_x3) to 0b10.

- * **SMD_x[1:0]**: Serial I/F Ch._x Transfer Mode Select Bits in the Serial I/F Ch._x Control Register (D[1:0]/0x300B_x3)

Setting the input clock

In the asynchronous mode, the operating clock can be selected between the internal clock and an external clock using SSCK_x (D2/0x300B_x3).

- * **SSCK_x**: Serial I/F Ch._x Input Clock Select Bit in the Serial I/F Ch._x Control Register (D2/0x300B_x3)

The external clock is selected (input from the #SCLK_x pin) by writing 1 to SSCK_x (D2/0x300B_x3), and an internal clock is selected by writing 0.

Note: SSCK_x (D2/0x300B_x3) 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 the clock generated using the baud-rate timer. Setup the baud-rate timer according to the transfer rate for each channel. For how to control the baud-rate timer, see Section V.1.2, “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 #SCLK_x pin. Therefore, there is no need to control the baud-rate timer.

Any desired clock frequency can be set. The clock input from the #SCLK_x 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, SIO_CLK (the clock output by the baud-rate timer or input from the #SCLK_x 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 DIVMD_x (D4/0x300Bx4), and a 1/8 ratio is selected by writing 1.

* **DIVMD_x**: Serial I/F Ch.x Clock Division Ratio Select Bit in the Serial I/F Ch.x IrDA Register (D4/0x300Bx4)

Note: DIVMD_x (D4/0x300Bx4) 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

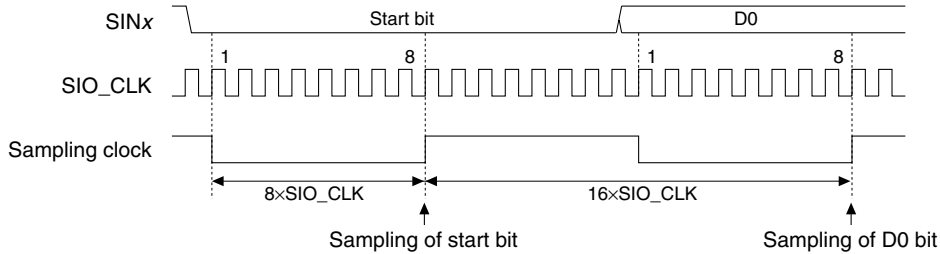


Figure V.1.4.2.1 Sampling Clock for Asynchronous Receive Operation (when 1/16 division is selected)

Each bit data is sampled in the timing shown in Figure V.1.4.2.1. When the SIN_x input signal is detected as a low level at the rising edge of SIO_CLK, sampling for the start bit is performed 8×SIO_CLK (4×SIO_CLK 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 SIN_x input signal is low at the start bit sampling, subsequent bit data is sampled in 16×SIO_CLK cycles (8×SIO_CLK cycles when 1/8 division is selected).

For transmitting

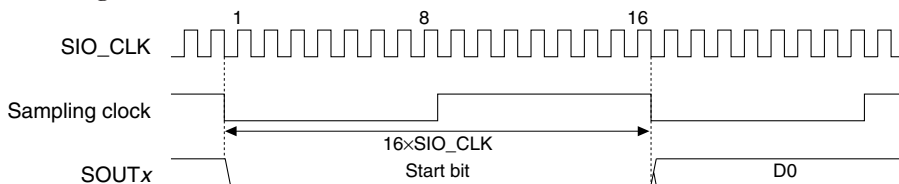


Figure V.1.4.2.2 Sampling Clock for Asynchronous Transmit Operation (when 1/16 division is selected)

During transmission, each bit data is output from the SOUT_x pin in 16×SIO_CLK cycles (8×SIO_CLK 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 V.1.4.2.2 using the control bits listed below.

Stop bit select

* **STPBx**: Serial I/F Ch.x Stop-Bit Length Select Bit in the Serial I/F Ch.x Control Register (D3/0x300Bx3)

Parity enable

* **EPRx**: Serial I/F Ch.x Parity Enable Bit in the Serial I/F Ch.x Control Register (D5/0x300Bx3)

Parity mode select

* **PMDx**: Serial I/F Ch.x Parity Mode Select Bit in the Serial I/F Ch.x Control Register (D4/0x300Bx3)

Table V.1.4.2.1 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.

Setting the receive FIFO level (advanced mode)

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 FIFOINTx[1:0] (D[6:5]/0x300Bx4) to set this number of data. Writing 0–3 to FIFOINTx[1:0] (D[6:5]/0x300Bx4) 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.

* **FIFOINTx[1:0]**: Serial I/F Ch.x Receive Buffer Full Interrupt Timing Select Bits in the Serial I/F Ch.x IrDA Register (D[6:5]/0x300Bx4)

V.1.4.3 Control and Operation of Asynchronous Transfer

Transmit control

(1) Enabling transmit operation

Use the transmit-enable bit TXENx (D7/0x300Bx3) 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 TXENx (D7/0x300Bx3).

* **TXENx**: Serial I/F Ch.x Transmit Enable Bit in the Serial I/F Ch.x Control Register (D7/0x300Bx3)

Note: Do not set TXENx (D7/0x300Bx3) 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 TXDx[7:0] (D[7:0]/0x300Bx0).

* **TXDx[7:0]**: Serial I/F Ch.x Transmit Data Bits in the Serial I/F Ch.x Transmit Data Register (D[7:0]/0x300Bx0)

In the 7-bit asynchronous mode, bit 7 (MSB) in each register is ignored.

The data written to TXD_x[7:0] (D[7:0]/0x300B_x0) 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 TDBE_x (D1/0x300B_x2) 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.

* **TDBE_x**: Serial I/F Ch.*x* Transmit Data Buffer Empty Flag in the Serial I/F Ch.*x* Status Register (D1/0x300B_x2)

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 TEND_x (D5/0x300B_x2). This flag goes 1 when data is being transmitted and goes 0 when the transmission has completed.

* **TEND_x**: Serial I/F Ch.*x* Transmit-Completion Flag in the Serial I/F Ch.*x* Status Register (D5/0x300B_x2)

When all the data in the transmit data buffer are transferred, a cause of the transmit-data empty interrupt 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 cause of interrupt 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 Section V.1.7, “Serial Interface Interrupts and DMA.”

Figure V.1.4.3.1 shows a transmit timing chart in the asynchronous mode.

Example: Data length: 8 bits, Stop bit: 1 bit, Parity bit: Included

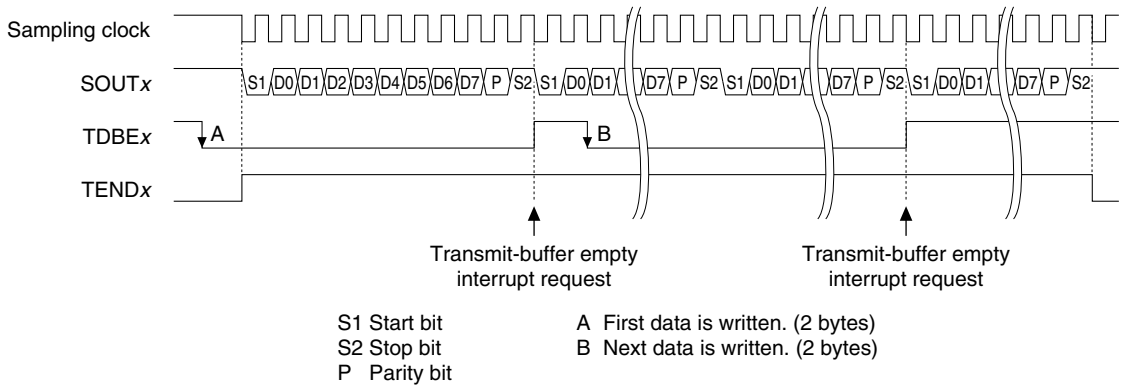


Figure V.1.4.3.1 Transmit Timing Chart in Asynchronous Mode

1. The contents of the buffer are transferred to the shift register synchronously with the first falling edge of the sampling clock. At the same time, the SOUT_x 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 EPR_x = 1) and the stop bit are transmitted in succession.

* **EPR_x**: Serial I/F Ch.*x* Parity Enable Bit in the Serial I/F Ch.*x* Control Register (D5/0x300B_x3)

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 TXEN_x (D7/0x300B_x3) 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 TXEN_x (D7/0x300B_x3).

Receive control

(1) Enabling receive operations

Use the receive-enable bit $RXEN_x$ (D6/0x300Bx3) 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 $RXEN_x$ (D6/0x300Bx3).

* **$RXEN_x$** : Serial I/F Ch.x Receive Enable Bit in the Serial I/F Ch.x Control Register (D6/0x300Bx3)

Note: Do not set $RXEN_x$ (D6/0x300Bx3) 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 $RXD_x[7:0]$ (D[7:0]/0x300Bx1). The older data is output first and cleared by reading.

* **$RXD_x[7:0]$** : Serial I/F Ch.x Receive Data Bits in the Serial I/F Ch.x Receive Data Register (D[7:0]/0x300Bx1)

The number of data in the receive data buffer can be checked by reading $RXD_xNUM[1:0]$ (D[7:6]/0x300Bx2). When $RXD_xNUM[1:0]$ (D[7:6]/0x300Bx2) is 0, the buffer contains 0 or 1 data. When $RXD_xNUM[1:0]$ (D[7:6]/0x300Bx2) is 1–3, the buffer contains 2–4 data.

* **$RXD_xNUM[1:0]$** : Number of Ch.x Receive Data in FIFO in the Serial I/F Ch.x Status Register (D[7:6]/0x300Bx2)

Furthermore, $RDBF_x$ (D0/0x300Bx2) is provided for indicating whether the receive 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.

* **$RDBF_x$** : Serial I/F Ch.x Receive Data Buffer Full Flag in the Serial I/F Ch.x Status Register (D0/0x300Bx2)

When the receive data buffer has received the specified number or more data (one in standard mode or one to four in advanced mode), a cause of the receive-buffer full interrupt 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 cause of interrupt 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 Section V.1.7, “Serial Interface Interrupts and DMA.”

Figure V.1.4.3.2 shows a receive timing chart in the asynchronous mode.

Example: Data length: 8 bits, Stop bit: 1 bit, Parity bit: Included

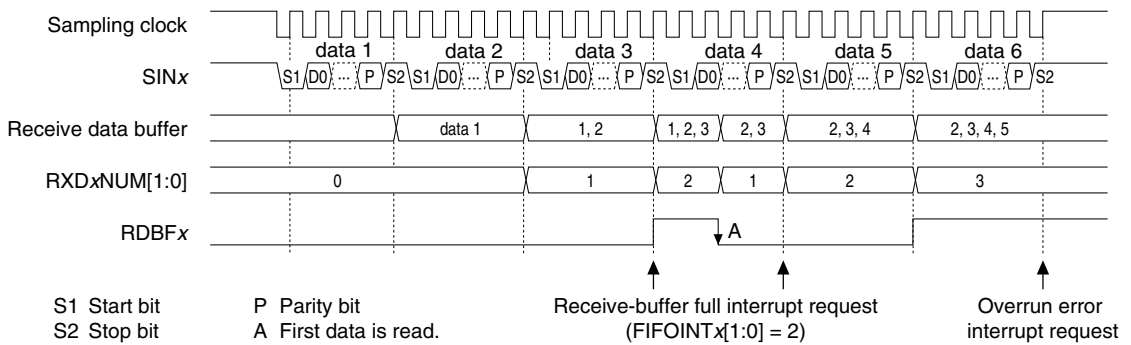


Figure V.1.4.3.2 Receive Timing Chart in Asynchronous Mode

1. The serial interface starts sampling when the start bit is input (SIN_x = 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 EPR_x = 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 EPR_x = 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 Section V.1.7, “Serial Interface Interrupts and DMA.”

• Parity error

If EPR_x (D5/0x300Bx3) 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 buffer in order to check conformity with PMD_x (D4/0x300Bx3) settings (odd or even parity).

* **PMD_x:** Serial I/F Ch._x Parity Mode Select Bit in the Serial I/F Ch._x Control Register (D4/0x300Bx3)

If any nonconformity is found in this check, a parity error is assumed and the parity error flag PER_x (D3/0x300Bx2) is set to 1.

* **PER_x:** Serial I/F Ch._x Parity Error Flag in the Serial I/F Ch._x Status Register (D3/0x300Bx2)

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.

PER_x (D3/0x300Bx2) 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 FER_x (D4/0x300B_{x2}) is set to 1.

* **FER_x**: Serial I/F Ch._x Framing Error Flag in the Serial I/F Ch._x Status Register (D4/0x300B_{x2})

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 framing error is flagged cannot be guaranteed, even if no framing error is found in the following data received.

The FER_x (D4/0x300B_{x2}) 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 OER_x (D2/0x300B_{x2}) is set to 1.

* **OER_x**: Serial I/F Ch._x Overrun Error Flag in the Serial I/F Ch._x Status Register (D2/0x300B_{x2})

Even when this error occurs, the receive operation is continued.

OER_x (D2/0x300B_{x2}) 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 RXEN_x (D6/0x300B_{x3}) 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 RXEN_x (D6/0x300B_{x3}) to 0.

V.1.5 IrDA Interface

V.1.5.1 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.

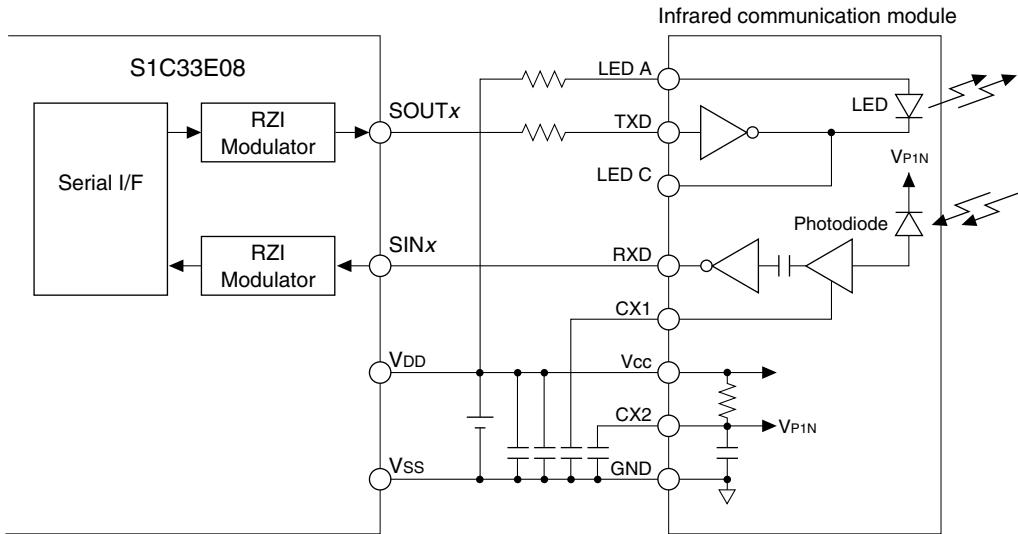


Figure V.1.5.1.1 Configuration Example of IrDA Interface

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 Section V.1.4, “Asynchronous Interface,” for details on how to set and control the data formats and data transfers.

V.1.5.2 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 Section V.1.4, “Asynchronous Interface.” For details on item 7, refer to Section V.1.7, “Serial Interface Interrupts and DMA.”

Note: Before making these settings, always make sure the serial interface is inactive (TXENx (D7/0x300Bx3) and RXENx (D6/0x300Bx3) 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 after selecting the IrDA interface function in (2).

- * **TXENx:** Serial I/F Ch.x Transmit Enable Bit in the Serial I/F Ch.x Control Register (D7/0x300Bx3)
- * **RXENx:** Serial I/F Ch.x Receive Enable Bit in the Serial I/F Ch.x Control Register (D6/0x300Bx3)

Selecting the IrDA interface function

To use the IrDA interface function, select it using IRMDx[1:0] (D[1:0]/0x300Bx4) and then set the 8-bit (or 7-bit) asynchronous mode as the transfer mode.

Table V.1.5.2.1 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

* IRMDx[1:0]: Serial I/F Ch.x Interface Mode Select Bits in the Serial I/F Ch.x IrDA Register (D[1:0]/0x300Bx4)

Note: IRMDx[1:0] (D[1:0]/0x300Bx4) 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 IRRLx (D2/0x300Bx4) and IRTLx (D3/0x300Bx4), respectively.

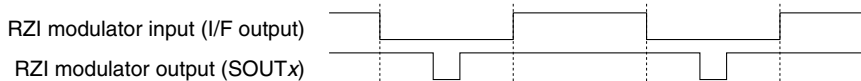
* IRRLx: Serial I/F Ch.x IrDA I/F Input Logic Inversion Bit in the Serial I/F Ch.x IrDA Register (D2/0x300Bx4)

* IRTLx: Serial I/F Ch.x IrDA I/F Output Logic Inversion Bit in the Serial I/F Ch.x IrDA Register (D3/0x300Bx4)

The logic of the input/output signal is inverted by writing 1 to IRRLx (D2/0x300Bx4)/IRTLx (D3/0x300Bx4). Logic is not inverted if the bit is set to 0.

When transmitting

(1) IRTLx = 0

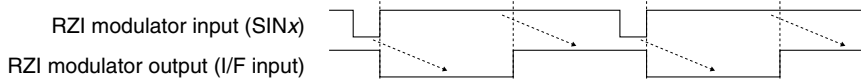


(2) IRTLx = 1



When receiving

(1) IRRLx = 0



(2) IRRLx = 1

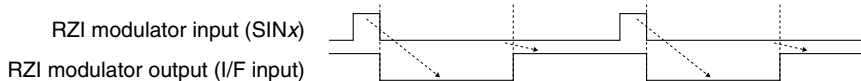


Figure V.1.5.2.1 IRRLx and IRTLx Settings

Note: IRRLx (D2/0x300Bx4) and IRTLx (D3/0x300Bx4) become indeterminate at initial reset, so be sure to initialize them in the software.

V.1.5.3 Control and Operation of IrDA Interface

The transmit/receive procedures have been explained in the section on the asynchronous interface, so refer to Section V.1.4.3, “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 SOUT_x pin.

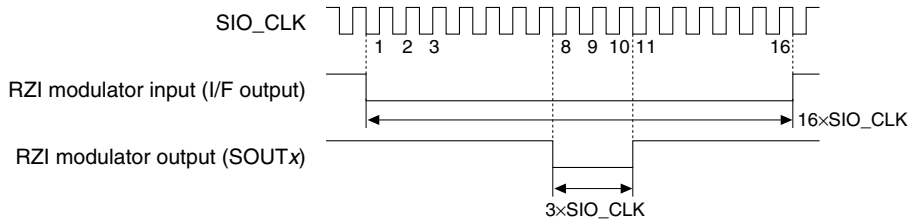


Figure V.1.5.3.1 Data Modulation by RZI Circuit

When receiving

During data reception, the pulse width of the input signal from SIN_x is set to 16/3 before the signal is transferred to the serial interface.

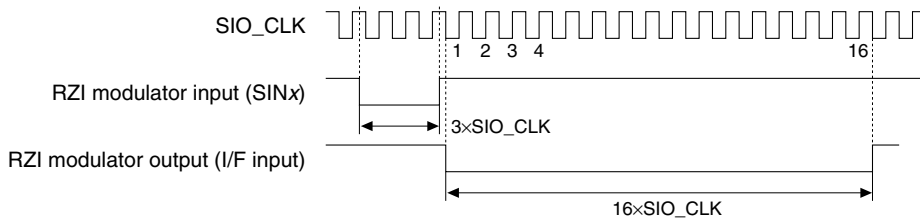


Figure V.1.5.3.2 Demodulation by RZI Circuit

Notes:

- When using the IrDA interface, set the internal division ratio of the serial interface 1/16 (DIVMD_x = 1), rather than 1/8 (DIVMD_x = 0).

* **DIVMD_x**: Serial I/F Ch.*x* Clock Division Ratio Select Bit in the Serial I/F Ch.*x* IrDA Register (D4/0x300Bx4)

- Although Figure V.1.5.3.2 shows the input signal as a low pulse of a 3×SIO_CLK width, the RZI circuit recognizes low pulses by means of the signal edge (rising edge when IRRL_x = 0; falling edge when IRRL_x = 1). Note that noise may cause a malfunction.

V.1.6 ISO7816 Interface (Ch.1)

Note: ISO7816 mode is available only for Ch.1.

V.1.6.1 Outline of ISO7816 Interface

S1C33E08 supports a Smart Card interface in conformity with ISO7816-3.

The ISO7816 interface performs serial communication using two wires for data transfer and clock.

The transfer method is an asynchronous system that transfers data with start and stop bits. This interface allows clock synchronized transfer using the same data format.

Since the data line is shared between the transmitter and receiver, the communication mode is half-duplex.

This interface supports T = 0 and T = 1 protocols, and it can be selected using 7816MD1[1:0] (D[1:0]/0x300B1A).

* **7816MD1[1:0]:** Serial I/F Ch.1 ISO7816 Mode Select Bits in the Serial I/F Ch.1 ISO7816 Mode Control Register (D[1:0]/0x300B1A)

Table V.1.6.1.1 Selecting ISO7816 Mode

7816MD11	7816MD10	Mode
1	1	Reserved
1	0	ISO7816 (T = 1) mode
0	1	ISO7816 (T = 0) mode
0	0	Normal interface

(Default: 0b00)

Figure V.1.6.1.1 shows a Smart Card connection example.

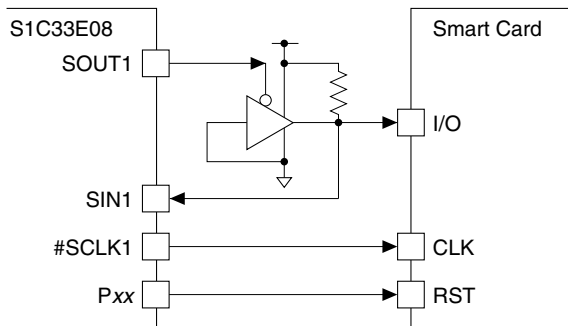


Figure V.1.6.1.1 Connection Example in ISO7816 Mode

ISO7816 transfer data format

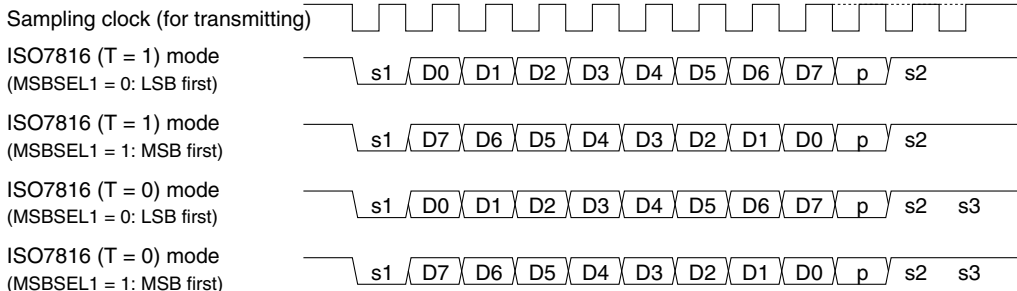
The data format for ISO7816 mode is shown below.

Data length: 8 bits, fixed

Start bit: 1 bit, fixed

Stop bit: 1 bit (T = 1) or 2 bits (T = 0)

Parity bit: Even parity



s1: start bit, s2 & s3: stop bit, p: parity bit

Figure V.1.6.1.2 ISO7816 Transfer Data Format

V.1.6.2 Setting ISO7816 Interface

When performing transfer in ISO7816 mode, 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 retransmit count for error recovery and time guard function
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 Section V.1.7, “Serial Interface Interrupts and DMA.”

Note: Always make sure the serial interface is inactive (TXEN1 (D7/0x300B13) and RXEN1 (D6/0x300B13) = 0) before making these settings. A change in settings during operation may result in a malfunction.

* **TXEN1:** Serial I/F Ch.1 Transmit Enable Bit in the Serial I/F Ch.1 Control Register (D7/0x300B13)

* **RXEN1:** Serial I/F Ch.1 Receive Enable Bit in the Serial I/F Ch.1 Control Register (D6/0x300B13)

Setting input/output pins

In ISO7816 mode, three pins—SIN1, SOUT1, and #SCLK1—are used. Configure the Port Function Select Registers to enable these pin functions. For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Setting the interface mode

Set 7816MD1[1:0] (D[1:0]/0x300B1A) to 0b10 when T = 1 protocol is used or 0b01 when T = 0 protocol is used.

* **7816MD1[1:0]:** Serial I/F Ch.1 ISO7816 Mode Select Bits in the Serial I/F Ch.1 ISO7816 Mode Control Register (D[1:0]/0x300B1A)

Setting the transfer mode

Use SMD1[1:0] (D[1:0]/0x300B13) to set the transfer mode of the serial interface as described earlier. When performing asynchronous transfer, set SMD1[1:0] to 0b11, when performing clock synchronized transfer, set SMD1[1:0] to 0b00.

* **SMD1[1:0]:** Serial I/F Ch.1 Transfer Mode Select Bits in the Serial I/F Ch.1 Control Register (D[1:0]/0x300B13)

Setting the input clock

This mode operates using the internal clock generated by the baud-rate timer. Setup the Ch.1 baud-rate timer according to the transfer rate. For how to control the baud-rate timer, see Section V.1.2, “Baud-Rate Timer (Setting Baud Rate).”

Asynchronous transfer

The transfer rate in ISO7816 mode is expressed by the following equation:

$$\text{bps} = \frac{D}{F} \times \text{fsio_clk}$$

bps: Bit rate (bits/second)

D: Bit rate adjustment value
1, 2, 4, 8, 16, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64

F: Clock frequency divide value
372 (5 MHz), 558 (6 MHz), 744 (8 MHz), 1116 (12 MHz), 1488 (16 MHz), 1860 (20 MHz),
512 (5 MHz), 768 (7.5 MHz), 1024 (10 MHz), 1536 (15 MHz), 2048 (20 MHz)
() indicates the maximum output clock frequency.

fsio_clk: ISO7816 clock frequency (baud-rate timer output, which is also output from the #SCLK1 pin to clock a card)

V PERIPHERAL MODULES 3 (INTERFACE): GENERAL-PURPOSE SERIAL INTERFACE (EFSIO)

The sampling clock frequency for asynchronous transfer is expressed by the following equation:

$$f_{\text{SAMPL}} = \frac{D}{F} \times f_{\text{SIO_CLK}} \times \frac{1}{\text{DIVMD}}$$

f_{SAMPL} : Sampling clock frequency

DIVMD: Divide ratio internally used by the serial interface (1/16 or 1/8, selected with DIVMD1)

Use FIDI1[13:0] (D[5:0]/0x300B1D, D[7:0]/0x300B1C) and DIVMD1 (D4/0x300B14) to set up the sampling clock.

* **FIDI1[13:0]**: Serial I/F Ch.1 ISO7816 Mode FI/DI Ratio Setup Bits in the Serial I/F Ch.1 ISO7816 Mode FI/DI Ratio Registers (D[5:0]/0x300B1D, D[7:0]/0x300B1C)

* **DIVMD1**: Serial I/F Ch.1 Clock Division Ratio Select Bit in the Serial I/F Ch.1 IrDA Register (D4/0x300B14)

DIVMD is set to 1/16 when 0 is written to DIVMD1 (D4/0x300B14) or 1/8 when 1 is written.

FIDI1[13:0] (D[5:0]/0x300B1D, D[7:0]/0x300B1C) should be set to $F \times \text{DIVMD} / D - 1$. Tables V.1.6.2.1 and V.1.6.2.2 list the values that can be set to FIDI1[13:0] (D[5:0]/0x300B1D, D[7:0]/0x300B1C).

Table V.1.6.2.1 FIDI1[13:0] Set Values (DIVMD = 1/8)

F/(D×8) -1	D											
	1	2	4	8	16	1/2	1/4	1/8	1/16	1/32	1/64	
F	372	46	22	11	5	2	92	185	371	743	1487	2975
	558	69	34	16	8	3	139	278	557	1115	2231	4463
	744	92	46	22	11	5	185	371	743	1487	2975	5951
	1116	139	69	34	16	8	278	557	1115	2231	4463	8927
	1488	185	92	46	22	11	371	743	1487	2975	5951	11903
	1860	232	115	57	28	14	464	929	1859	3719	7439	14879
	512	63	31	15	7	3	127	255	511	1023	2047	4095
	768	95	47	23	11	5	191	383	767	1535	3071	6143
	1024	127	63	31	15	7	255	511	1023	2047	4095	8191
	1536	191	95	47	23	11	383	767	1535	3071	6143	12287
2048	255	127	63	31	15	511	1023	2047	4095	8191	16383	

Table V.1.6.2.2 FIDI1[13:0] Set Values (DIVMD = 1/16)

F/(D×16) -1	D											
	1	2	4	8	16	1/2	1/4	1/8	1/16	1/32	1/64	
F	372	22	11	5	2	0	46	92	185	371	743	1487
	558	34	16	8	3	1	69	139	278	557	1115	2231
	744	46	22	11	5	2	92	185	371	743	1487	2975
	1116	69	34	16	8	3	139	278	557	1115	2231	4463
	1488	92	46	22	11	5	185	371	743	1487	2975	5951
	1860	115	57	28	14	6	232	464	929	1859	3719	7439
	512	31	15	7	3	1	63	127	255	511	1023	2047
	768	47	23	11	5	2	95	191	383	767	1535	3071
	1024	63	31	15	7	3	127	255	511	1023	2047	4095
	1536	95	47	23	11	5	191	383	767	1535	3071	6143
2048	127	63	31	15	7	255	511	1023	2047	4095	8191	

For receiving

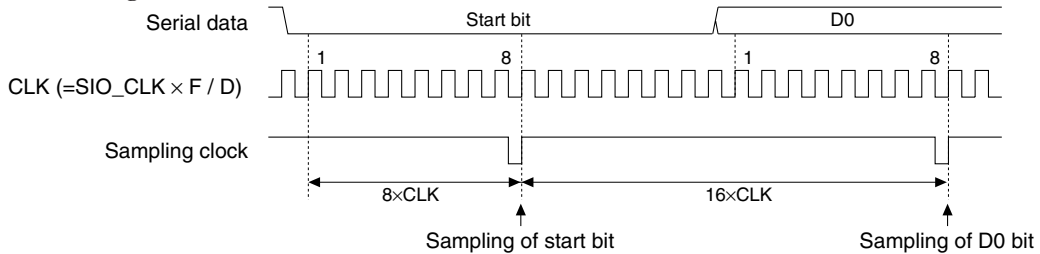


Figure V.1.6.2.1 Sampling Clock for Asynchronous Receive Operation (when 1/16 division is selected)

Each bit data is sampled in the timing shown in Figure V.1.6.2.1.

When the SIN1 input signal is detected as a low level at the rising edge of CLK, sampling for the start bit is performed $8 \times \text{CLK}$ ($4 \times \text{CLK}$ 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 SIN1 input signal is low at the start bit sampling, subsequent bit data is sampled in $16 \times \text{CLK}$ cycles ($8 \times \text{CLK}$ cycles when 1/8 division is selected).

For transmitting

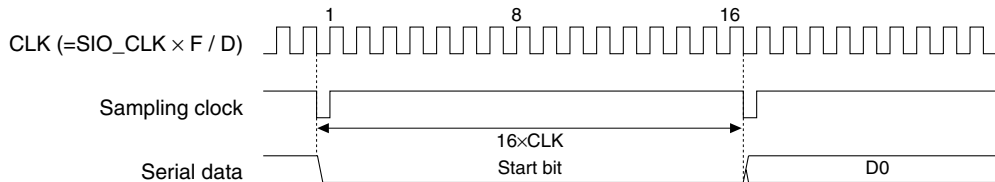


Figure V.1.6.2.2 Sampling Clock for Asynchronous Transmit Operation (when 1/16 division is selected)

During transmission, each bit data is output from the SOUT1 pin in $16 \times \text{CLK}$ cycles ($8 \times \text{CLK}$ cycles when 1/8 division is selected).

Clock synchronized transfer

The transfer rate in ISO7816 mode clock synchronized transfer can be calculated similar to the asynchronous transfer. Note, however, that DIVMD equals 1 (not 1/16 or 1/8) in clock synchronized transfer regardless of how DIVMD1 is set.

Setting the retransmit count for error recovery (T = 0 protocol)

The T = 0 protocol allows retransmission of data when an error occurs in data transmission (when the receiver returns NACK). Retransmission can be repeated if the error occurs successively, and RPNUM1[2:0] (D[7:5]/0x300B1A) is used to set the retransmit count. A maximum of seven retransmissions may be specified. When RPNUM1[2:0] (D[7:5]/0x300B1A) is set to 0, this interface does not retransmit data even if a transmit error occurs.

When a transmit error occurs in T = 1 protocol mode, this interface does not retransmit data regardless of how RPNUM1[2:0] (D[7:5]/0x300B1A) is set.

- * **RPNUM1[2:0]**: Serial I/F Ch.1 Number of Transmit Repetition Setup Bits in the Serial I/F Ch.1 ISO7816 Mode Control Register (D[7:5]/0x300B1A)

Setting the time guard function

The ISO7816 mode supports a time guard function that inserts an idle time between characters during transmission. The idle time to be inserted can be specified in ETU (bit cycle) units using TTGR1[7:0] (D[7:0]/0x300B1E). When TTGR1[7:0] (D[7:0]/0x300B1E) is set to 0, no idle time is inserted. When a value other than 0 is set, the SOUT1 output is fixed at high for the specified ETU period after a stop bit is output. This high output period is regarded as a long stop bit.

- * **TTGR1[7:0]**: Serial I/F Ch.1 Transmit Time Guard Setup Bits in the Serial I/F Ch.1 Transmit Time Guard Register (D[7:0]/0x300B1E)

Setting the receive FIFO level (advanced mode)

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 FIFOINT1[1:0] (D[6:5]/0x300B14) to set this number of data. Writing 0–3 to FIFOINT1[1:0] (D[6:5]/0x300B14) 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.

- * **FIFOINT1[1:0]**: Serial I/F Ch.1 Receive Buffer Full Interrupt Timing Select Bits in the Serial I/F Ch.1 IrDA Register (D[6:5]/0x300B14)

V.1.6.3 Control and Operation of ISO7816 Mode

Transmit control

(1) Clock output

First, start clock output in the following procedure:

1. Set CLKOL1 (D3/0x300B1A) to 1 (forced low output is released).
2. Set CLKOEN1 (D4/0x300B1A) to 1 (clock output begins).

- * **CLKOL1**: Serial I/F Ch.1 Clock Output Forced Low Bit in the Serial I/F Ch.1 ISO7816 Mode Control Register (D3/0x300B1A)
- * **CLKOEN1**: Serial I/F Ch.1 Clock Output Enable Bit in the Serial I/F Ch.1 ISO7816 Mode Control Register (D4/0x300B1A)

The clock for asynchronous transfer is output in synchronization with the sampling clock.

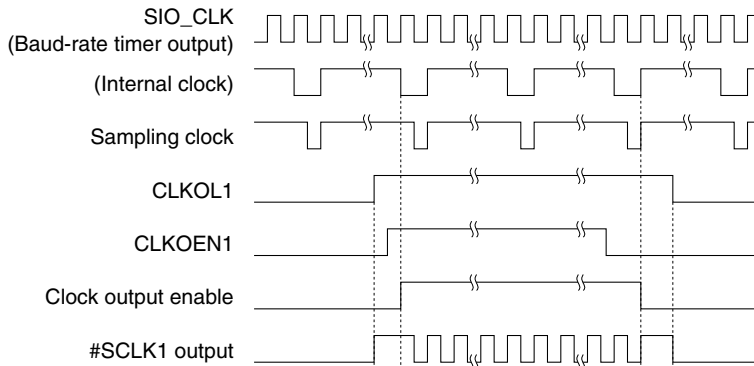


Figure V.1.6.3.1 Clock Output Control (example in asynchronous mode)

(2) Enabling transmit operation

Use the transmit-enable bit TXEN1 (D7/0x300B13) 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 TXEN1 (D7/0x300B13).

- * **TXEN1**: Serial I/F Ch.1 Transmit Enable Bit in the Serial I/F Ch.1 Control Register (D7/0x300B13)

Note: Do not set TXEN1 (D7/0x300B13) to 0 during a transmit operation.

(3) 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 TXD1[7:0] (D[7:0]/0x300B10).

- * **TXD1[7:0]**: Serial I/F Ch.1 Transmit Data Bits in the Serial I/F Ch.1 Transmit Data Register (D[7:0]/0x300B10)

The data written to TXD1[7:0] (D[7:0]/0x300B10) 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 TDBE1 (D1/0x300B12) 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.

- * **TDBE1**: Serial I/F Ch.1 Transmit Data Buffer Empty Flag in the Serial I/F Ch.1 Status Register (D1/0x300B12)

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 TEND1 (D5/0x300B12). This flag goes 1 when data is being transmitted and goes 0 when the transmission has completed.

* **TEND1**: Serial I/F Ch.1 Transmit-Completion Flag in the Serial I/F Ch.1 Status Register (D5/0x300B12)

When all the data in the transmit data buffer are transferred, a cause of the transmit-data empty interrupt 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 cause of interrupt 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 Section V.1.7, “Serial Interface Interrupts and DMA.”

Figures V.1.6.3.2 to V.1.6.3.4 show transmit timing charts in ISO7816 mode.

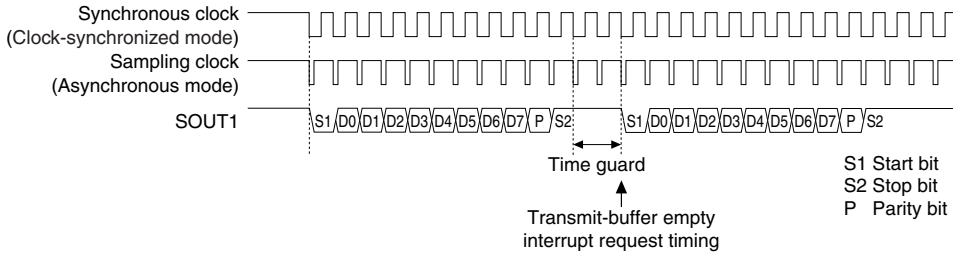


Figure V.1.6.3.2 Transmit Timing Chart in ISO7816 (T = 1) Mode (LSB first, time guard = 2)

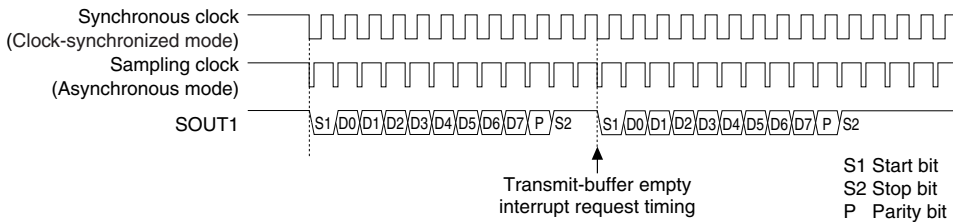


Figure V.1.6.3.3 Transmit Timing Chart in ISO7816 (T = 0) Mode (LSB first, time guard = 0, no parity error occurred)

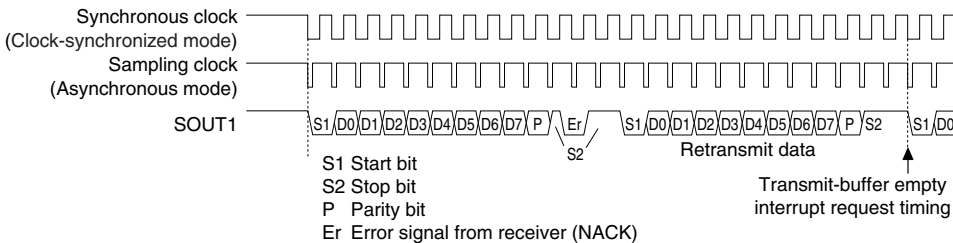


Figure V.1.6.3.4 Transmit Timing Chart in ISO7816 (T = 0) Mode (LSB first, time guard = 0, parity error occurred)

1. The data line (SOUT1) in idle state is set into high-impedance (pulled up to high).
2. The contents of the data buffer are transferred to the shift register synchronously with the first falling edge of the clock. At the same time, the SOUT1 pin is setting to a low level to send the start bit.
3. Each bit of data in the shift register is transmitted at each falling edge of the subsequent clock.
4. After sending the 8th data bit, the parity bit and the stop bit are transmitted in succession. Then SOUT1 is set into high-impedance state (pulled up to high).
5. The interface idles for a time guard period after sending a stop bit if the time guard period is set.
6. The next data transfer begins if the transmit data buffer contains other data.
7. If a parity error occurs in the receiver when data is being transferred in T = 0 mode, the receiver returns a low-level error signal (NACK). The interface transmits the same data again when the retransmit count for error recovery has been set.

In T = 1 mode, the receiver does not return an error signal even if an error occurs in the receiver.

(4) Terminating transmit operations

When data transmission is completed, write 0 to the transmit-enable bit TXEN1 (D7/0x300B13) 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 TXEN1 (D7/0x300B13).

To disable clock output, first reset CLKOEN1 (D4/0x300B1A) to 0 and then CLKOL1 (D3/0x300B1A) to 0.

Receive control

(1) Clock output

Start clock output in the following procedure if it is disabled:

1. Set CLKOL1 (D3/0x300B1A) to 1 (forced low output is released).
2. Set CLKOEN1 (D4/0x300B1A) to 1 (clock output begins).

* **CLKOL1**: Serial I/F Ch.1 Clock Output Forced Low Bit in the Serial I/F Ch.1 ISO7816 Mode Control Register (D3/0x300B1A)

* **CLKOEN1**: Serial I/F Ch.1 Clock Output Enable Bit in the Serial I/F Ch.1 ISO7816 Mode Control Register (D4/0x300B1A)

(2) Enabling receive operations

Use the receive-enable bit RXEN1 (D6/0x300B13) 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 RXEN1 (D6/0x300B13).

* **RXEN1**: Serial I/F Ch.1 Receive Enable Bit in the Serial I/F Ch.1 Control Register (D6/0x300B13)

Note: Do not set RXEN1 (D6/0x300B13) to 0 during a receive operation.

(3) 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 RXD1[7:0] (D[7:0]/0x300B11). The older data is output first and cleared by reading.

* **RXD1[7:0]**: Serial I/F Ch.1 Receive Data Bits in the Serial I/F Ch.1 Receive Data Register (D[7:0]/0x300B11)

The number of data in the receive data buffer can be checked by reading RXD1NUM[1:0] (D[7:6]/0x300B12). When RXD1NUM[1:0] (D[7:6]/0x300B12) is 0, the buffer contains 0 or 1 data. When RXD1NUM[1:0] (D[7:6]/0x300B12) is 1–3, the buffer contains 2–4 data.

* **RXD1NUM[1:0]**: Number of Ch.1 Receive Data in FIFO in the Serial I/F Ch.1 Status Register (D[7:6]/0x300B12)

Furthermore, RDBF1 (D0/0x300B12) is provided for indicating whether the receive 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.

* **RDBF1**: Serial I/F Ch.1 Receive Data Buffer Full Flag in the Serial I/F Ch.1 Status Register (D0/0x300B12)

When the receive data buffer has received the specified number or more data (one in standard mode or one to four in advanced mode), a cause of the receive-buffer full interrupt 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 cause of interrupt 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 Section V.1.7, “Serial Interface Interrupts and DMA.”

Figures V.1.6.3.5 to V.1.6.3.7 show receive timing charts in ISO7816 mode.

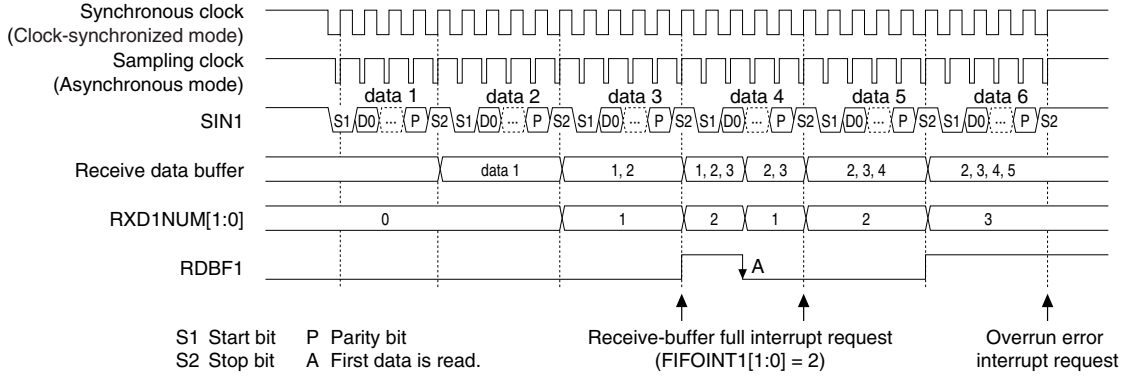


Figure V.1.6.3.5 Receive Timing Chart in ISO7816 (T = 1) Mode (LSB first)

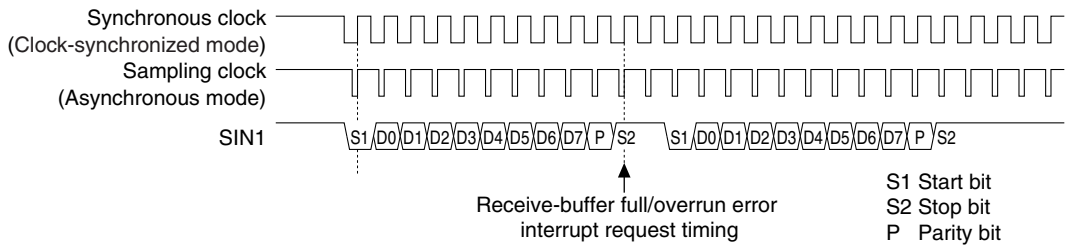


Figure V.1.6.3.6 Receive Timing Chart in ISO7816 (T = 0) Mode (LSB first, no parity error occurred)

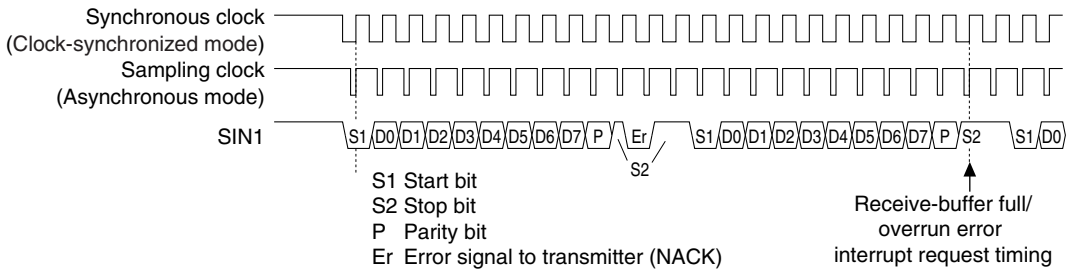


Figure V.1.6.3.7 Receive Timing Chart in ISO7816 (T = 0) Mode (LSB first, parity error occurred)

1. The serial interface in asynchronous mode starts sampling when the start bit is input (SIN1 = low). The serial interface in clock synchronized mode starts sampling at the first rising edge of the synchronous clock.
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 at each rising edge of the subsequent clock. This operation is repeated until the 8th data is received.
3. When the 8th data bit is taken in, the parity bit that follows is also taken in.
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.
5. If a parity error occurs in T = 0 mode, the interface returns a low-level error signal (NACK) to the transmitter.
In T = 1 mode, the interface does not return an error signal even if a parity error occurs.

(4) Receive errors

Three types of receive errors can be detected when receiving data in ISO7816 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 Section V.1.7, “Serial Interface Interrupts and DMA.”

- **Parity error**

In ISO7816 mode, 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 buffer.

In T = 1 mode, if any nonconformity between the received data and parity bit is found in this check, a parity error is assumed and the parity error flag PER1 (D3/0x300B12) is set to 1.

* **PER1**: Serial I/F Ch.1 Parity Error Flag in the Serial I/F Ch.1 Status Register (D3/0x300B12)

Even when this error occurs, the received data in error is transferred to the receive data buffer and the receive operation is continued in T=1 mode. However, the content of the received data for which a parity error is flagged cannot be guaranteed.

In T = 0 mode, the received data in error is not loaded to the receive data buffer and a parity error cannot be detected. The parity error flag PER1 (D3/0x300B12) will not be set to 1.

PER1 (D3/0x300B12) is reset to 0 by writing 0.

As described above, if a parity error occurs in T = 0 mode, the interface returns a low-level error signal (NACK) to the transmitter (see Figure V.1.6.3.7).

In T = 1 mode, the interface does not return an error signal even if a parity error occurs.

- **Framing error (asynchronous mode)**

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 FER1 (D4/0x300B12) is set to 1.

* **FER1**: Serial I/F Ch.1 Framing Error Flag in the Serial I/F Ch.1 Status Register (D4/0x300B12)

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 framing error is flagged cannot be guaranteed, even if no framing error is found in the following data received.

FER1 (D4/0x300B12) 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 OER1 (D2/0x300B12) is set to 1.

* **OER1**: Serial I/F Ch.1 Overrun Error Flag in the Serial I/F Ch.1 Status Register (D2/0x300B12)

Even when this error occurs, the receive operation is continued.

OER1 (D2/0x300B12) is reset to 0 by writing 0.

(5) Terminating receive operation

When a data receive operation is completed, write 0 to the receive-enable bit RXEN1 (D6/0x300B13) 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 RXEN1 (D6/0x300B13) to 0.

To disable clock output, first reset CLKOEN1 (D4/0x300B1A) to 0 and then CLKOL1 (D3/0x300B1A) to 0.

V.1.7 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

This cause of interrupt occurs when the transmit data set in the transmit data register is transferred to the shift register, in which case the cause-of-interrupt flag FSTX_x 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 cause of interrupt indicates that the next transmit data can be written to the transmit data register. This cause of interrupt 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 cause of interrupt occurs when the number of data specified with FIFOINT_x[1:0] (D[6:5]/0x300B_{x4}) (one data in standard mode) has been received in the receive data buffer, in which case the cause-of-interrupt flag FSRX_x 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 cause of interrupt indicates that the received data can be read out. This cause of interrupt can also be used to invoke IDMA, enabling the received data to be written into specified memory locations by means of a DMA transfer.

- * **FIFOINT_x[1:0]**: Serial I/F Ch._x Receive Buffer Full Interrupt Timing Select Bits in the Serial I/F Ch._x IrDA Register (D[6:5]/0x300B_{x4})

Receive-error interrupt

This cause of interrupt occurs when a parity, framing, or overrun error is detected during data reception, or when a transmit error is detected during data transmission in ISO7816 T = 0 mode, in which case the cause-of-interrupt flag FSERR_x 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 four types of errors generate the same cause of interrupt, check the error flags PER_x (parity error), OER_x (overrun error), FER_x (framing error), and TER1 (transmit error flag) to identify the type of error that has occurred. In the clock-synchronized mode, parity and framing errors do not occur.

- * **PER_x**: Serial I/F Ch._x Parity Error Flag in the Serial I/F Ch._x Status Register (D3/0x300B_{x2})
- * **OER_x**: Serial I/F Ch._x Overrun Error Flag in the Serial I/F Ch._x Status Register (D2/0x300B_{x2})
- * **FER_x**: Serial I/F Ch._x Framing Error Flag in the Serial I/F Ch._x Status Register (D4/0x300B_{x2})
- * **TER1**: Serial I/F Ch.1 ISO7816 Transmit Error Flag in the Serial I/F Ch.1 ISO7816 Mode Status Register (D0/0x300B1B)

Note: If a receive error (parity or framing error) occurs, the receive-error interrupt and receive-buffer full interrupt causes 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 cause flag be cleared through the use of the receive-error interrupt processing routine.

Control registers of the interrupt controller

Table V.1.7.1 shows the interrupt controller's control registers provided for each interrupt source (channel).

Table V.1.7.1 Control Register of Interrupt Controller

Channel	Cause of interrupt	Cause-of-interrupt flag	Interrupt enable register	Interrupt priority register
Ch.0	Receive-error	FSERR0(D0/0x300286)	ESERR0(D0/0x300276)	PSIO0[2:0](D[6:4]/0x300269)
	Receive-buffer full	FSRX0(D1/0x300286)	ESRX0(D1/0x300276)	
	Transmit-buffer empty	FSTX0(D2/0x300286)	ESTX0(D2/0x300276)	
Ch.1	Receive-error interrupt	FSERR1(D3/0x300286)	ESERR1(D3/0x300276)	PSIO1[2:0](D[2:0]/0x30026A)
	Receive-buffer full	FSRX1(D4/0x300286)	ESRX1(D4/0x300276)	
	Transmit-buffer empty	FSTX1(D5/0x300286)	ESTX1(D5/0x300276)	
Ch.2	Receive-error	FSERR2(D0/0x300289)	ESERR2(D0/0x300279)	PSIO2[2:0](D[2:0]/0x30026E)
	Receive-buffer full	FSRX2(D1/0x300289)	ESRX2(D1/0x300279)	
	Transmit-buffer empty	FSTX2(D2/0x300289)	ESTX2(D2/0x300279)	

When a cause of interrupt described above occurs, the corresponding cause-of-interrupt flag is set to 1. If the interrupt enable register bit for that cause of interrupt has been set to 1, an interrupt request is generated.

Interrupts can be disabled by leaving the interrupt enable register bit for that cause of interrupt set to 0. The cause-of-interrupt 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 Section III.2, "Interrupt Controller (ITC)."

Intelligent DMA

The receive-buffer full interrupt and transmit-buffer empty interrupt causes 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 DMA transfer.

The following shows the IDMA channel numbers set for each cause of interrupt:

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

The IDMA request and enable bits shown in Table V.1.7.2 must be set to 1 for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

Table V.1.7.2 Control Bits for IDMA Transfer

Channel	Cause of interrupt	IDMA request bit	IDMA enable bit
Ch.0	Receive-buffer full	RSRX0(D6/0x300292)	DESRX0(D6/0x300296)
	Transmit-buffer empty	RSTX0(D7/0x300292)	DESTX0(D7/0x300296)
Ch.1	Receive-buffer full	RSRX1(D0/0x300293)	DESRX1(D0/0x300297)
	Transmit-buffer empty	RSTX1(D1/0x300293)	DESTX1(D1/0x300297)
Ch.2	Receive-buffer full	RSRX2(D2/0x30029B)	DESRX2(D2/0x30029C)
	Transmit-buffer empty	RSTX2(D3/0x30029B)	DESTX2(D3/0x30029C)

If a cause of interrupt 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 DMA transfer performed.

For details on DMA transfer and how to control interrupts upon completion of DMA transfer, refer to Section II.2, "Intelligent DMA (IDMA)."

High-speed DMA

The receive-buffer full interrupt and transmit-buffer empty interrupt causes can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit corresponding to each channel:

Table V.1.7.3 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 (0x300298)
1	1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0–1 trigger set-up register (0x300298)
2	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2–3 trigger set-up register (0x300299)

For HSDMA to be invoked by a cause of receive-buffer full interrupt, the trigger set-up bits should be set to “1010.” For HSDMA to be invoked by a cause of transmit-buffer empty interrupt, 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 cause of interrupt.

For details on HSDMA transfer, refer to Section II.1, “High-Speed DMA (HSDMA).”

Trap vectors

The trap-vector address of each default cause of interrupt is set as follows:

Ch.0 receive-error interrupt: 0xC000E0
 Ch.0 receive-buffer full interrupt: 0xC000E4
 Ch.0 transmit-buffer empty interrupt: 0xC000E8
 Ch.1 receive-error interrupt: 0xC000F0
 Ch.1 receive-buffer full interrupt: 0xC000F4
 Ch.1 transmit-buffer empty interrupt: 0xC000F8
 Ch.2 receive-error interrupt: 0xC00130
 Ch.2 receive-buffer full interrupt: 0xC00134
 Ch.2 transmit-buffer empty interrupt: 0xC00138

The base address of the trap table can be changed using the TTBR register.

V.1.8 Details of Control Registers

Table V.1.8.1 List of Serial Interface Registers

Address	Register name	Size	Function
0x00300B00	Serial I/F Ch.0 Transmit Data Register (pEFSIF0_TXD)	8	Ch.0 transmit data
0x00300B01	Serial I/F Ch.0 Receive Data Register (pEFSIF0_RXD)	8	Ch.0 receive data
0x00300B02	Serial I/F Ch.0 Status Register (pEFSIF0_STATUS)	8	Ch.0 transfer/error status
0x00300B03	Serial I/F Ch.0 Control Register (pEFSIF0_CTL)	8	Sets Ch.0 transfer mode and controls transfer.
0x00300B04	Serial I/F Ch.0 IrDA Register (pEFSIF0_IRDA)	8	Sets Ch.0 asynchronous/IrDA mode.
0x00300B05	Serial I/F Ch.0 Baud-rate Timer Control Register (pEFSIF0_BRTRUN)	8	Controls Ch.0 baud-rate timer.
0x00300B06	Serial I/F Ch.0 Baud-rate Timer Reload Data Register (LSB) (pEFSIF0_BRTRDL)	8	Ch.0 baud-rate timer reload data low-order 8 bits
0x00300B07	Serial I/F Ch.0 Baud-rate Timer Reload Data Register (MSB) (pEFSIF0_BRTRDM)	8	Ch.0 baud-rate timer reload data high-order 4 bits
0x00300B08	Serial I/F Ch.0 Baud-rate Timer Count Data Register (LSB) (pEFSIF0_BRTCDL)	8	Ch.0 baud-rate timer count data low-order 8 bits
0x00300B09	Serial I/F Ch.0 Baud-rate Timer Count Data Register (MSB) (pEFSIF0_BRTCDM)	8	Ch.0 baud-rate timer count data high-order 4 bits
0x00300B10	Serial I/F Ch.1 Transmit Data Register (pEFSIF1_TXD)	8	Ch.1 transmit data
0x00300B11	Serial I/F Ch.1 Receive Data Register (pEFSIF1_RXD)	8	Ch.1 receive data
0x00300B12	Serial I/F Ch.1 Status Register (pEFSIF1_STATUS)	8	Ch.1 transfer/error status
0x00300B13	Serial I/F Ch.1 Control Register (pEFSIF1_CTL)	8	Sets Ch.1 transfer mode and controls transfer.
0x00300B14	Serial I/F Ch.1 IrDA Register (pEFSIF1_IRDA)	8	Sets Ch.1 asynchronous/IrDA mode.
0x00300B15	Serial I/F Ch.1 Baud-rate Timer Control Register (pEFSIF1_BRTRUN)	8	Controls Ch.1 baud-rate timer.
0x00300B16	Serial I/F Ch.1 Baud-rate Timer Reload Data Register (LSB) (pEFSIF1_BRTRDL)	8	Ch.1 baud-rate timer reload data low-order 8 bits
0x00300B17	Serial I/F Ch.1 Baud-rate Timer Reload Data Register (MSB) (pEFSIF1_BRTRDM)	8	Ch.1 baud-rate timer reload data high-order 4 bits
0x00300B18	Serial I/F Ch.1 Baud-rate Timer Count Data Register (LSB) (pEFSIF1_BRTCDL)	8	Ch.1 baud-rate timer count data low-order 8 bits
0x00300B19	Serial I/F Ch.1 Baud-rate Timer Count Data Register (MSB) (pEFSIF1_BRTCDM)	8	Ch.1 baud-rate timer count data high-order 4 bits
0x00300B1A	Serial I/F Ch.1 ISO7816 Mode Control Register (pEFSIF1_7816CTL)	8	Sets Ch.1 ISO7816 mode and controls clock output.
0x00300B1B	Serial I/F Ch.1 ISO7816 Mode Status Register (pEFSIF1_7816STA)	8	Ch.1 ISO7816 error status
0x00300B1C	Serial I/F Ch.1 ISO7816 Mode FI/DI Ratio Register (LSB) (pEFSIF1_FIDIL)	8	Ch.1 ISO7816 FI/DI ratio low-order 8 bits
0x00300B1D	Serial I/F Ch.1 ISO7816 Mode FI/DI Ratio Register (MSB) (pEFSIF1_FIDIM)	8	Ch.1 ISO7816 FI/DI ratio high-order 6 bits
0x00300B1E	Serial I/F Ch.1 Transmit Time Guard Register (pEFSIF1_TTGR)	8	Sets Ch.1 transmit time guard function.
0x00300B1F	Serial I/F Ch.1 ISO7816 Mode Output Clock Setup Register (pEFSIF1_CLKNUM)	8	Sets number of output clocks for Ch.1 ISO7816 mode.
0x00300B20	Serial I/F Ch.2 Transmit Data Register (pEFSIF2_TXD)	8	Ch.2 transmit data
0x00300B21	Serial I/F Ch.2 Receive Data Register (pEFSIF2_RXD)	8	Ch.2 receive data
0x00300B22	Serial I/F Ch.2 Status Register (pEFSIF2_STATUS)	8	Ch.2 transfer/error status
0x00300B23	Serial I/F Ch.2 Control Register (pEFSIF2_CTL)	8	Sets Ch.2 transfer mode and controls transfer.
0x00300B24	Serial I/F Ch.2 IrDA Register (pEFSIF2_IRDA)	8	Sets Ch.2 asynchronous/IrDA mode.
0x00300B25	Serial I/F Ch.2 Baud-rate Timer Control Register (pEFSIF2_BRTRUN)	8	Controls Ch.2 baud-rate timer.
0x00300B26	Serial I/F Ch.2 Baud-rate Timer Reload Data Register (LSB) (pEFSIF2_BRTRDL)	8	Ch.2 baud-rate timer reload data low-order 8 bits
0x00300B27	Serial I/F Ch.2 Baud-rate Timer Reload Data Register (MSB) (pEFSIF2_BRTRDM)	8	Ch.2 baud-rate timer reload data high-order 4 bits
0x00300B28	Serial I/F Ch.2 Baud-rate Timer Count Data Register (LSB) (pEFSIF2_BRTCDL)	8	Ch.2 baud-rate timer count data low-order 8 bits
0x00300B29	Serial I/F Ch.2 Baud-rate Timer Count Data Register (MSB) (pEFSIF2_BRTCDM)	8	Ch.2 baud-rate timer count data high-order 4 bits
0x00300B4F	Serial I/F STD/ADV Mode Select Register (pEFSIF_ADV)	8	Selects standard or advanced mode.

The following describes each serial interface control register.

The serial interface control registers are mapped in the 8-bit device area from 0x300B00 to 0x300B4F, and can be accessed in units of bytes.

Note: When setting the serial interface control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x300B00–0x300B20: Serial I/F Ch.x Transmit Data Registers (pEFSIFx_TXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.x transmit data register (pEFSIFx_TXD)	00300B00 00300B20 (B)	D7	TXDx7	Serial I/F Ch.x transmit data TXDx7(x6) = MSB TXDx0 = LSB	0x0 to 0xFF(0x7F)	X	R/W	7-bit asynchronous mode does not use TXDx7.
		D6	TXDx6					
		D5	TXDx5					
		D4	TXDx4					
		D3	TXDx3					
		D2	TXDx2					
		D1	TXDx1					
		D0	TXDx0					

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 2.

0x300B00 Serial I/F Ch.0 Transmit Data Register (pEFSIF0_TXD)

0x300B10 Serial I/F Ch.1 Transmit Data Register (pEFSIF1_TXD)

0x300B20 Serial I/F Ch.2 Transmit Data Register (pEFSIF2_TXD)

D[7:0] TXDx[7:0]: Serial I/F Ch.x Transmit Data Bits

Sets transmit data. (Default: indeterminate)

When data is written to this register (transmit data buffer) after 1 is written to TXENx, a transmit operation is begun. The data written to TXDx[7:0] 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 cause of transmit-data empty interrupt occurs.

In 7-bit asynchronous mode, TXDx7 (MSB) is ignored.

The serial-converted data is output from the SOUTx 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.

0x300B01–0x300B21: Serial I/F Ch.x Receive Data Registers (pEFSIF_x_RXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.x receive data register (pEFSIF _x _RXD)	00300B01 00300B21 (B)	D7	RXD _x 7	Serial I/F Ch.x receive data RXD _x 7 (x6) = MSB RXD _x 0 = LSB	0x0 to 0xFF(0x7F)	X	R	7-bit asynchronous mode does not use RXD _x 7 (fixed at 0).
		D6	RXD _x 6					
		D5	RXD _x 5					
		D4	RXD _x 4					
		D3	RXD _x 3					
		D2	RXD _x 2					
		D1	RXD _x 1					
		D0	RXD _x 0					

Note: The letter ‘x’ in bit names, etc., denotes a channel number from 0 to 2.

0x300B01 Serial I/F Ch.0 Receive Data Register (pEFSIF0_RXD)

0x300B11 Serial I/F Ch.1 Receive Data Register (pEFSIF1_RXD)

0x300B21 Serial I/F Ch.2 Receive Data Register (pEFSIF2_RXD)

D[7:0] RXD_x[7:0]: Serial I/F Ch.x Receive Data Bits

The data in the receive data buffer can be read from this register beginning with the oldest data first. The received data enters the receive 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 RDBF_x 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_x[1:0] (one data in standard mode), a cause of receive buffer full interrupt occurs.

In 7-bit asynchronous mode, 0 is stored in RXD_x7.

The serial data input from the SIN_x pin is converted into parallel data beginning with the LSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in this buffer.

This register is a read-only register, so no data can be written to it. (Default: indeterminate)

0x300B02–0x300B22: Serial I/F Ch.x Status Registers (pEFSIFx_STATUS)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks			
Serial I/F Ch.x status register (pEFSIFx_STATUS)	00300B02 00300B22 (B)	D7	RXDxNUM1	Number of Ch.x receive data in FIFO	RXDxNUM[1:0]		Number of data	0	R			
		D6	RXDxNUM0		1	1		4				
					1	0		3				
					0	1		2				
						0	0	1 or 0				
		D5	TENDx	Ch.x transmit-completion flag	1	Transmitting	0	End	0		R	
		D4	FERx	Ch.x framing error flag	1	Error	0	Normal	0		R/W	Reset by writing 0.
		D3	PERx	Ch.x parity error flag	1	Error	0	Normal	0		R/W	
D2	OERx	Ch.x overrun error flag	1	Error	0	Normal	0	R/W				
D1	TDBEx	Ch.x transmit data buffer empty	1	Empty	0	Not empty	1	R				
D0	RDBFx	Ch.x receive data buffer full	1	Full	0	Not full	0	R				

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 2.

- 0x300B02 Serial I/F Ch.0 Status Register (pEFSIF0_STATUS)
- 0x300B12 Serial I/F Ch.1 Status Register (pEFSIF1_STATUS)
- 0x300B22 Serial I/F Ch.2 Status Register (pEFSIF2_STATUS)

D[7:6] RXDxNUM[1:0]: Number of Ch.x Receive Data in FIFO

Indicates the number of data in the receive data buffer (FIFO) that have not been read.

Table V.1.8.2 Number of Receive Data

RXDxNUM1	RXDxNUM0	Number of data
1	1	4
1	0	3
0	1	2
0	0	1 or 0

(Default: 0b00)

When RXDxNUM[1:0] is 0, it indicates that the receive data buffer contains 0 or 1 received data. When RXDxNUM[1:0] is 1 to 3, it indicates that the receive data buffer contains 2 to 4 received data.

D5 TENDx: Serial I/F Ch.x Transmit-Completion Flag

Indicates the transmission status.

- 1 (R): During transmitting
- 0 (R): End of transmission (default)

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 V.1.3.3.1 and Figure V.1.4.3.1). In clock-synchronized slave mode, TENDx goes 0 every time 1-byte data is transmitted (see Figure V.1.3.3.2).

D4 FERx: Serial I/F Ch.x Framing Error Flag

Indicates whether a framing error occurred.

- 1 (R): An error occurred
- 0 (R): No error occurred (default)
- 1 (W): Has no effect
- 0 (W): Reset to 0

FERx 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 ISO7816 or asynchronous mode.

FERx is reset by writing 0 or when RXENx and TXENx both are set to 0.

D3 PERx: Serial I/F Ch.x Parity Error Flag

Indicates whether a parity error occurred.

- 1 (R): An error occurred
- 0 (R): No error occurred (default)
- 1 (W): Has no effect
- 0 (W): Reset to 0

PERx 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 ISO7816 mode or 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 buffer.

PERx is reset by writing 0 or when RXENx and TXENx both are set to 0.

D2 OERx: Serial I/F Ch.x Overrun Error Flag

Indicates whether an overrun error occurred.

- 1 (R): An error occurred
- 0 (R): No error occurred (default)
- 1 (W): Has no effect
- 0 (W): Reset to 0

OERx 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. 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.

OERx is reset by writing 0 or when RXENx and TXENx both are set to 0.

D1 TDBEx: Serial I/F Ch.x Transmit Data Buffer Empty Flag

Indicates the status of the transmit data buffer.

- 1 (R): Not full (default)
- 0 (R): Buffer full

TDBEx 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.

D0 RDBFx: Serial I/F Ch.x Receive Data Buffer Full Flag

Indicates the status of the receive data buffer.

- 1 (R): Not empty
- 0 (R): Buffer empty (default)

RDBFx 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.

0x300B03–0x300B23: Serial I/F Ch.x Control Registers (pEFSIFx_CTL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks			
Serial I/F Ch.x control register (pEFSIFx_CTL)	00300B03 00300B23 (B)	D7	TXEN_x	Ch.x transmit enable	1	Enabled	0	Disabled	0	R/W	Valid only in asynchronous mode.	
		D6	RXEN_x	Ch.x receive enable	1	Enabled	0	Disabled	0	R/W		
		D5	EPR_x	Ch.x parity enable	1	With parity	0	No parity	X	R/W		
		D4	PMD_x	Ch.x parity mode select	1	Odd	0	Even	X	R/W		
		D3	STPB_x	Ch.x stop bit select	1	2 bits	0	1 bit	X	R/W		
		D2	SSCK_x	Ch.x input clock select	1	#SCLK _x	0	Internal clock	X	R/W		
		D1	SMD_x1 SMD_x0	Ch.x transfer mode select	1	SMD _x [1:0]	Transfer mode		X	R/W		
		11					8-bit asynchronous		X			
		10					7-bit asynchronous					
		01					Clock sync. Slave					
D0				00	Clock sync. Master							

Note: The letter ‘x’ in bit names, etc., denotes a channel number from 0 to 2.

- 0x300B03 Serial I/F Ch.0 Control Register (pEFSIF0_CTL)
- 0x300B13 Serial I/F Ch.1 Control Register (pEFSIF1_CTL)
- 0x300B23 Serial I/F Ch.2 Control Register (pEFSIF2_CTL)

D7 TXEN_x: Serial I/F Ch.x Transmit Enable Bit

Enables each channel for transmit operations.

- 1 (R/W): Transmit enabled
- 0 (R/W): Transmit disabled (default)

When TXEN_x for a channel is set to 1, the channel is enabled for transmit operations. When TXEN_x is set to 0, the channel is disabled for transmit operations.

Always make sure TXEN_x = 0 before setting the transfer mode and other conditions.

Writing 0 to TXEN_x clears the transmit data buffer (FIFO) as well as disabling transmit operations.

D6 RXEN_x: Serial I/F Ch.x Receive Enable Bit

Enables each channel for receive operations.

- 1 (R/W): Receive enabled
- 0 (R/W): Receive disabled (default)

When RXEN_x for a channel is set to 1, the channel is enabled for receive operations. When RXEN_x is set to 0, the channel is disabled for receive operations.

Always make sure RXEN_x = 0 before setting the transfer mode and other conditions.

Writing 0 to RXEN_x clears the receive data buffer (FIFO) as well as disabling receive operations.

D5 EPR_x: Serial I/F Ch.x Parity Enable Bit

Selects a parity function for asynchronous transfer. (Default: indeterminate)

- 1 (R/W): Parity added
- 0 (R/W): No parity added

EPR_x 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 EPR_x is set to 1, the receive data is checked for parity. A parity bit is automatically added to the transmit data. When EPR_x is set to 0, parity is not checked and no parity bit is added.

EPR_x is only effective in asynchronous mode. Settings of EPR_x have no effect in clock-synchronized mode. In ISO7816 mode, the parity function is always enabled no matter how EPR_x is set.

D4 PMD_x: Serial I/F Ch._x Parity Mode Select Bit

Selects an odd or even parity for asynchronous transfer. (Default: indeterminate)

1 (R/W): Odd parity

0 (R/W): Even parity

Odd parity is selected by writing 1 to PMD_x, 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 EPR_x is set to 1. If EPR_x = 0, settings of PMD_x do not have any effect. ISO7816 mode supports even parity only.

D3 STPB_x: Serial I/F Ch._x Stop-Bit Length Select Bit

Selects a stop-bit length for asynchronous transfer. (Default: indeterminate)

1 (R/W): 2 bits

0 (R/W): 1 bit

STPB_x is only valid in asynchronous mode. Two stop bits are selected by writing 1 to STPB_x, and one stop bit is selected by writing 0. The start bit is fixed at 1 bit.

Settings of STPB_x are ignored in clock-synchronized mode.

In ISO7816 mode, the stop-bit length is fixed at 2 bits for T = 0 protocol or 1 bit for T = 1 protocol.

D2 SSCK_x: Serial I/F Ch._x Input Clock Select Bit

Selects the clock source for asynchronous transfer. (Default: indeterminate)

1 (R/W): #SCLK_x (external clock)

0 (R/W): Internal clock

During operation in asynchronous mode, this bit is used to select the clock source between an internal clock (output from the baud-rate timer) and an external clock (input from the #SCLK_x pin). An external clock is selected by writing 1 to this bit, and an internal clock is selected by writing 0.

D[1:0] SMD_x[1:0]: Serial I/F Ch._x Transfer Mode Select Bits

Sets the transfer mode of the serial interface as shown in Table V.1.8.3 below.

Table V.1.8.3 Setting of Transfer Mode

SMD _x 1	SMD _x 0	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

(Default: indeterminate)

SMD_x[1:0] can be read as well as written.

When using the IrDA interface, always be sure to set asynchronous mode for the transfer mode.

0x300B04–0x300B24: Serial I/F Ch.x IrDA Registers (pEFSIFx_IRDA)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Serial I/F Ch.x IrDA register (pEFSIFx_IRDA)	00300B04 00300B24 (B)	D7	SRDYCTLx	Ch.x #SRDY control	1	High mask	0	Normal	0	R/W Writing is disabled when SIOADV (D0/0x300B4F) = "0".	
		D6	FIFOINTx1	Ch.x receive buffer full interrupt timing	FIFOINTx[1:0]		Receive level		0		R/W
		D5	FIFOINTx0		11	4	0				
					10	3					
						01	2				
						00	1				
		D4	DIVMDx	Ch.x async. clock division ratio	1	1/8	0	1/16	X	R/W	
		D3	IRTLx	Ch.x IrDA I/F output logic inversion	1	Inverted	0	Direct	X	R/W	Valid only in asynchronous mode.
D2	IRRLx	Ch.x IrDA I/F input logic inversion	1	Inverted	0	Direct	X	R/W			
D1	IRMDx1	Ch.x interface mode select	IRMDx[1:0]		I/F mode		X	R/W			
D0	IRMDx0		11	reserved	X						
			10	IrDA 1.0							
			01	reserved							
				00	General I/F						

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 2.

- 0x300B04 Serial I/F Ch.0 IrDA Register (pEFSIF0_IRDA)
- 0x300B14 Serial I/F Ch.1 IrDA Register (pEFSIF1_IRDA)
- 0x300B24 Serial I/F Ch.2 IrDA Register (pEFSIF2_IRDA)

D7 SRDYCTLx: Serial I/F Ch.x #SRDY Control Bit

Selects a control method for the #SRDYx signal.

1 (R/W): High mask mode

0 (R/W): Normal output (default)

When SRDYCTLx is set to 0, the #SRDYx signal is controlled normally and indicates ready to receive even if the receive data buffer is full. When SRDYCTLx is set to 1, high-mask mode is selected. The following shows the #SRDYx controls in clock-synchronized slave mode and master mode:

Clock-synchronizes slave mode

When the receive data buffer is full, the #SRDYx 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 #SRDYx signal (low) from the slave device is ignored and the serial interface stops outputting the #SCLKx 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 asynchronous mode and ISO7816 mode, this bit is ignored as they do not use the #SRDYx signal.

Note: This bit can be rewritten only when SIOADV (D0/0x300B4F) is set to 1 (advanced mode).

D[6:5] FIFOINTx[1:0]: Serial I/F Ch.x Receive Buffer Full Interrupt Timing Select Bits

Sets the number of data in the receive data buffer to generate a receive-buffer full interrupt.

Table V.1.8.4 Number of Receive Data Buffer

FIFOINTx1	FIFOINTx0	Receive level
1	1	4
1	0	3
0	1	2
0	0	1

(Default: 0b00)

Writing 0–3 to FIFOINTx[1:0] 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 cause flag FSRXx are set to 1.

Note: This bit can be rewritten only when SIOADV (D0/0x300B4F) is set to 1 (advanced mode).

D4 DIVMDx: Serial I/F Ch.x Clock Division Ratio Select Bit

Selects the division ratio of the sampling clock. (Default: indeterminate)

1 (R/W): 1/8

0 (R/W): 1/16

Select the division ratio necessary to generate the sampling clock for asynchronous or ISO7816 transfer. When DIVMDx is set to 1, the sampling clock is generated from the input clock of the serial interface (output from the baud-rate timer or input from #SCLKx) by dividing it by 8. When DIVMDx is set to 0, the input clock is divided by 16.

D3 IRTLx: Serial I/F Ch.x IrDA I/F Output Logic Inversion Bit

Inverts the logic of the IrDA output signal. (Default: indeterminate)

1 (R/W): Inverted

0 (R/W): Not inverted

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).

D2 IRRLx: Serial I/F Ch.x IrDA I/F Input Logic Inversion Bit

Inverts the logic of the IrDA input signal. (Default: indeterminate)

1 (R/W): Inverted

0 (R/W): Not inverted

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.

D[1:0] IRMDx[1:0]: Serial I/F Ch.x Interface Mode Select Bits

Selects the IrDA interface function.

Table V.1.8.5 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

(Default: indeterminate)

When using the IrDA interface function, write 0b10 to IRMDx[1:0] while setting to asynchronous mode for the transfer mode. If the IrDA interface function is not to be used, write 0b00 to IRMDx[1:0].

Note: This selection must always be performed before the transfer mode and other conditions are set.

0x300B05–0x300B25: Serial I/F Ch.x Baud-rate Timer Control Registers (pEFSIFx_BRTRUN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.x baud-rate timer control register (pEFSIFx_BRTRUN)	00300B05	D7–1	–	reserved		–	–	0 when being read.
	00300B25 (B)	D0	BRTRUNx	Baud-rate timer Run/Stop control	1 Run 0 Stop	0	R/W	

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 2.

0x300B05 Serial I/F Ch.0 Baud-rate Timer Control Register (pEFSIF0_BRTRUN)

0x300B15 Serial I/F Ch.1 Baud-rate Timer Control Register (pEFSIF1_BRTRUN)

0x300B25 Serial I/F Ch.2 Baud-rate Timer Control Register (pEFSIF2_BRTRUN)

D[7:1] Reserved

D0 BRTRUNx: Baud-rate Timer Run/Stop Control Bit

Controls the baud-rate timer's RUN/STOP states.

1 (R/W): Run

0 (R/W): Stop (default)

The baud-rate timer loads the reload data BRTRDx[11:0] (0x300Bx6–0x300Bx7) to its counter and starts counting down when 1 is written to BRTRUNx. The baud-rate timer stops counting when 0 is written to BRTRUNx.

0x300B06–0x300B26: Serial I/F Ch.x Baud-rate Timer Reload Data Registers (LSB) (pEFSIFx_BRTRDL)

0x300B07–0x300B27: Serial I/F Ch.x Baud-rate Timer Reload Data Registers (MSB) (pEFSIFx_BRTRDM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.x baud-rate timer reload data register (LSB) (pEFSIFx_BRTRDL)	00300B06 00300B26 (B)	D7	BRTRDx7	Serial I/F Ch.x	baud-rate timer reload data [7:0]	0x0 to 0xFF (BRTRDx[11:0] = 0x0 to 0xFFF)	0	R/W
		D6	BRTRDx6					
		D5	BRTRDx5					
		D4	BRTRDx4					
		D3	BRTRDx3					
		D2	BRTRDx2					
		D1	BRTRDx1					
		D0	BRTRDx0					
Serial I/F Ch.x baud-rate timer reload data register (MSB) (pEFSIFx_BRTRDM)	00300B07 00300B27 (B)	D7–4	–	reserved		–	–	0 when being read.
		D3	BRTRDx11	Serial I/F Ch.x	baud-rate timer reload data [11:8]	0x0 to 0xF (BRTRDx[11:0] = 0x0 to 0xFFF)	0	R/W
		D2	BRTRDx10					
		D1	BRTRDx9					
		D0	BRTRDx8					

Note: The letter ‘x’ in bit names, etc., denotes a channel number from 0 to 2.

- 0x300B06 Serial I/F Ch.0 Baud-rate Timer Reload Data Register (LSB) (pEFSIF0_BRTRDL)
- 0x300B07 Serial I/F Ch.0 Baud-rate Timer Reload Data Register (MSB) (pEFSIF0_BRTRDM)
- 0x300B16 Serial I/F Ch.1 Baud-rate Timer Reload Data Register (LSB) (pEFSIF1_BRTRDL)
- 0x300B17 Serial I/F Ch.1 Baud-rate Timer Reload Data Register (MSB) (pEFSIF1_BRTRDM)
- 0x300B26 Serial I/F Ch.2 Baud-rate Timer Reload Data Register (LSB) (pEFSIF2_BRTRDL)
- 0x300B27 Serial I/F Ch.2 Baud-rate Timer Reload Data Register (MSB) (pEFSIF2_BRTRDM)

D[7:0]/0x300Bx6 BRTRDx[7:0]: Baud-rate Timer Reload Data [7:0]

D[3:0]/0x300Bx7 BRTRDx[11:8]: Baud-rate Timer Reload Data [11:8]

Set the initial counter value of the baud-rate timer. (Default: 0x000)

The reload data set in this register is loaded into the counter, and the counter starts counting down beginning with this value, 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 BRTRUNx (D0/0x300Bx5), or when data is automatically reloaded upon counter under-flow.

0x300B08–0x300B28: Serial I/F Ch.x Baud-rate Timer Count Data Registers (LSB) (pEFSIF_x_BRTCDL)

0x300B09–0x300B29: Serial I/F Ch.x Baud-rate Timer Count Data Registers (MSB) (pEFSIF_x_BRTCDM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.x baud-rate timer count data register (LSB) (pEFSIF _x _BRTCDL)	00300B08 00300B28 (B)	D7	BRTCDx7	Serial I/F Ch.x baud-rate timer count data [7:0]	0x0 to 0xFF (BRTCDx[11:0] = 0x0 to 0xFFFF)	0	R	
		D6	BRTCDx6			0		
		D5	BRTCDx5			0		
		D4	BRTCDx4			0		
		D3	BRTCDx3			0		
		D2	BRTCDx2			0		
		D1	BRTCDx1			0		
		D0	BRTCDx0			0		
Serial I/F Ch.x baud-rate timer count data register (MSB) (pEFSIF _x _BRTCDM)	00300B09 00300B29 (B)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	BRTCDx11	Serial I/F Ch.x baud-rate timer count data [11:8]	0x0 to 0xF (BRTCDx[11:0] = 0x0 to 0xFFFF)	0	R	
		D2	BRTCDx10			0		
		D1	BRTCDx9			0		
		D0	BRTCDx8			0		
		D0	BRTCDx8			0		

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 2.

- 0x300B08 Serial I/F Ch.0 Baud-rate Timer Count Data Register (LSB) (pEFSIF0_BRTCDL)
- 0x300B09 Serial I/F Ch.0 Baud-rate Timer Count Data Register (MSB) (pEFSIF0_BRTCDM)
- 0x300B18 Serial I/F Ch.1 Baud-rate Timer Count Data Register (LSB) (pEFSIF1_BRTCDL)
- 0x300B19 Serial I/F Ch.1 Baud-rate Timer Count Data Register (MSB) (pEFSIF1_BRTCDM)
- 0x300B28 Serial I/F Ch.2 Baud-rate Timer Count Data Register (LSB) (pEFSIF2_BRTCDL)
- 0x300B29 Serial I/F Ch.2 Baud-rate Timer Count Data Register (MSB) (pEFSIF2_BRTCDM)

D[7:0]/0x300Bx8 BRTCDx[7:0]: Baud-rate Timer Count Data [7:0]

D[3:0]/0x300Bx9 BRTCDx[11:8]: Baud-rate Timer Count Data [11:8]

The baud-rate timer data can be read out from this register. (Default: 0x000)

This register function as a buffer that retain the counter data when read out, enabling the data to be read out at any time.

0x300B1A: Serial I/F Ch.1 ISO7816 Mode Control Register (pEFSIF1_7816CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Serial I/F Ch.1 ISO7816 mode control register (pEFSIF1_7816CTL)	00300B1A (B)	D7	RPNUM12	Serial I/F Ch.1 number of transmit repetition	0x0 to 0x7		0	R/W			
		D6	RPNUM11				0				
		D5	RPNUM10				0				
		D4	CLKOEN1	Ch.1 clock output enable	1	Enabled	0	Disabled		0	R/W
		D3	CLKOL1	Ch.1 clock output forced low	1	Normal	0	Forced low		0	R/W
		D2	MSBSEL1	Ch.1 MSB first selection	1	MSB first	0	LSB first		0	R/W
		D1	7816MD11	Serial I/F Ch.1 ISO7816 mode selection	7816MD1[1:0]	Mode		0		R/W	
		D0	7816MD10			11	reserved	0			
	10		ISO7816, T = 1								
	01		ISO7816, T = 0								
			00	Normal I/F							

D[7:5] RPNUM1[2:0]: Serial I/F Ch.1 Number of Transmit Repetition Setup Bits

Sets the retransmit count when a transmit error occurs. (Default: 0)

The T = 0 protocol allows retransmission of data when an error occurs in data transmission (when the receiver returns NACK). Retransmission can be repeated if the error occurs successively, and RPNUM1[2:0] is used to set the retransmit count. A maximum of seven retransmissions may be specified. When RPNUM1[2:0] is set to 0, this interface does not retransmit data even if a transmit error occurs.

When a transmit error occurs in T = 1 protocol mode, this interface does not retransmit data regardless of how RPNUM1[2:0] is set.

D4 CLKOEN1: Serial I/F Ch.1 Clock Output Enable Bit

Enables clock output in ISO7816 mode.

1 (W): Enable

0 (W): Disable

1 (R): Clock is being output.

0 (R): Clock is stopped (default)

By setting CLKONE1 to 1 after CLKOL1 (D3) is set to 1, the clock is output from the #SCLK1 pin. When reading, CLKOEN1 indicates whether the clock is being output or stopped. To disables the clock output, wait until CLKOEN1 is actually cleared to 0 after writing 0 to CLKOEN1, and then set CLKOL1 (D3) to 0.

CLKOEN1 and CLKOL1 (D3) must be set to 0 to output the specified number of clocks using CLKNEN1 (D7/0x300B1F) and CLKN1[6:0] (D[6:0]/0x300B1F).

D3 CLKOL1: Serial I/F Ch.1 Clock Output Forced Low Bit

Sets the clock output signal forcibly to low in ISO7816 mode.

1 (R/W): Normal output

0 (R/W): Forced low (default)

When CLKOL1 is set to 0, the #SCLK1 pin output is fixed at a low level. To output the clock normally, set CLKOL1 to 1 then CLKOEN1 (D4) to 1. To disable the clock output, wait until CLKOEN1 (D4) is actually cleared to 0 after writing 0 to CLKOEN1 (D4), and then set CLKOL1 to 0.

CLKOL1 and CLKOEN1 (D4) must be set to 0 to output the specified number of clocks using CLKNEN1 (D7/0x300B1F) and CLKN1[6:0] (D[6:0]/0x300B1F).

D2 MSBSEL1: Serial I/F Ch.1 MSB First Select Bit

Selects the data shift direction, MSB first or LSB first, in ISO7816 mode.

1 (R/W): MSB first

0 (R/W): LSB first (default)

When MSBSEL1 is 0, data is serially output from the D0 bit first. To output data beginning with the D7 bit, set MSBSEL1 to 1.

D[1:0] 7816MD1[1:0]: Serial I/F Ch.1 ISO7816 Mode Select Bits

Sets the serial interface in ISO7816 mode.

Table V.1.8.6 Selecting ISO7816 Mode

7816MD11	7816MD10	Mode
1	1	Reserved
1	0	ISO7816 (T = 1) mode
0	1	ISO7816 (T = 0) mode
0	0	Normal interface

(Default: 0b00)

0x300B1B: Serial I/F Ch.1 ISO7816 Mode Status Register (pEFSIF1_7816STA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Serial I/F Ch.1 ISO7816 mode status register (pEFSIF1_7816STA)	00300B1B (B)	D7-1	–	reserved	–			–	–	0 when being read.
		D0	TER1	Ch.1 ISO7816 transmit error flag	1	Error	0	Normal	0	R/W

D[7:1] Reserved**D0 TER1: Serial I/F Ch.1 ISO7816 Transmit Error Flag**

Indicates that a transmit error occurs in ISO7816 (T = 0) mode.

- 1 (R): An error occurred
- 0 (R): No error occurred (default)
- 1 (W): Has no effect
- 0 (W): Reset to 0

TER1 is set to 1 if the receiver returns an error signal (NACK) when data has been transmitted in ISO7816 (T = 0) mode. When RPNUM1[2:0] (D[7:5]/0x300B1A) has been set to 0 (retransmission disabled), a transmit error occurs when NACK is returned once. If a retransmit count (not 0) is set to RPNUM1[2:0] (D[7:5]/0x300B1A), the serial interface retransmit data when NACK is returned. This retransmission is repeated up to the set retransmit count if the receiver cannot receive that data correctly. The NACK signals returned in this period do not cause a transmit error. If NACK has returned after the last retransmission, TER1 is set to 1.

TER1 is reset by writing 0 or when RXEN1 and TXEN1 both are set to 0.

**0x300B1C: Serial I/F Ch.1 ISO7816 Mode FI/DI Ratio Register (LSB)
(pEFSIF1_FIDIL)**

**0x300B1D: Serial I/F Ch.1 ISO7816 Mode FI/DI Ratio Register (MSB)
(pEFSIF1_FIDIM)**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.1 ISO7816 mode FI/DI ratio register (LSB) (pEFSIF1_FIDIL)	00300B1C (B)	D7	FIDI17	Serial I/F Ch.1 ISO7816 mode FI/DI ratio [7:0]	0x0 to 0xFF (FIDI1[13:0] = 0x0 to 0x3FFF)	0	R/W	Valid only in ISO7816 mode.
		D6	FIDI16			0		
		D5	FIDI15			0		
		D4	FIDI14			0		
		D3	FIDI13			0		
		D2	FIDI12			0		
		D1	FIDI11			0		
		D0	FIDI10			0		
		Serial I/F Ch.1 ISO7816 mode FI/DI ratio register (MSB) (pEFSIF1_FIDIM)	00300B1D (B)			D7-6	—	
D5	FIDI113			Serial I/F Ch.1 ISO7816 mode FI/DI ratio [13:8]	0x0 to 0x3F (FIDI1[13:0] = 0x0 to 0x3FFF)	0	R/W	Valid only in ISO7816 mode.
D4	FIDI112					0		
D3	FIDI111					0		
D2	FIDI110					0		
D1	FIDI19					0		
D0	FIDI18					0		

D[7:0]/0x300B1C FIDI1[7:0]: ISO7816 Mode FI/DI Ratio [7:0]

D[5:0]/0x300B1D FIDI1[13:8]: ISO7816 Mode FI/DI Ratio [13:8]

Sets the FI/DI ratio for generating the ISO7816 clock. (Default: 0x0000)

The bit rate in ISO7816 mode is determined by the equation shown below.

$$\text{bps} = \frac{D}{F} \times \text{fsio_clk}$$

bps: Bit rate (bits/second)

D: Bit rate adjustment value

1, 2, 4, 8, 16, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64

F: Clock frequency divide value

372 (5 MHz), 558 (6 MHz), 744 (8 MHz), 1116 (12 MHz), 1488 (16 MHz), 1860 (20 MHz),
512 (5 MHz), 768 (7.5 MHz), 1024 (10 MHz), 1536 (15 MHz), 2048 (20 MHz)

() indicates the maximum output clock frequency.

fsio_clk: ISO7816 clock frequency (baud-rate timer output and #SCLK1 output)

The sampling clock frequency for asynchronous transfer is determined by the following equation:

$$\text{fsAMPL} = \frac{D}{F} \times \text{fsio_clk} \times \frac{1}{\text{DIVMD}}$$

fsAMPL: Sampling clock frequency

DIVMD: Divide ratio internally used by the serial interface (1/16 or 1/8, selected with DIVMD1)

FIDI1[13:0] should be set to $F \times \text{DIVMD} / D - 1$. Tables V.1.8.7 and V.1.8.8 list the values that can be set to FIDI1[13:0].

V PERIPHERAL MODULES 3 (INTERFACE): GENERAL-PURPOSE SERIAL INTERFACE (EFSIO)

Table V.1.8.7 FID1[13:0] Set Values (DIVMD = 1/8)

F/(D×8) -1		D										
		1	2	4	8	16	1/2	1/4	1/8	1/16	1/32	1/64
F	372	46	22	11	5	2	92	185	371	743	1487	2975
	558	69	34	16	8	3	139	278	557	1115	2231	4463
	744	92	46	22	11	5	185	371	743	1487	2975	5951
	1116	139	69	34	16	8	278	557	1115	2231	4463	8927
	1488	185	92	46	22	11	371	743	1487	2975	5951	11903
	1860	232	115	57	28	14	464	929	1859	3719	7439	14879
	512	63	31	15	7	3	127	255	511	1023	2047	4095
	768	95	47	23	11	5	191	383	767	1535	3071	6143
	1024	127	63	31	15	7	255	511	1023	2047	4095	8191
	1536	191	95	47	23	11	383	767	1535	3071	6143	12287
2048	255	127	63	31	15	511	1023	2047	4095	8191	16383	

Table V.1.8.8 FID1[13:0] Set Values (DIVMD = 1/16)

F/(D×16) -1		D										
		1	2	4	8	16	1/2	1/4	1/8	1/16	1/32	1/64
F	372	22	11	5	2	0	46	92	185	371	743	1487
	558	34	16	8	3	1	69	139	278	557	1115	2231
	744	46	22	11	5	2	92	185	371	743	1487	2975
	1116	69	34	16	8	3	139	278	557	1115	2231	4463
	1488	92	46	22	11	5	185	371	743	1487	2975	5951
	1860	115	57	28	14	6	232	464	929	1859	3719	7439
	512	31	15	7	3	1	63	127	255	511	1023	2047
	768	47	23	11	5	2	95	191	383	767	1535	3071
	1024	63	31	15	7	3	127	255	511	1023	2047	4095
	1536	95	47	23	11	5	191	383	767	1535	3071	6143
2048	127	63	31	15	7	255	511	1023	2047	4095	8191	

0x300B1E: Serial I/F Ch.1 Transmit Time Guard Register (pEFSIF1_TTGR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Serial I/F Ch.1 transmit time guard register (pEFSIF1_TTGR)	00300B1E (B)	D7	TTGR17	Serial I/F Ch.1	0x0 to 0xFF	0	R/W	
		D6	TTGR16	transmit time guard		0		
		D5	TTGR15			0		
		D4	TTGR14			0		
		D3	TTGR13			0		
		D2	TTGR12			0		
		D1	TTGR11			0		
		D0	TTGR10			0		

D[7:0] TTGR1[7:0]: Serial I/F Ch.1 Transmit Time Guard Setup Bits

Sets the time guard function for data transmission. (Default: 0x00)

The ISO7816 mode supports a time guard function that inserts an idle time between characters during transmission. The idle time to be inserted can be specified in ETU (bit cycle) units using TTGR1[7:0]. When TTGR1[7:0] is set to 0, no idle time is inserted. When a value other than 0 is set, the SOUT1 output is fixed at high for the specified ETU period after a stop bit is output. This high output period is regarded as a long stop bit.

0x300B1F: Serial I/F Ch.1 ISO7816 Mode Output Clock Setup Register (pEFSIF1_CLKNUM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Serial I/F Ch.1 ISO7816 mode output clock setup register (pEFSIF1_CLKNUM)	00300B1F (B)	D7	CLKNEN1	Ch.1 CLKN enable	1	Enabled	0	Disabled	0	R/W	
		D6	CLKN16	Serial I/F Ch.1 number of output clocks	0x0 to 0x7F		0	R/W			
		D5	CLKN15		0						
		D4	CLKN14		0						
		D3	CLKN13		0						
		D2	CLKN12		0						
		D1	CLKN11		0						
		D0	CLKN10		0						

D7 CLKNEN1: Serial I/F Ch.1 Transmit Enable Bit

Controls ISO7816 clock output according to the number of clocks specified using CLKN1[6:0].

1 (W): Enable

0 (W): Disable

1 (R): Clock is being output.

0 (R): Clock is stopped (default)

By writing 1 to CLKNEN1, the clocks of which the number is specified with CLKN1[6:0] (D[6:0]) is output from the #SCLK1 pin. The clock output starts in synchronization with the sampling clock, therefore, it is not synchronized with writing to CLKNEN1. When reading, CLKNEN1 indicates whether the clock is being output or stopped.

CLKOEN1 (D4/0x300B1A) and CLKOL1 (D3/0x300B1A) must be set to 0 to output the specified number of clocks using CLKNEN1 and CLKN1[6:0] (D[6:0]).

Use CLKOEN1 (D4/0x300B1A) and CLKOL1 (D3/0x300B1A) to control normal clock output in ISO7816 mode.

D[6:0] CLKN1[6:0]: Serial I/F Ch.1 Number of Output Clocks Setup Bits

Sets the number of clocks output in ISO7816 mode. (Default: 0x00)

If the set value is an even number, the number of clocks is set to $CLKN1[6:0] / 2$.

If the set value is an odd number, the number of clocks is set to $(CLKN1[6:0] - 1) / 2$.

CLKOEN1 (D4/0x300B1A) and CLKOL1 (D3/0x300B1A) must be set to 0 to output the specified number of clocks using CLKNEN1 (D7) and CLKN1[6:0].

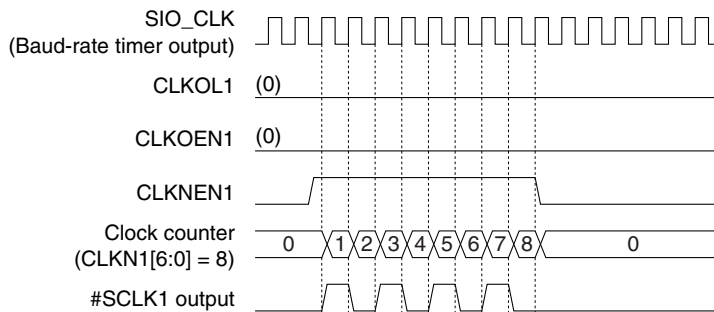


Figure V.1.8.1 Clock Output with Number of Clocks Specified

0x300B4F: Serial I/F STD/ADV Mode Select Register (pEFSIF_ADV)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
Serial I/F STD/ADV mode select register (pEFSIF_ADV)	00300B4F (B)	D7-1	-	reserved	-			-	-	Writing 1 not allowed.
		D0	SIOADV	Standard mode/advanced mode select	1	Advanced mode	0	Standard mode	0	R/W

D[7:1] Reserved

D0 SIOADV: Standard/Advanced Mode Select Bit

Selects standard or advanced mode.

1 (R/W): Advanced mode

0 (R/W): Standard mode (default)

The serial interface in the S1C33E08 is extended from that of the C33 STD models. The S1C33E08 serial interface has two operating modes, standard (STD) mode of which functions are compatible with the existing C33 STD models and an advanced (ADV) mode allowing use of the extended functions. Table V.1.8.9 shows differences between standard mode and advanced mode.

Table V.1.8.9 Differences between Standard Mode and Advanced Mode

Function	Standard mode	Advanced mode
#SRDY mask control	Disabled	Enabled
Number of received data in the buffer to generate a receive-buffer full interrupt	One	One to four can be specified.

To configure the serial interface in advanced mode, set SIOADV to 1. The control bits (SRDYCTLx and FIFOINTx[1:0]) for the extended functions are enabled to write after this setting.

Note: Standard or advanced mode currently set is applied to all the serial interface channels. It cannot be selected for each channel individually.

V.1.9 Precautions

- Before setting various serial-interface parameters, make sure the transmit and receive operations are disabled (TXEN_x = RXEN_x = 0).
 - * **TXEN_x**: Serial I/F Ch._x Transmit Enable Bit in the Serial I/F Ch._x Control Register (D7/0x300Bx3)
 - * **RXEN_x**: Serial I/F Ch._x Receive Enable Bit in the Serial I/F Ch._x Control Register (D6/0x300Bx3)
- When the serial interface is transmitting or receiving data, do not set TXEN_x or RXEN_x to 0, and do not execute the slp instruction.
- In clock-synchronized transfers, the mode of communication is half-duplex, in which the clock line is shared between the transmit and receive units. Therefore, RXEN_x and TXEN_x cannot be enabled simultaneously.
- After an initial reset, the cause-of-interrupt flags become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, reset these flags in the program.
- If a receive error occurs, the receive-error interrupt and receive-buffer full interrupt causes 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 cause flag through the use of the receive-error interrupt processing routine.
- To prevent the regeneration of interrupts due to the same cause of interrupt following the occurrence of an interrupt, always be sure to reset the cause-of-interrupt flag before setting the PSR again or executing the reti instruction.
- Follow the procedure described below to initialize the serial interface.

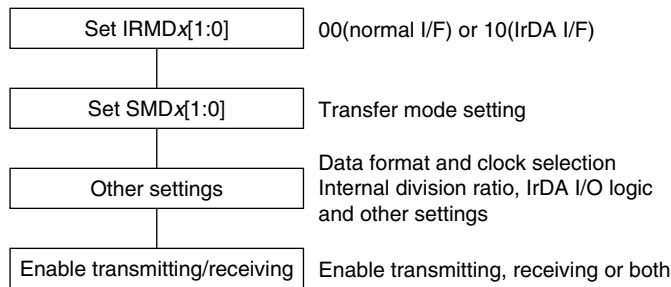


Figure V.1.9.1 Serial Interface Initialize Procedure

- When transmitting data in clock-synchronized master mode, transmit data is written to the transmit data register after the initial setting is performed following the flow above. 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.
- The maximum transfer rate of the serial interface is limited to 8 Mbps in clock-synchronized mode or 1 Mbps in asynchronous mode. Do not set a transfer rate (baud rate) that exceeds the limit.
- If the receive circuit is stopped during reception, set both transmission and reception to the disabled status.
- 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.
- When the transmit-enable bit TXEN_x is set to 0 to disable transmit operations, the transmit data buffer (FIFO) is cleared (initialized). Similarly, when the receive-enable bit RXEN_x 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.
- During IrDA receive operations, the RZI circuit recognizes low pulses by means of the signal edge (rising edge when IRRL_x = 0; falling edge when IRRL_x = 1). Note that noise may cause a malfunction.
 - * **IRRL_x**: Serial I/F Ch._x IrDA I/F Input Logic Inversion Bit in the Serial I/F Ch._x IrDA Register (D2/0x300Bx4)

V.2 Serial Peripheral Interface (SPI)

V.2.1 Outline of SPI Module

The S1C33E08 contains a synchronous serial interface module (hereafter SPI module) compatible with Motorola SPI®. The following shows its features:

- Supports both master and slave modes.
- Supports 1 to 32-bit data transfer.
- Programmable bit-rate configuration
- 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, receive data register full, and receive data overflow interrupts.
- Generates DMA requests from transmit data register empty and receive data register full.
- SPI-EEPROM boot is possible (see Appendix for details on booting).

Figure V.2.1.1 shows the structure of the SPI module.

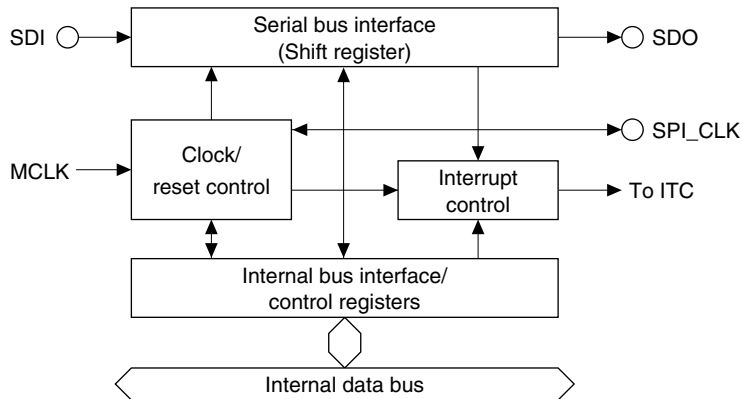


Figure V.2.1.1 Structure of SPI Module

V.2.2 I/O Pins of SPI Module

Table V.2.2.1 lists the I/O pins used by the SPI module.

Table V.2.2.1 SPI Pin Configuration

Pin name	I/O	Function
SDI	I	Data input
SDO	O	Data output
SPI_CLK	I/O	Clock output (master mode) or clock input (slave mode)

SDI pin

This pin is used to input serial data.

SDO pin

This pin is used to output serial data.

SPI_CLK pin

In master mode, this pin is used to output the SPI clock to slave devices.

In slave mode, this pin is used to input the SPI clock from the master device.

Note: The SPI input/output pins are shared with general-purpose I/O ports or other peripheral circuit inputs/outputs, so that functionality in the initial state is set to other than the SPI input/output. Before the SPI input/output signals assigned to these pins can be used, the function of these pins must be switched for the SPI input/output by setting the corresponding Port Function Select Registers. For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

V.2.3 SPI Operating Clock

The SPI module use the SPI_CLK clock (= MCLK) generated by the CMU as the operating clock. The transfer clock is generated in the SPI module by dividing SPI_CLK.

Controlling the supply of the operating clock

SPI_CLK is supplied to the SPI module with default settings. It can be turned off using SPI_CKE (D6/0x301B04) to reduce the amount of power consumed on the chip if the SPI module is not used.

* **SPI_CKE**: SPI Clock Control Bit in the Gated Clock Control Register 1 (D6/0x301B04)

Setting SPI_CKE (D6/0x301B04) to 0 (1 by default) turns off the clock supply to the SPI module. When the clock supply is turned off, the SPI module control registers cannot be accessed.

For details on how to set and control the clock, refer to Section III.1, “Clock Management Unit (CMU).”

Note: The Gated Clock Control Register 1 (0x301B04) is write-protected. Write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless re-writing said registers.

Clock state in standby mode

The clock supply to the SPI module stops depending on type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

SLEEP mode: The operating clock supply stops.

Therefore, the SPI module also stops operating in SLEEP mode.

V.2.4 Setting SPI Module

When performing data transfers via the SPI bus, the following settings must be made before data transfer is actually begun:

1. Setting input/output pins
2. Selecting master or slave mode
3. Setting the data bit width
4. Setting the bit rate
5. Setting the SPI_CLK polarity and phase
6. Setting the inter-character wait cycle
7. Setting the receive data mask
8. Setting interrupts and IDMA/HSDMA

The following explains the content of each setting. For details on interrupt/DMA settings, refer to Section V.2.6, “SPI Interrupts and DMA.”

Note: Always make sure the SPI module is inactive (ENA (D0/0x301708) = 0) before these settings are made. A change of settings during operation may cause a malfunction.

* **ENA:** SPI Enable Bit in the SPI Control Register 1 (D0/0x301708)

Setting input/output pins

The SDI, SDO, and SPI_CLK pins are used for SPI. Configure the Port Function Select Registers to enable these pin functions. For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Selecting master or slave mode

Use MODE (D1/0x301708) to select whether the SPI module is set to master mode or slave mode. Setting MODE (D1/0x301708) to 1 selects master mode, and setting to 0 (default) selects slave mode. In master mode, the SPI performs data transfer using the clock generated in the module. In slave mode, the SPI performs data transfer using a clock input from the master device.

* **MODE:** SPI Mode Select Bit in the SPI Control Register 1 (D1/0x301708)

Setting the data bit width

Use BPT[4:0] (D[14:10]/0x301708) to set the data bit width of the transfer data (characters). Data bit width is set as the BPT[4:0] (D[14:10]/0x301708) set value + 1 (for example, 16 bits when BPT[4:0] = 15).

* **BPT[4:0]:** Number of Data Bits Per Transfer Setup Bits in the SPI Control Register 1 (D[14:10]/0x301708)

Setting the bit rate

When the SPI module is set in master mode, the synchronous clock is generated inside the module. The synchronous clock drives the shift register and is output from the SPI_CLK pin to slave devices. Specify the clock frequency using MCBR[2:0] (D[6:4]/0x301708) to determine the bit rate.

* **MCBR[2:0]:** Master Clock Bit Rate Setup Bits in the SPI Control Register 1 (D[6:4]/0x301708)

Table V.2.4.1 Setting the Clock Frequency

MCBR2	MCBR1	MCBR0	Clock frequency (Hz)
1	1	1	MCLK/512
1	1	0	MCLK/256
1	0	1	MCLK/128
1	0	0	MCLK/64
0	1	1	MCLK/32
0	1	0	MCLK/16
0	0	1	MCLK/8
0	0	0	MCLK/4

Slave mode does not need to set a bit rate as the SPI module operates with the clock input from the master device.

Setting the SPI_CLK polarity and phase

Use CPOL (D8/0x301708) to select the SPI_CLK clock polarity. The SPI_CLK is configured as active low when CPOL (D8/0x301708) is set to 1 or active high when CPOL (D8/0x301708) is set to 0 (default).

* **CPOL**: SPI_CLK Polarity Select Bit in the SPI Control Register 1 (D8/0x301708)

The SPI_CLK clock phase is selected with CPHA (D9/0x301708).

* **CPHA**: SPI_CLK Phase Select Bit in the SPI Control Register 1 (D9/0x301708)

Setting these control bits determines the transfer timing as in the figure shown below.

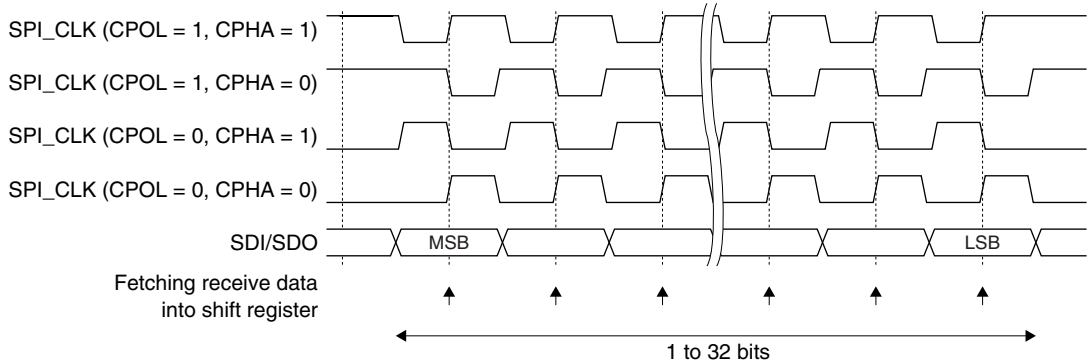


Figure V.2.4.1 Clock and Data Transfer Timing

Setting the inter-character wait cycle

1 to 65536 SPI_CLK clocks of delay time can be inserted between data transfers (in each transfer for specified number of data bits) using the SPI Wait Register (0x301710). The value set in the register (0 to 65535) + 1 is used as the number of wait cycles.

Setting the receive data mask

(1) RXME (D1/0x30171C) = 0 (default)

The SPI Receive Data Register (0x301700) will receive the data bits specified with BPT[4:0] (D[14:10]/0x301708) + 1. The ineffective upper bits are masked with 0.

(2) RXME (D1/0x30171C) = 1

This setting enables user specified bit mask. The SPI receive buffer will receive the data bits specified with BPT[4:0] (D[14:10]/0x301708) + 1. The ineffective upper bits are masked with 0. Then only the effective data bits specified with RXMASK[4:0] (D[14:10]/0x30171C) + 1 in the receive data buffer are loaded to the SPI Receive Data Register (0x301700). The ineffective upper bits are masked with 0.

* **RXME**: Receive Data Mask Enable Bit in the SPI Receive Data Mask Register (D1/0x30171C)

* **RXMASK[4:0]**: Receive Data Mask Setup Bits in the SPI Receive Data Mask Register (D[14:10]/0x30171C)

Figure V.2.4.2 shows the relationship between the mask control bit settings and the receive data loaded to the SPI Receive Data Register (0x301700).

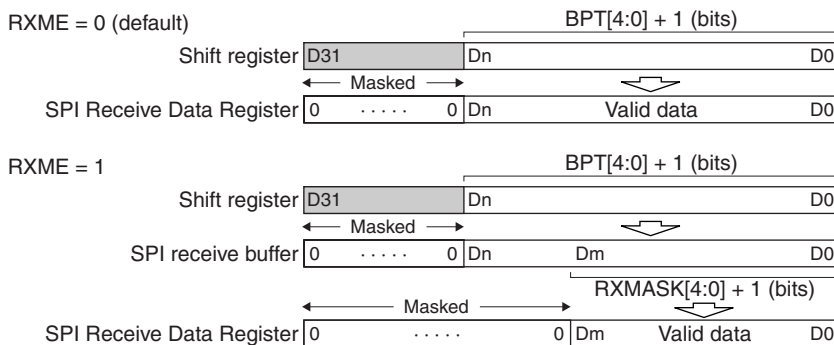


Figure V.2.4.2 Receive Data Mask

V.2.5 Control of Data Transfer

Data transmission

The following shows the data-transmit procedure:

1. Set up the SPI conditions as described in the previous section.
2. Set up the interrupt and DMA conditions using the ITC registers and the SPI interrupt control register (explained later). When using the SPI interrupt, the cause of SPI interrupt flag in the ITC must be cleared before enabling the interrupt.
3. Write 1 to the ENA (D0/0x301708) to turn the SPI circuit on.
In master mode, the SPI circuit starts frequency division of the source clock.

* **ENA**: SPI Enable Bit in the SPI Control Register 1 (D0/0x301708)

4. In slave mode, write 1 to SS (D10/0x30170C) to set this slave SPI into selected status. This enables clock input from the SPI_CLK pin.

* **SS**: Slave Select Control Bit in the SPI Control Register 2 (D10/0x30170C)

In master mode, SS (D10/0x30170C) must be set to 0.

5. Write the transmit data to the SPI Transmit Data Register (0x301704).

The SPI circuit loads the data written to the register into the shift register. In master mode, the SPI circuit starts outputting the clock from the SPI_CLK pin. In slave mode, the SPI circuit waits for clock input from the SPI_CLK pin. The data bits in the shift register are shifted one by one at the rising or falling edge configured with CPHA (D9/0x301708) and CPOL (D8/0x301708) (see Figure V.2.4.1), and are output from the SDO pin. The MSB of data is transmitted first.

* **CPHA**: SPI_CLK Phase Select Bit in the SPI Control Register 1 (D9/0x301708)

* **CPOL**: SPI_CLK Polarity Select Bit in the SPI Control Register 1 (D8/0x301708)

The SPI circuit provides TDEF (D4/0x301714) to indicate the SPI Transmit Data Register (0x301704) 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 (D4/0x301714) is set to 1 by polling or using this interrupt before the next transmit data can be written to the SPI Transmit Data Register (0x301704).

* **TDEF**: Transmit Data Empty Flag in the SPI Status Register (D4/0x301714)

Furthermore, by setting TXDE (D3/0x301708) to 1, a transmit DMA request is output to the ITC. This DMA request can be used to set transmit data without using the interrupt above.

* **TXDE**: Transmit DMA Enable Bit in the SPI Control Register 1 (D3/0x301708)

The SPI circuit continues data output from the SDO pin and clock input/output from/to the SPI_CLK pin until data transmission for the number of bits specified with the BPT[4:0] (D[14:10]/0x301708) is finished. If the next transmit data exists in the SPI Transmit Data Register (0x301704) when a data transfer has finished in master mode, the SPI circuit repeats the same transmit operation as above. However, the SPI circuit delays starting the next transmission for the number of SPI_CLK cycles specified with the SPI Wait Register (0x301710). The data written to the transmit data register is not loaded into the shift register until after the expiration of the delay time.

When a continuous data transmission is being performed in slave mode, the SPI Wait Register (0x301710) does not affect the transmission (no delay is inserted) as the clock is controlled by the master device. The next transmit data must be written to the SPI Transmit Data Register (0x301704) before the master starts sending the next data transfer clocks.

In master mode, the transmitter status sets/resets BSYP (D6/0x301714). BSYP (D6/0x301714) is set to 1 when transmission is in progress or in the wait cycles specified with the SPI Wait Register (0x301710), and reset to 0 upon completion of a transmit operation. Use this flag to check if a transmission has completed. This flag is ineffective in slave mode (always 0 is read).

* **BSYP**: Transfer Busy Flag in the SPI Status Register (D6/0x301714)

- After all transmissions have completed, write 0 to ENA (D0/0x301708) to turn the SPI circuit off.
In slave mode, write 0 to SS (D10/0x30170C) to set this SPI slave to deselected status before writing 0 to ENA (D0/0x301708).

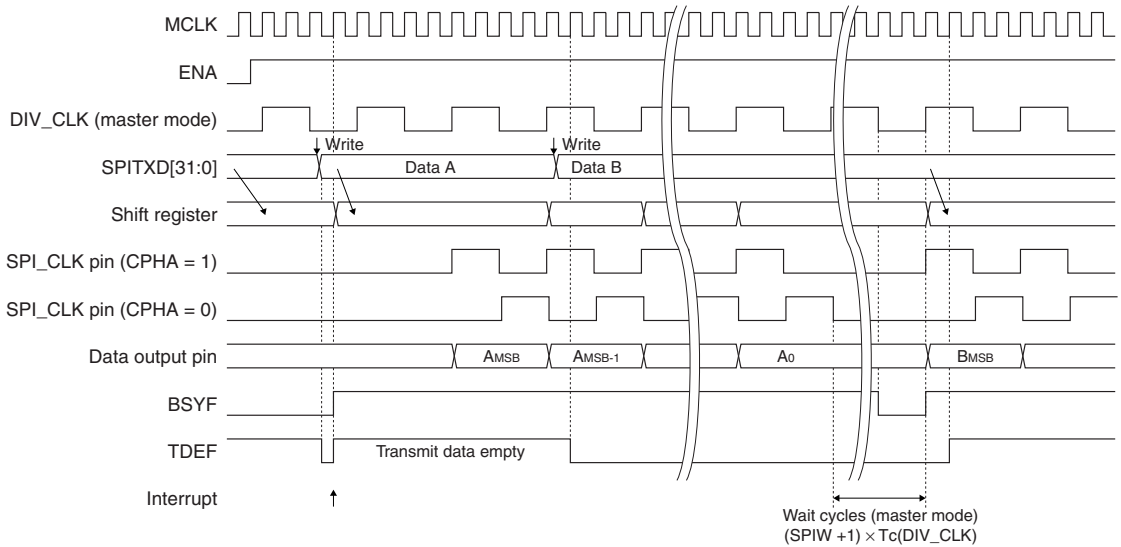


Figure V.2.5.1 Data Transmit Timing Chart (CPOL = 0)

Data receiving

The following shows the data-receive procedure:

- Set up the SPI conditions as described in the previous section.
- Set up the interrupt and DMA conditions using the ITC registers and the SPI interrupt control register (explained later). When using the SPI interrupt, the cause of SPI interrupt flag in the ITC must be cleared before enabling the interrupt.
- Write 1 to the ENA (D0/0x301708) to turn the SPI circuit on.
In master mode, the SPI circuit starts frequency division of the source clock.
- In slave mode, write 1 to SS (D10/0x30170C) to set this slave SPI into selected state. This enables clock input from the SPI_CLK pin.
In master mode, SS (D10/0x30170C) must be set to 0.
- In master mode, write dummy data to the SPI Transmit Data Register (0x301704). Writing to the SPI Transmit Data Register (0x301704) is used as the trigger for data receiving as well as start of data transmission. Also actual data to be transmitted can be written as the SPI circuit performs data transmission and reception simultaneously.

The SPI circuit starts output of the generated clock from the SPI_CLK pin.

In slave mode, the SPI circuit waits for clock input from the SPI_CLK pin. When performing data transmission and reception simultaneously, the transmit data should be written to the SPI Transmit Data Register (0x301704) before a clock is input.

The data bits are fetched in the shift register one by one at the rising or falling edge configured with CPHA (D9/0x301708) and CPOL (D8/0x301708) (see Figure V.2.4.1). 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 (0x301700). The bit mask processing is performed in this loading stage. At the same time, RDFF (D2/0x301714) is set to 1 (data full) to indicate that the receive data can be read from the SPI Receive Data Register (0x301700) and a data receive interrupt can be generated.

* **RDFF**: Receive Data Full Flag in the SPI Status Register (D2/0x301714)

- Check to see if the RDFFF (D2/0x301714) is set to 1 by polling or using the interrupt and read data from the SPI Receive Data Register (0x301700). When data is read from the SPI Receive Data Register (0x301700), RDFFF (D2/0x301714) is cleared to 0.

Furthermore, by setting RXDE (D2/0x301708) to 1, a receive DMA request is output to the ITC. This DMA request can be used to store the received data to other memory without using the interrupt above.

* **RXDE**: Receive DMA Enable Bit in the SPI Control Register 1 (D2/0x301708)

To receive data successively in master mode, write dummy data or transmit data to the SPI Transmit Data Register (0x301704) every time a data frame is received. The SPI circuit continues the receive operation when any data has been written to the SPI Transmit Data Register (0x301704). However, the SPI circuit delays starting the next receiving for the number of SPI_CLK cycles specified with the SPI Wait Register (0x301710). The clock output is suspended until after the expiration of the delay time.

When receiving data in slave mode without any data transmission, it is not necessary to write data to the SPI Transmit Data Register (0x301704). The receive process activates by the clock input from the master device. When performing data transmission simultaneously, write transmit data to the SPI Transmit Data Register (0x301704) according to the data transmit procedure.

- Repeat Steps 5 and 6 until all data are received.

In the same manner as transmission, BSYF (D6/0x301714) is set to 1 when data is being received in master mode.

- After all data has been received, write 0 to ENA (D0/0x301708) to turn the SPI circuit off.

In slave mode, write 0 to SS (D10/0x30170C) to set this SPI slave to deselected status before writing 0 to ENA (D0/0x301708).

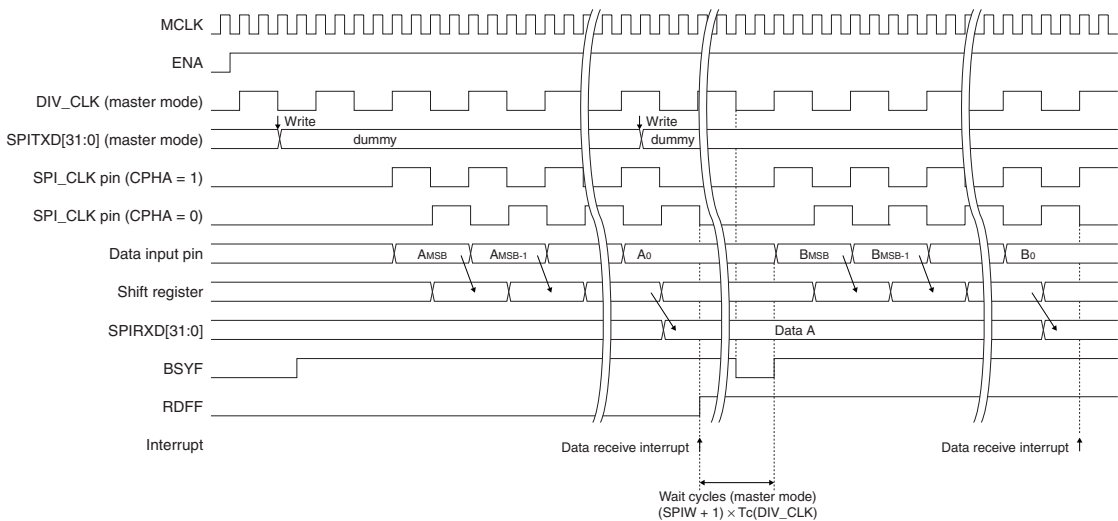


Figure V.2.5.2 Data Receive Timing Chart (CPOL = 0)

Receive data overflow

The SPI Receive Data Register (0x301700) is overwritten if a data reception has finished when the previously received data has not been read from the register. Therefore, when data is being received continuously, receive data must be read before the following data reception finishes.

If the SPI Receive Data Register (0x301700) is overwritten when RDFFF (D2/0x301714) = 1 (the received data has not been read yet), RDOF (D3/0x301714) is set to 1. A receive data overflow interrupt can be generated simultaneous with this flag set to 1. Use this interrupt for error recovery.

* **RDOF**: Receive Data Overflow Flag in the SPI Status Register (D3/0x301714)

RDOF (D3/0x301714) is reset to 0 by reading data from the SPI Receive Data Register (0x301700).

Data transmit/receive trigger and precautions

The SPI circuit performs data transmission and reception simultaneously.

When data transmission is started, a data receive operation is also performed. Therefore, if received data that has not been read exists in the SPI Receive Data Register (0x301700), it will be overwritten. Furthermore, undesired data receive interrupts 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 SPI Receive Data Register (0x301700) to clear RDFE (D2/0x301714) and RDOF (D3/0x301714), since the flags have been set due to the receive operation performed simultaneously with the previous data transmission.

When receiving data, the data transmit interrupt should be disabled in the same way as the data transmission.

V.2.6 SPI Interrupts and DMA

The SPI module can generate the following three types of interrupts:

- Transmit DMA interrupt (transmit data empty)
- Receive DMA interrupt (receive data full)
- SPI interrupt (transmit data empty, receive data full, receive data overflow)

Transmit DMA interrupt

A cause of interrupt occurs when the transmit data set in the SPI Transmit Data Register (0x301704) is transferred to the shift register. When TXDE (D3/0x301708) has been set to 1, the interrupt request signal is output to the ITC and it sets the cause-of-interrupt flag FSPITX (D5/0x300289) in the ITC to 1.

* **TXDE**: Transmit DMA Enable Bit in the SPI Control Register 1 (D3/0x301708)

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 cause of interrupt indicates that the next transmit data can be written to the transmit data register. This cause of interrupt can also be used to invoke DMA, enabling transmit data to be written to the register by means of a DMA transfer.

Receive DMA interrupt

A cause of interrupt occurs when the data received in the shift register is loaded into the SPI Receive Data Register (0x301700). When RXDE (D2/0x301708) has been set to 1, the interrupt request signal is output to the ITC and it sets the cause-of-interrupt flag FSPIRX (D4/0x300289) in the ITC to 1.

* **RXDE**: Receive DMA Enable Bit in the SPI Control Register 1 (D2/0x301708)

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 cause of interrupt indicates that the received data can be read out. This cause of interrupt can also be used to invoke DMA, enabling the received data to be written into specified memory locations by means of a DMA transfer.

SPI interrupt

In addition to the two interrupt request signals shown above, the SPI module outputs one more interrupt request signal. This interrupt request circuit is configured as Figure V.2.6.1 and it allows selection of one or more causes of interrupt.

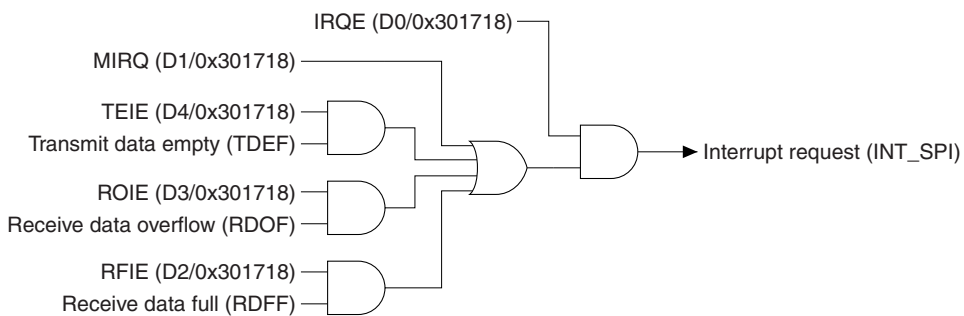


Figure V.2.6.1 SPI Interrupt Request Circuit

To output the SPI interrupt requests, enable interrupts of the causes described below and set IRQE (D0/0x301718) to 1. When IRQE (D0/0x301718) is set to 0, no SPI interrupt request is output.

* **IRQE**: Interrupt Request Enable Bit in the SPI Interrupt Control Register (D0/0x301718)

Transmit data empty

A cause of interrupt occurs when the transmit data set in the SPI Transmit Data Register (0x301704) is transferred to the shift register, in which case TDEF (D4/0x301714) is set to 1. Set TEIE (D4/0x301718) to 1 to output an interrupt request by this cause of interrupt.

- * **TDEF**: Transmit Data Empty Flag in the SPI Status Register (D4/0x301714)
- * **TEIE**: Transmit Data Empty Interrupt Enable Bit in the SPI Interrupt Control Register (D4/0x301718)

This interrupt request occurs by the same cause of interrupt as the transmit DMA interrupt.

Receive data full

A cause of interrupt occurs when the data received in the shift register is loaded into the SPI Receive Data Register (0x301700), in which case Rdff (D2/0x301714) is set to 1. Set RFIE (D2/0x301718) to 1 to output an interrupt request by this cause of interrupt.

- * **Rdff**: Receive Data Full Flag in the SPI Status Register (D2/0x301714)
- * **RFIE**: Receive Data Full Interrupt Enable Bit in the SPI Interrupt Control Register (D2/0x301718)

This interrupt request occurs by the same cause of interrupt as the receive DMA interrupt.

Receive data overflow

A cause of interrupt occurs when receive data is loaded into the SPI Receive Data Register (0x301700) before the previous data in the register is read out, in which case RDOF (D3/0x301714) is set to 1. Set ROIE (D3/0x301718) to 1 to output an interrupt request by this cause of interrupt.

- * **RDOF**: Receive Data Overflow Flag in the SPI Status Register (D3/0x301714)
- * **ROIE**: Receive Data Overflow Interrupt Enable Bit in the SPI Interrupt Control Register (D3/0x301718)

Manual interrupt request

An SPI interrupt request can be output manually by setting MIRQ (D1/0x301718) to 1. After an interrupt occurs by this operation, write 0 to MIRQ (D1/0x301718) to negate the SPI interrupt request signal.

- * **MIRQ**: Manual IRQ Set/Clear Bit in the SPI Interrupt Control Register (D1/0x301718)

The SPI interrupt request is sent to the ITC as the port 8 input interrupt (FPT8) signal and it sets the cause-of-interrupt flag FP8 (D0/0x3002A9) in the ITC to 1. However, INT_SPI must be selected for the port 8 input interrupt.

Control registers of the interrupt controller

Table V.2.6.1 shows the interrupt controller's control registers provided for each interrupt source.

Table V.2.6.1 Control Register of Interrupt Controller

Interrupt	Cause-of-interrupt flag	Interrupt enable register	Interrupt priority register
Transmit DMA interrupt	FSPITX(D5/0x300289)	ESPITX(D5/0x300279)	PSP1[2:0](D[6:4]/0x30026E)
Receive DMA interrupt	FSPIRX(D4/0x300289)	ESPIRX(D4/0x300279)	
SPI interrupt	FP8(D0/0x3002A9)	EP8(D0/0x3002A6)	PP8L[2:0](D[2:0]/0x3002A0)

When a cause of interrupt described above occurs, the corresponding cause-of-interrupt flag is set to 1. If the interrupt enable register bit for that cause of interrupt has been set to 1, an interrupt request is generated.

Interrupts can be disabled by leaving the interrupt enable register bit for that cause of interrupt set to 0. The cause-of-interrupt 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 Section III.2, "Interrupt Controller (ITC)."

Note: The SPI interrupt request signal is input to the port 8 input interrupt (FPT8) system. The port 8 input interrupt circuit is configured by selecting a port (signal) to be used for generating an interrupt from P90, INT_SPI, P80, and P70. When using the SPI interrupt, set SPT8[1:0] (D[1:0]/0x3003C4) to 10 to select INT_SPI. This setting enables the INT_SPI signal to be sent to the ITC as the port 8 input interrupt signal. Furthermore, SPPT8 (D0/0x3003C6), which selects the polarity of the FPT8 input signal, should be set to 1 (high level or rising edge).

- * **SPT8[1:0]:** FPT8 Interrupt Input Port Select Bits in the Port Input Interrupt Select Register 3 (D[1:0]/0x3003C4)
- * **SPPT8:** FPT8 Input Polarity Select Bit in the Port Input Interrupt Polarity Select Register 2 (D0/0x3003C6)

Intelligent DMA

The transmit DMA, receive DMA and SPI interrupt requests can be used to invoke intelligent DMA (IDMA). This enables successive transmit/receive operations between memory and the transmit/receive-register to be performed by means of a DMA transfer.

The following shows the IDMA channel numbers set for each cause of interrupt:

IDMA Ch.

Receive DMA interrupt: 0x24
 Transmit DMA interrupt: 0x25
 SPI interrupt (FPT8 interrupt): 0x26

The IDMA request and enable bits shown in Table V.2.6.2 must be set to 1 for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

Table V.2.6.2 Control Bits for IDMA Transfer

Interrupt	IDMA request bit	IDMA enable bit
Transmit DMA interrupt	RSPITX(D5/0x30029B)	DESPITX(D5/0x30029C)
Receive DMA interrupt	RSPIRX(D4/0x30029B)	DESPIRX(D4/0x30029C)
SPI interrupt	RP8(D0/0x3002AC)	DEP8(D0/0x3002AE)

If a cause of interrupt 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 DMA transfer performed.

For details on DMA transfer and how to control interrupts upon completion of DMA transfer, refer to Section II.2, “Intelligent DMA (IDMA).”

High-speed DMA

Each interrupt can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit corresponding to each interrupt:

Table V.2.6.3 HSDMA Trigger Set-up Bits

Interrupt	HSDMA Ch.	Trigger set-up bits
Transmit DMA interrupt	2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2–3 Trigger Set-up Register (0x300299)
Receive DMA interrupt	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2–3 Trigger Set-up Register (0x300299)
SPI interrupt	0	HSD0S[3:0] (D[3:0]) / HSDMA Ch.0–1 Trigger Set-up Register (0x300298)

For HSDMA to be invoked by a transmit DMA interrupt request, the trigger set-up bits for HSDMA Ch.2 should be set to “1001.” For HSDMA to be invoked by a receive DMA interrupt request, the trigger set-up bits for HSDMA Ch.3 should be set to “1001.” For HSDMA to be invoked by an SPI interrupt (FPT8 interrupt) request, the trigger set-up bits for HSDMA Ch.0 should be set to “1101.” Transfer conditions, etc. must also be set on the HSDMA side. The HSDMA channel is invoked through generation of the cause of interrupt.

For details on HSDMA transfer, refer to Section II.1, “High-Speed DMA (HSDMA).”

Trap vectors

The default trap-vector address of each cause of interrupt is set as follows:

Receive DMA interrupt: 0xC00144
 Transmit DMA interrupt: 0xC00148
 SPI interrupt (FPT8 interrupt): 0xC00150

The base address of the trap table can be changed using the TTBR register.

V.2.7 Details of Control Registers

Table V.2.7.1 List of SPI Registers

Address	Register name	Size	Function
0x00301700	SPI Receive Data Register (pSPI_RXD)	32	Receive data
0x00301704	SPI Transmit Data Register (pSPI_TXD)	32	Transmit data
0x00301708	SPI Control Register 1 (pSPI_CTL1)	32	Sets SPI transfer conditions
0x0030170C	SPI Control Register 2 (pSPI_CTL2)	32	Controls slave mode
0x00301710	SPI Wait Register (pSPI_WAIT)	32	Sets inter-character wait cycle
0x00301714	SPI Status Register (pSPI_STAT)	32	SPI transfer/error status
0x00301718	SPI Interrupt Control Register (pSPI_INT)	32	Controls SPI interrupts
0x0030171C	SPI Receive Data Mask Register (pSPI_RXMK)	32	Sets receive data bit mask

The following describes each SPI control register.

The SPI control registers are mapped in the 32-bit device area from 0x301700 to 0x30171C, and can be accessed in units of words.

- Notes:**
- The SPI control registers allow accessing in word size only. Do not read/write the registers in half-word or byte size.
 - When setting the SPI control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x301700: SPI Receive Data Register (pSPI_RXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI receive data register (pSPI_RXD)	00301700 (W)	D31	SPIRXD31	SPI receive data	0x0 to 0xFFFFFFFF	0x0	R	
		D0	SPIRXD0	SPIRXD31 = MSB SPIRXD0 = LSB				

D[31:0] SPIRXD[31:0]: SPI Receive Data Bits

Stores received data. (Default: 0x0)

When a receive operation is completed and the data received in the shift register is loaded to this register, RDFF (D2/0x301714) is set to 1 (data full). At the same time, a cause of receive data full interrupt occurs. 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 and RDOF (D3/0x301714) is set to 1 (data overflow). At the same time, a cause of receive data overflow interrupt occurs.

The serial data input from the SDI pin is converted into parallel data beginning with the MSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in this register.

Furthermore, the upper bits of the received data specified with the SPI Receive Data Mask Register (0x30171C) can be masked (set to 0) when it is loaded from the shift register.

This register is a read-only register, so no data can be written to it.

0x301704: SPI Transmit Data Register (pSPI_TXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI transmit data register (pSPI_TXD)	00301704 (W)	D31	SPITXD31	SPI transmit data	0x0 to 0xFFFFFFFF	0x0	R/W	
				SPITXD31 = MSB				
		D0	SPITXD0	SPITXD0 = LSB				

D[31:0] SPITXD[31:0]: SPI Transmit Data Bits

Sets transmit data. (Default: 0x0)

In master mode, data transmission begins by writing data to this register. In slave mode, the register contents are transferred to the shift register to start data transmission when a clock is input from the master device.

TDEF (D4/0x301714) is set to 1 (empty) when the data is transferred to the shift register. A cause of data transmit interrupt 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 32 using BPT[4:0] (D[14:10]/0x301708), only the specified number of low-order bits in this register is transmitted.

0x301708: SPI Control Register 1 (pSPI_CTL1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI control register 1 (pSPI_CTL1)	00301708 (W)	D31-15	–	reserved	–	–	–	0 when being read.
		D14	BPT4	Number of data bits per transfer	Number of data bits per transfer = BPT + 1	0	R/W	
		D13	BPT3			0		
		D12	BPT2			0		
		D11	BPT1			0		
		D10	BPT0			0		
		D9	CPHA	SPI_CLK phase selection	1 Phase 1	0 Phase 0	0	R/W
		D8	CPOL	SPI_CLK polarity selection	1 Active low	0 Active high	0	R/W
		D7	MWEN	reserved	Fix at 0.		0	–
		D6	MCBR2	Master clock bit rate (in master mode only)	Master clock divided value = $4 \times 2^{\text{MCBR}}$		0	R/W
		D5	MCBR1				0	
		D4	MCBR0				0	
		D3	TXDE	Transmit DMA enable	1 Enabled	0 Disabled	0	R/W
D2	RXDE	Receive DMA enable	1 Enabled	0 Disabled	0	R/W		
D1	MODE	SPI mode selection	1 Master	0 Slave	0	R/W		
D0	ENA	SPI enable	1 Enabled	0 Disabled	0	R/W		

D[31:15] Reserved

D[14:10] BPT[4:0]: Number of Data Bits Per Transfer Setup Bits

Sets the number of transfer data bits. (Default: 0x0)

The set value in this register + 1 (1 to 32) is the number of bits to be transmitted/received per data transfer.

D9 CPHA: SPI_CLK Phase Select Bit

Selects the phase of the SPI clock. (Default: 0)

This bit controls the data transfer timing in conjunction with the CPOL (D8) bit (see Figure V.2.7.1).

D8 CPOL: SPI_CLK Polarity Select Bit

Selects the polarity of the SPI clock.

1 (R/W): Active low

0 (R/W): Active high (default)

This bit controls the data transfer timing in conjunction with the CPHA (D9) bit (see Figure V.2.7.1).

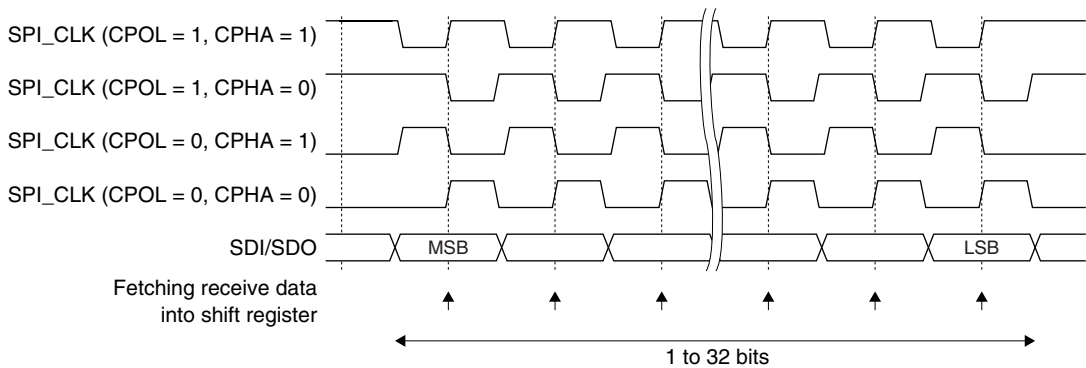


Figure V.2.7.1 Clock and Data Transfer Timing

D7 Reserved (Do not write 1 to this bit.)

D[6:4] MCBR[2:0]: Master Clock Bit Rate Setup Bits

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

Table V.2.7.2 Setting Bit Rate

MCBR2	MCBR1	MCBR0	Clock frequency (Hz)
1	1	1	MCLK/512
1	1	0	MCLK/256
1	0	1	MCLK/128
1	0	0	MCLK/64
0	1	1	MCLK/32
0	1	0	MCLK/16
0	0	1	MCLK/8
0	0	0	MCLK/4

(Default: 0x000)

Slave mode does not need to set a bit rate as the SPI module operates with the clock input from the master device.

D3 TXDE: Transmit DMA Enable Bit

Enables/disables transmit DMA interrupts.

1 (R/W): Enable

0 (R/W): Disable (default)

When TXDE is set to 1, transmit DMA interrupt requests to the ITC are enabled. A transmit DMA interrupt request occurs when the data written to the SPI Transmit Data Register (0x301704) is transferred to the shift register (transmit operation started). At this time, the cause-of-interrupt flag FSPITX (D5/0x300289) in the ITC is set to 1 if TXDE has been set to 1 (enabled). This interrupt request can invoke HSDMA.

When TXDE is set to 0, transmit DMA interrupts are not generated.

D2 RXDE: Receive DMA Enable Bit

Enables/disables receive DMA interrupts.

1 (R/W): Enable

0 (R/W): Disable (default)

When RXDE is set to 1, receive DMA interrupt requests to the ITC are enabled. A receive DMA interrupt request occurs when the data received in the shift register is loaded to the SPI Receive Data Register (0x301700) (receive operation completed). At this time, the cause-of-interrupt flag FSPIRX (D4/0x300289) in the ITC is set to 1 if RXDE has been set to 1 (enabled). This interrupt request can invoke HSDMA.

When RXDE is set to 0, receive DMA interrupts are not generated.

D1 MODE: SPI Mode Select Bit

Sets the SPI module in master or slave mode.

1 (R/W): Master mode

0 (R/W): Slave mode (default)

Setting MODE to 1 selects master mode, and setting to 0 selects slave mode. In master mode, the SPI performs data transfer using the clock generated in the module. In slave mode, the SPI performs data transfer using a clock input from the master device.

D0 ENA: SPI Enable Bit

Enables/disables operation of the SPI module.

1 (R/W): Enabled (on)

0 (R/W): Disabled (off) (default)

When ENA is set to 1, the SPI module starts operating and data transfer is enabled.

When ENA is set to 0, the SPI module goes off.

Make sure that this bit is 0 before setting up data transfer conditions using the SPI registers.

0x30170C: SPI Control Register 2 (pSPI_CTL2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
SPI control register 2 (pSPI_CTL2)	0030170C (W)	D31-12	–	reserved	–	–	–	0 when being read.	
		D11	SSA	reserved	Fix at 0.	0	–		
		D10	SS	Slave select control	Fix at 0.		0	R/W	Master mode
					1 SPI select	0 SPI deselect			Slave mode
		D9	SSP	reserved	Fix at 0.	0	–		
		D8	SSC	reserved	Fix at 0.	0	–		
		D7-3	–	reserved	–	–	–	–	0 when being read.
		D2	RDYP	reserved	Fix at 0.	0	–		
		D1	RDYS	reserved	Fix at 0.	0	–		
		D0	RDYE	reserved	Fix at 0.	0	–		

D[31:11] Reserved (Do not write 1 to this bit.)

D10 SS: Slave Select Control Bit

Sets the SPI module in selected state in slave mode.

1 (R/W): Selected

0 (R/W): Not selected (default)

Write 1 to SS before performing data transmission/reception in slave mode. In slave mode, setting both ENA and SS to 1 enables clock input from the master device and data transmission/reception is enabled. In master mode, SS must be fixed at 0.

D[9:0] Reserved (Do not write 1 to this bit.)

0x301710: SPI Wait Register (pSPI_WAIT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI wait register (pSPI_WAIT)	00301710 (W)	D31 D0	SPIW31 SPIW0	Wait cycle control SPIW31 = MSB SPIW0 = LSB	Number of wait cycles = SPIW[31:0] + 1 (1 to 65536)	0x0	R/W	

D[31:0] SPIW[31:0]: Wait Cycle Control Bits

Sets the number of wait cycles to be inserted between data transfers (characters). The set value in this register + 1 is the number of wait cycles. 1 to 65536 SPI_CLK clock cycles can be specified.

0x301714: SPI Status Register (pSPI_STAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
SPI status register (pSPI_STAT)	00301714 (W)	D31-7	–	reserved		–	–	0 when being read.	
		D6	BSYF	Transfer busy flag	1 Busy 0 Idle	0	R	Master mode	
		D5	MFEF	reserved		–	–	–	0 when being read.
		D4	TDEF	Transmit data empty flag	1 Empty 0 Not empty	1	R		
		D3	RDOF	Receive data overflow flag	1 Occurred 0 Not occurred	0	R		
		D2	RDFE	Receive data full flag	1 Full 0 Not full	0	R		
		D1-0	–	reserved		–	–	–	0 when being read.

D[31:7] Reserved**D6 BSYF: Transfer Busy Flag**

Indicates the SPI transmit/receive operation status in master mode.

1 (R): Busy

0 (R): Idle (default)

BSYF is set to 1 when the SPI starts data transmission/reception in master mode and stays 1 while data transmission/reception is in progress including the wait cycles inserted. BSYF is cleared to 0 upon completion of transmit/receive operation.

BSYF is ineffective in slave mode (always 0).

D5 Reserved**D4 TDEF: Transfer Data Empty Flag**

Indicates the SPI Transmit Data Register (0x301704) status.

1 (R): Empty (default)

0 (R): Not empty

TDEF is cleared to 0 when transmit data is written to the SPI Transmit Data Register (0x301704) 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 SPI Transmit Data Register (0x301704) when this bit = 1.

D3 RDOF: Receive Data Overflow Flag

Indicates receive data overflow status.

1 (R): Overflow occurred

0 (R): Overflow not occurred (default)

RDOF is set to 1 to indicate that the SPI Receive Data Register (0x301700) is overwritten when a data reception has completed before the previously received data in the register is read out. This bit is reset to 0 when the data is read out.

D2 RDFF: Receive Data Full Flag

Indicates the SPI Receive Data Register (0x301700) status.

1 (R): Data full

0 (R): Not full (default)

RDFF is set to 1 when the data received in the shift register is loaded to the SPI Receive Data Register (0x301700) (receive operation completed), indicating that the received data can be read out. This bit is reset to 0 when the data is read out.

D[1:0] Reserved

0x301718: SPI Interrupt Control Register (pSPI_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
SPI interrupt control register (pSPI_INT)	00301718 (W)	D31–6	–	reserved	–	–	–	0 when being read.		
		D5	MFIE	reserved	Fix at 0.	0	–			
		D4	TEIE	Transmit data empty int. enable	1	Enabled	0	Disabled	0	R/W
		D3	ROIE	Receive overflow interrupt enable	1	Enabled	0	Disabled	0	R/W
		D2	RFIE	Receive data full 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

D[31:5] Reserved (Do not write 1 to this bit.)

D4 TEIE: Transmit Data Empty Interrupt Enable Bit

Enables/disables SPI interrupt caused by transmit data empty.

1 (R/W): Enable

0 (R/W): Disable (default)

When TEIE is set to 1, SPI (transmit data empty) interrupt requests to the ITC are enabled. A transmit data empty interrupt request occurs when the data written to the SPI Transmit Data Register (0x301704) is transferred to the shift register (transmit operation started). At this time, the cause-of-interrupt flag FP8 (D0/0x3002A9) in the ITC is set to 1 if both TEIE and IRQE (D0) have been set to 1 (enabled).

When TEIE is set to 0, SPI interrupts caused by transmit data empty are not generated.

D3 ROIE: Receive Data Overflow Interrupt Enable Bit

Enables/disables SPI interrupt caused by receive data overflow.

1 (R/W): Enable

0 (R/W): Disable (default)

When ROIE is set to 1, SPI (receive data overflow) interrupt requests to the ITC are enabled. A receive data overflow interrupt request occurs when a data reception has completed before the previously received data in the SPI Receive Data Register (0x301700) is read out. At this time, the cause-of-interrupt flag FP8 (D0/0x3002A9) in the ITC is set to 1 if both ROIE and IRQE (D0) have been set to 1 (enabled).

When ROIE is set to 0, SPI interrupts caused by receive data overflow are not generated.

D2 RFIE: Receive Data Full Interrupt Enable Bit

Enables/disables SPI interrupt caused by receive data full.

1 (R/W): Enable

0 (R/W): Disable (default)

When RFIE is set to 1, SPI (receive data full) interrupt requests to the ITC are enabled. A receive data full interrupt request occurs when the data received in the shift register is loaded to the SPI Receive Data Register (0x301700) (receive operation completed). At this time, the cause-of-interrupt flag FP8 (D0/0x3002A9) in the ITC is set to 1 if both RFIE and IRQE (D0) have been set to 1 (enabled).

When RFIE is set to 0, SPI interrupts caused by receive data full are not generated.

D1 MIRQ: Manual IRQ Set/Clear Bit

Generates an SPI interrupt request to the ITC by manual control.

1 (R/W): Set IRQ

0 (R/W): Clear IRQ (default)

If MIRQ is set to 1 when IRQE (D0) is set to 1, the SPI interrupt request signal to be delivered to the ITC becomes active. As a result, the cause-of-interrupt flag FP8 (D0/0x3002A9) 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 cause-of-interrupt flag FP8 (D0/0x3002A9) cannot be cleared to 0 by writing 0 to this bit.

D0 IRQE: Interrupt Request Enable Bit

Enables/disables SPI interrupt requests to the ITC.

1 (R/W): Enable

0 (R/W): Disable (default)

When IRQE is set to 1, SPI interrupt requests to the ITC are enabled. An interrupt request is generated and the cause-of-interrupt flag FP8 (D0/0x3002A9) is set to 1 when an enabled cause of SPI interrupt occurs or MIRQ (D1) is set to 1 manually.

When IRQE is set to 0, an SPI interrupt request to the ITC is not generated even if the SPI interrupt for each cause is enabled.

Also manual interrupt requests using MIRQ (D1) are disabled.

0x30171C: SPI Receive Data Mask Register (pSPI_RXMK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI receive data mask register (pSPI_RXMK)	0030171C (W)	D31-15	–	reserved	–	–	–	0 when being read.
		D14	RXMASK4	Bit mask for reading received data	0x0 to 0x1F	0	R/W	
		D13	RXMASK3					
		D12	RXMASK2					
		D11	RXMASK1					
		D10	RXMASK0					
		D9-2	–	reserved	–	–	–	0 when being read.
		D1	RXME	Receive data mask enable	1 Enabled	0 Disabled	0	R/W
D0	–	reserved	–	–	–	–	Do not write 1.	

D[31:15] Reserved

D[14:10] RXMASK[4:0]: Receive Data Mask Setup Bits

Specifies the number of bits for data mask when reading only the required lower bits of the receive data. (Default: 0x0)

Set the MSB of the effective bits (e.g., 31 = not masked, 15 = D[31:16] is masked).

To enable the bit mask using RXMASK[4:0], set RXME (D1) to 1. When the specified bit mask is enabled, the received data is read out with the masked bits set to 0 from the SPI Receive Data Register (0x301700).

D[9:2] Reserved

D1 **RXME: Receive Data Mask Enable Bit**

Enables the RXMASK[4:0] (D[14:10]) setting.

1 (R/W): Enable

0 (R/W): Disable (default)

By setting RXME to 1, the upper bits of the received data are masked (set to 0) according to the RXMASK[4:0] setting when it is loaded from the receive data buffer to the SPI Receive Data Register (0x301700).

By setting RXME to 0, the upper bits of the received data that exceed the data bit length specified with BPT[4:0] (D[14:10]/0x301708) are masked (set to 0) when it is loaded from the shift register to the SPI Receive Data Register (0x301700).

Figure V.2.7.2 shows the relationship between the mask control bit settings and the receive data loaded to the SPI Receive Data Register (0x301700).

D0 Reserved

Do not set this bit to 1.

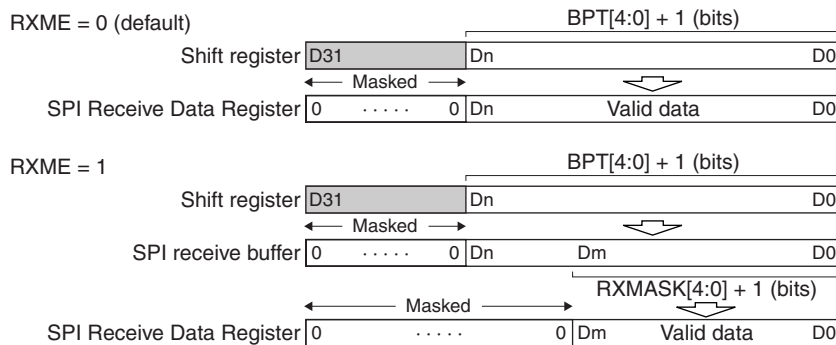


Figure V.2.7.2 Receive Data Mask

V.2.8 Precautions

- Be sure to use 32-bit access instructions for reading/writing from/to the SPI control registers (0x301700 to 0x30171C). The SPI control registers do not allow reading/writing using 16-bit and 8-bit access instructions.
- Do not access the SPI Control Register 1 (0x301708), SPI Control Register 2 (0x30170C), and SPI Wait Register (0x301710) while the BSYF (D6/0x301714) is set to 1 (during data transfer).
 - * **BSYF**: Transfer Busy Flag in the SPI Status Register (D6/0x301714)
- To prevent malfunctions, write 0x0 to the SPI Interrupt Control Register (0x301718) to disable all the SPI interrupt requests, before disabling the SPI circuit (before setting ENA (D0/0x301708) to 0).
 - * **ENA**: SPI Enable Bit in the SPI Control Register 1 (D0/0x301708)

V.3 Direction Control Serial Interface (DCSIO)

V.3.1 Outline of DCSIO

The S1C33E08 has a built-in DCSIO module that is composed of two bidirectional single-wire serial buses. It features a simple structure for high-speed and efficient data transfer. By using one channel for clock output, an I²C interface can be realized with software control.

The following shows its features:

- Maximum transfer rate: 400 kbps (when a 48-MHz system clock is used)
- Data width: 8 bits
- Programmable bit-rate configuration
- Supports clock stretch (software control wait cycle insertion).
Clock can be controlled as a bit stream.
- Data transfer direction (MSB first/LSB first) is selectable.
- Monitors both lines even in output mode and records output data as received data.
- Supports two interrupt sources (receive data full and transmit data empty).
- Supports single master mode only.
Any transfer protocol may be used without constraints.
Multiple slave devices are connectable to the serial bus (no master device is connectable).

Figure V.3.1.1 shows the structure of the DCSIO module.

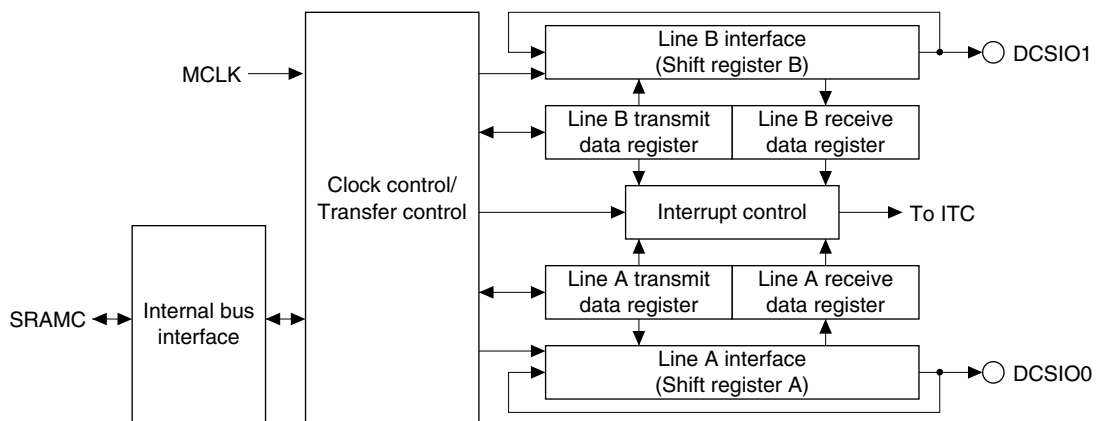


Figure V.3.1.1 Structure of DCSIO Module

V.3.2 I/O Pins of DCSIO

Table V.3.2.1 lists the I/O pins used by the DCSIO module.

Table V.3.2.1 DCSIO Pin Configuration

Pin name	I/O	Function
DCSIO0	I/O	Line A input/output (data input/output or clock output)
DCSIO1	I/O	Line B input/output (data input/output or clock output)

DCSIO0 pin

This is the input/output pin of the Line A serial bus.

DCSIO1 pin

This is the input/output pin of the Line B serial bus.

Note: The DCSIO input/output pins are shared with general-purpose I/O ports or other peripheral circuit inputs/outputs, so that functionality in the initial state is set to other than the DCSIO input/output. Before the DCSIO input/output signals assigned to these pins can be used, the function of these pins must be switched for the DCSIO input/output by setting the corresponding Port Function Select Registers.

For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

V.3.3 DCSIO Operating Clock

The DCSIO module use the DCSIO_CLK clock (= MCLK) generated by the CMU as the operating clock. The transfer clock is generated in the DCSIO module by dividing DCSIO_CLK.

Controlling the supply of the operating clock

DCSIO_CLK is supplied to the DCSIO module with default settings. It can be turned off using DCSIO_CKE (D10/0x301B04) to reduce the amount of power consumed on the chip if the DCSIO module is not used.

* **DCSIO_CKE**: DCSIO Clock Control Bit in the Gated Clock Control Register 1 (D10/0x301B04)

Setting DCSIO_CKE (D10/0x301B04) to 0 (1 by default) turns off the clock supply to the DCSIO module. When the clock supply is turned off, the DCSIO module control registers cannot be accessed.

For details on how to set and control the clock, refer to Section III.1, “Clock Management Unit (CMU).”

Note: The Gated Clock Control Register 1 (0x301B04) is write-protected. Write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless rewriting said registers.

Clock state in standby mode

The clock supply to the DCSIO module stops depending on type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

SLEEP mode: The operating clock supply stops.

Therefore, the DCSIO module also stops operating in SLEEP mode.

V.3.4 Setting DCSIO Module

When performing data transfers via the DCSIO bus, the following settings must be made before data transfer is actually begun:

1. Setting input/output pins
2. Setting base line and transfer rate
3. Setting data transfer direction (MSB first/LSB first)
4. Setting interrupts and IDMA/HSDMA

The following explains the content of each setting. For details on interrupt/DMA settings, refer to Section V.3.6, “DCSIO Interrupts and DMA.”

Note: Always make sure the DCSIO module is inactive (DCSIOEN (D0/0x301800) = 0) before these settings are made. A change of settings during operation may cause a malfunction.

* **DCSIOEN:** DCSIO Enable Bit in the DCSIO Control Register (D0/0x301800)

Setting input/output pins

The DCSIO0 and DCSIO1 pins are used for DCSIO. Configure the Port Function Select Registers to enable these pin functions. For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Set Line A and Line B in output mode using DIRA[1:0] (D[3:2]/0x30181C) and DIRB[1:0] (D[1:0]/0x30181C), respectively. As shown in Table V.3.4.1, either open-drain or push-pull is selectable for output.

* **DIRA[1:0]:** Line A Direction Select Bits in the DCSIO Port Direction Control Register (D[3:2]/0x30181C)

* **DIRB[1:0]:** Line B Direction Select Bits in the DCSIO Port Direction Control Register (D[1:0]/0x30181C)

Table V.3.4.1 Selecting Input/Output Mode

DIRA1/DIRB1	DIRA0/DIRB0	Input/output mode
1	*	Input only
0	1	Input/push-pull output
0	0	Input/open-drain output

(Default: 0b00 = Input/open-drain output)

Setting base line and transfer rate

First, select either Line A (DCSIO0) or Line B (DCSIO1) as the base line (normally used for a clock output) using BASESEL (D2/0x301800). Set BASESEL (D2/0x301800) to 1 when using Line A as the base line, or set it to 0 when using Line B as the base line.

* **BASESEL:** Base Line Select Bit in the DCSIO Control Register (D2/0x301800)

Next, configure the transfer rate of the base line using DIVRT[7:0] (D[15:8]/0x301800).

* **DIVRT[7:0]:** DCSIO System Clock Baud Rate Setup Bits in the DCSIO Control Register (D[15:8]/0x301800)

$$\text{DIVRT} = \frac{f_{\text{DCSIO_CLK}}}{2 \times \text{bps}}$$

DIVRT: DIVRT[7:0] (D[15:8]/0x301800) set value (cannot be set to less than two)

$f_{\text{DCSIO_CLK}}$: DCSIO input clock frequency (= MCLK Hz)

bps: Transfer rate (bits/second)

For example, to configure the transfer rate to 400 kbps when the CPU runs with a 48 MHz clock, set DIVRT[7:0] (D[15:8]/0x301800) to 60 (0x3C).

The transfer rate of the other line (non-base line) can be selected as shown in Table V.3.4.2 using ADVRATE (D3/0x301800).

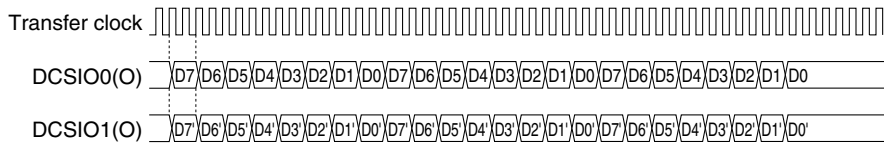
* **ADVRATE:** Non-Base Line Transfer Rate Setup Bit in the DCSIO Control Register (D3/0x301800)

Table V.3.4.2 Transfer Rate of Non-Base Line

ADVRATE	Transfer rate
1	Base line × 1/8
0	= Base line

Figure V.3.4.1 shows the data formats according to the ADVRATE (D3/0x301800) set value.

ADVRATE = 0 (= Base line), BASESEL = 0 (Line B)



ADVRATE = 1 (Base line × 1/8), BASESEL = 0 (Line B)

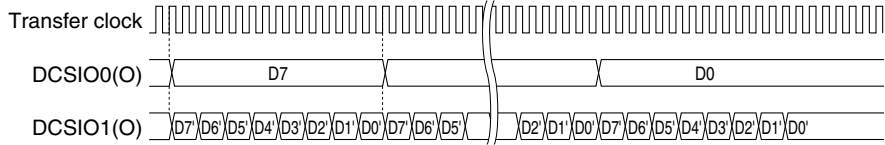


Figure V.3.4.1 DCSIO Data Formats (MSB First)

When ADVRATE (D3/0x301800) is set to 1 (Base line × 1/8), the base line transmits the same 8-bit data eight times while the non-base line performs one 8-bit data transfer. Furthermore, the TXBE (D1/0x301818) and RXBF (D2/0x301818) flags (see Section V.3.5) will be active depending on the non-base line condition only.

Setting data transfer direction (MSB first/LSB first)

Use LSBFIRST (D1/0x301800) to select whether data is input/output from the MSB first or the LSB first.

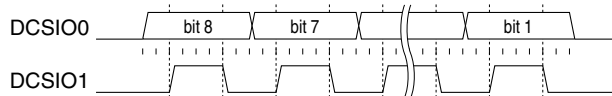
When LSBFIRST (D1/0x301800) = 0 (default), MSB first is selected and when LSBFIRST (D1/0x301800) is set to 1, LSB first is selected.

* **LSBFIRST**: LSB First Select Bit in the DCSIO Control Register (D1/0x301800)

Example of transmit/receive bit configurations

The following shows example of transmit/receive bit configurations.

Example 1



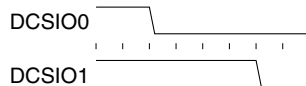
Transmission

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	1	0	0	bit[8:1]	bit[8:1]
DCSIO1	00	(BL × 1/8)	(B)	(MSB)	0x3C	—

Reception

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	1	0	0	0xFF	bit[8:1]
DCSIO1	00	(BL × 1/8)	(B)	(MSB)	0x3C	—

Example 2



Transmission

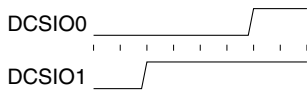
Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	0	0	0	0xC0	0xF0
DCSIO1	00	(BL)	(B)	(MSB)	0xFC	—

Reception

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	0	0	0	0xFF	0xF0
DCSIO1	00	(BL)	(B)	(MSB)	0xFC	—

V PERIPHERAL MODULES 3 (INTERFACE): DIRECTION CONTROL SERIAL INTERFACE (DCSIO)

Example 3



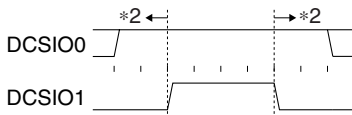
Transmission

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	0	0	0	0x03	0x0F or 0x07 *1
DCSIO1	00	(BL)	(B)	(MSB)	0x3F	–

Reception

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	0	0	0	0xFF	0x0F or 0x07 *1
DCSIO1	00	(BL)	(B)	(MSB)	0x3F	–

Example 4



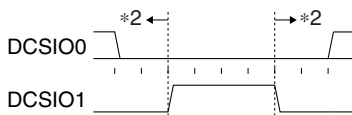
Transmission

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	0	0	0	0xFF	0xFF or 0x3C *2
DCSIO1	00	(BL)	(B)	(MSB)	0x3C	–

Reception

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	0	0	0	0xFF	0xFF or 0x3C *2
DCSIO1	00	(BL)	(B)	(MSB)	0x3C	–

Example 5



Transmission

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	0	0	0	0x00	0x00
DCSIO1	00	(BL)	(B)	(MSB)	0x3C	–

Reception

Pin	DIRA/DIRB	ADVRATE	BASESEL	LSBFIRST	TXD	RXD
DCSIO0	00	0	0	0	0xFF	0x00 or 0xC3 *2
DCSIO1	00	(BL)	(B)	(MSB)	0x3C	–

*1 Note that the open-drain output signals may rise very slowly when they change from 0 to 1, and the rising waveform depends on the pull-up resistor.

*2 Note that the receive data and high-level open-drain output signals outside the range indicated by the dashed lines are controlled by external devices.

V.3.5 Control of Data Transfer

Preparation for data transfer

Perform the following procedure before starting data transfer:

1. Set up the DCSIO conditions as described in the previous section.
2. Set up the interrupt and DMA conditions using the ITC registers and the DCSIO interrupt control register (explained later). When using the DCSIO interrupt, the cause of DCSIO interrupt flag in the ITC must be cleared before enabling the interrupt.
3. Write 1 to the DCSIOEN (D0/0x301800) to turn the DCSIO circuit on.
The DCSIO circuit starts frequency division of the source clock.

* **DCSIOEN**: DCSIO Enable Bit in the DCSIO Control Register (D0/0x301800)

The following explains how to control data transfer assuming that Line A is used for data transfer, Line B is used for clock output, and the transfer rate of Line A is set to Line B (base line) \times 1/8.

Data transmission

Write a transmit data to TXDA[7:0] (D[23:16]/0x301804) and a bit stream of the clock (e.g., 00001111) to TXDB[7:0] (D[7:0]/0x301804). Be sure to write these data simultaneously as these bits are assigned in the same register.

* **TXDA[7:0]**: Line A Transmit Data Bits in the DCSIO Data Load Register (D[23:16]/0x301804)

* **TXDB[7:0]**: Line B Transmit Data Bits in the DCSIO Data Load Register (D[7:0]/0x301804)

DCSIO starts data transmission by writing data to the DCSIO Data Load Register (D[23:16]/0x301804).

The DCSIO circuit loads the data written to the register into the respective shift registers. The data bits in the shift registers are shifted one by one at the rising edge of the transfer clock and are output from the DCSIO0 and DCSIO1 pins. Data is output from the MSB or the LSB according to the LSBFIRST (D1/0x301800) set value.

* **LSBFIRST**: LSB First Select Bit in the DCSIO Control Register (D1/0x301800)

The DCSIO circuit provides TXBE (D1/0x301818) to indicate the DCSIO Data Load Register (0x301804) status. This flag is reset to 0 (not empty) when data is written to the DCSIO Data Load Register (0x301804) and is set to 1 (empty) when the written data is loaded into the shift register. An interrupt can be generated simultaneously with this flag set to 1. Check to see if the TXBE (D1/0x301818) is set to 1 by polling or using this interrupt before the next transmit data can be written to the DCSIO Data Load Register (0x301804).

* **TXBE**: Transmit Data Empty Flag in the DCSIO Status Register (D1/0x301818)

The DCSIO circuit continues data output until the 8-bit data has been transmitted.

ADVRATE (D3/0x301800) is set to 1 (Base line \times 1/8) in this example, therefore, Line B transmits the contents of TXDB[7:0] (D[7:0]/0x301804) 8 times while Line A transmits an 8-bit data.

If the next transmit data exists in the DCSIO Data Load Register (0x301804) when a data transfer has finished, the DCSIO circuit repeats the same transmit operation as above.

If no data exists, data transfer is terminated. The DCSIO0 and DCSIO1 pins maintain the status of the last bit transmitted. It continues until the next data transfer is started, 0 is written to DCSIOEN (D0/0x301800), or the S1C33E08 is reset.

The transmitter/receiver status sets/resets BUSY (D0/0x301818). BUSY (D0/0x301818) is set to 1 when data is being transferred and reset to 0 upon completion of a data transfer. Use this flag to check if a data transfer has completed.

* **BUSY**: DCSIO Busy Flag in the DCSIO Status Register (D0/0x301818)

V PERIPHERAL MODULES 3 (INTERFACE): DIRECTION CONTROL SERIAL INTERFACE (DCSIO)

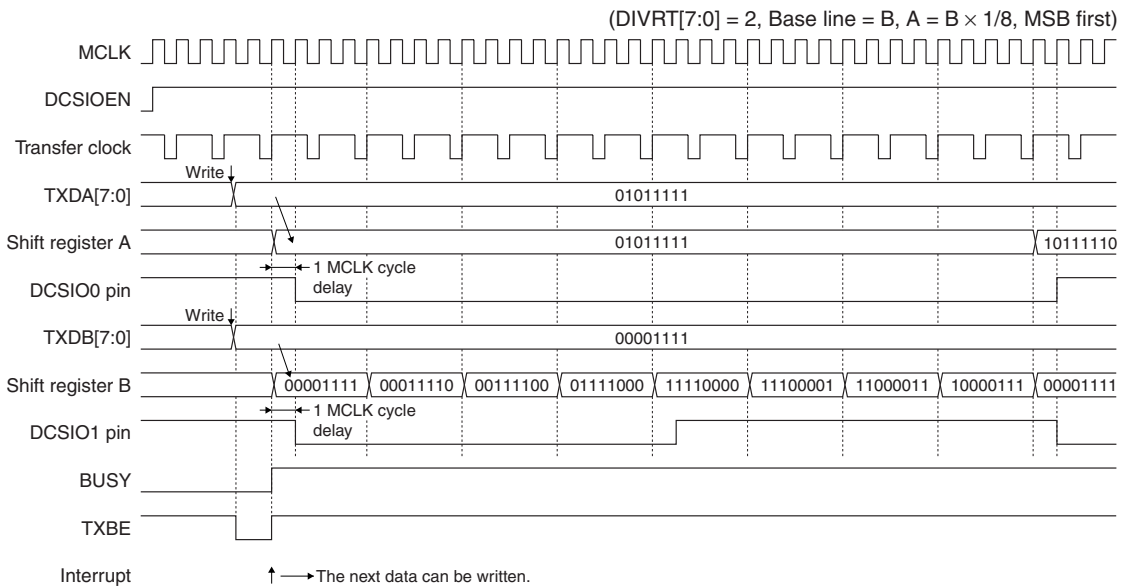


Figure V.3.5.1 Data Transmit Timing Chart (at start of transmission)

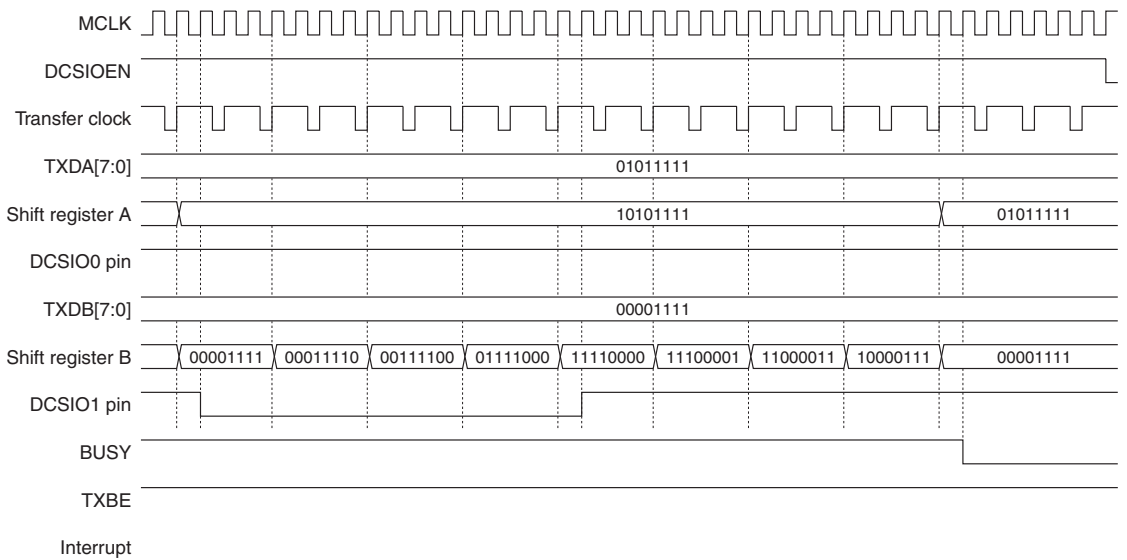


Figure V.3.5.2 Data Transmit Timing Chart (at end of transmission)

Data reception

The DCSIO module monitors both lines and records the line status as receive data regardless of how DIRA (D[3:2]/0x30181C) and DIRB (D[1:0]/0x30181C) are configured. A receive operation starts at the same time a data transmission starts, so writing to the DCSIO Data Load Register (0x301804) is also used as a trigger.

The data bits are fetched in the shift register one by one at the rising edge of the transfer clock from the MSB or LSB according to the LSBFIRST (D1/0x301800) set value.

When an 8-bit data is received in the shift register, the received data is loaded to RXDA[7:0] (D[23:16]/0x301808) in Line A or RXDB[7:0] (D[7:0]/0x301808) in Line B.

- * **RXDA[7:0]**: Line A Receive Data Bits in the DCSIO Receive Data Register (D[23:16]/0x301808)
- * **RXDB[7:0]**: Line B Receive Data Bits in the DCSIO Receive Data Register (D[7:0]/0x301808)

At the same time, RXBF (D2/0x301818) is set to 1 (data full) to indicate that the receive data can be read from the DCSIO Receive Data Register (0x301808) and a data receive interrupt can be generated.

- * **RXBF**: Receive Data Full Flag in the DCSIO Status Register (D2/0x301818)

Check to see if the RXBF (D2/0x301818) is set to 1 by polling or using the interrupt and read data from the DCSIO Receive Data Register (0x301808). When data is read from the DCSIO Receive Data Register (0x301808), RXBF (D2/0x301818) is cleared to 0.

To receive data successively, write 0xFF to TXDA[7:0] (D[23:16]/0x301804) every time an 8-bit data is received. The DCSIO circuit continues the receive operation when data has been written to the DCSIO Data Load Register (0x301804).

BUSY (D0/0x301818) is set to 1 when data is being received.

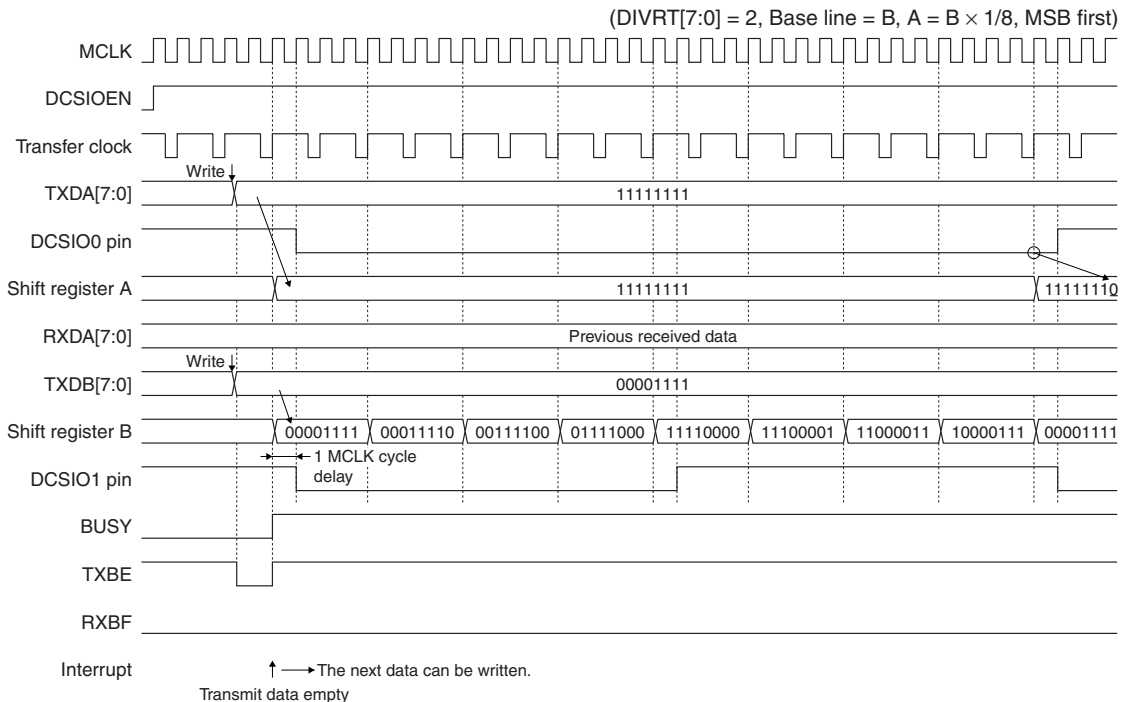


Figure V.3.5.3 Data Receive Timing Chart (at start of reception)

V PERIPHERAL MODULES 3 (INTERFACE): DIRECTION CONTROL SERIAL INTERFACE (DCSIO)

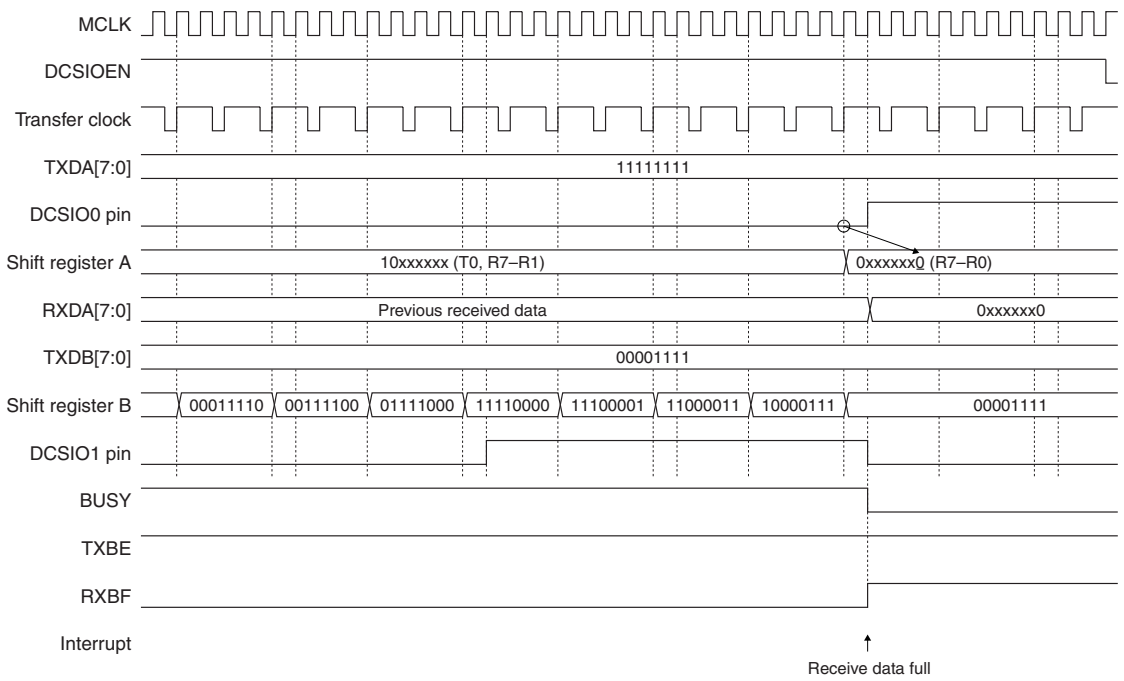


Figure V.3.5.4 Data Receive Timing Chart (at end of reception)

Terminating data transfer

After all data has been transmitted/received, write 0 to DCSIOEN (D0/0x301800) to turn the DCSIO circuit off.

V.3.6 DCSIO Interrupts and DMA

Causes of DCSIO interrupt

The DCSIO module can generate the following two types of interrupts:

- Transmit data empty
- Receive data full

This interrupt request circuit is configured as Figure V.3.6.1 and it allows selection of one or more causes of interrupt.

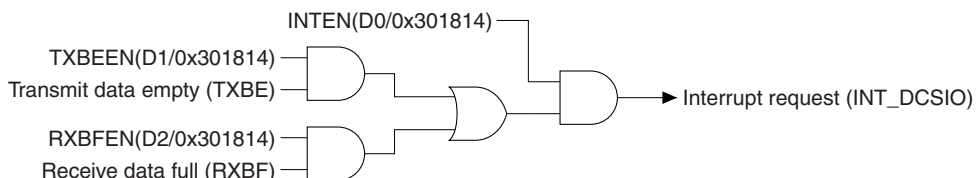


Figure V.3.6.1 DCSIO Interrupt Request Circuit

To output the DCSIO interrupt requests, enable interrupts of the causes described below and set INTEN (D0/0x301814) to 1. When INTEN (D0/0x301814) is set to 0, no DCSIO interrupt request is output.

- * **INTEN**: DCSIO Interrupt Request Enable Bit in the DCSIO Interrupt Control Register (D0/0x301814)

Transmit data empty

This cause of interrupt occurs when the transmit data set in the DCSIO Data Load Register (0x301804) is transferred to the shift register, in which case TXBE (D1/0x301818) is set to 1. Set TXBEEN (D1/0x301814) to 1 to output an interrupt request by this cause of interrupt.

- * **TXBE**: Transmit Data Empty Flag in the DCSIO Status Register (D1/0x301818)
- * **TXBEEN**: Transmit Data Empty Interrupt Enable Bit in the DCSIO Interrupt Control Register (D1/0x301814)

Receive data full

This cause of interrupt occurs when the data received in the shift register is loaded into the DCSIO Receive Data Register (0x301808), in which case RXBF (D2/0x301818) is set to 1. Set RXBFEN (D2/0x301814) to 1 to output an interrupt request by this cause of interrupt.

- * **RXBF**: Receive Data Full Flag in the DCSIO Status Register (D2/0x301818)
- * **RXBFEN**: Receive Data Full Interrupt Enable Bit in the DCSIO Interrupt Control Register (D2/0x301814)

The DCSIO interrupt request is sent to the ITC as the port 11 input interrupt (FPT11) signal and it sets the cause-of-interrupt flag FP11 (D3/0x3002A9) in the ITC to 1. However, INT_DCSIO must be selected for the port 11 input interrupt.

Control registers of the interrupt controller

Table V.3.6.1 shows the interrupt controller's control registers provided for the DCSIO interrupt.

Table V.3.6.1 Control Register of Interrupt Controller

Interrupt	Cause-of-interrupt flag	Interrupt enable register	Interrupt priority register
DCSIO interrupt	FP11(D3/0x3002A9)	EP11(D3/0x3002A6)	PP11L[2:0](D[6:4]/0x3002A1)

When a cause of interrupt described above occurs, the cause-of-interrupt flag is set to 1. If the interrupt enable register bit for that cause of interrupt has been set to 1, an interrupt request is generated.

Interrupts can be disabled by leaving the interrupt enable register bit for that cause of interrupt set to 0. The cause-of-interrupt 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).

V PERIPHERAL MODULES 3 (INTERFACE): DIRECTION CONTROL SERIAL INTERFACE (DCSIO)

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 Section III.2, "Interrupt Controller (ITC)."

Note: The DCSIO interrupt request signal is input to the port 11 input interrupt (FPT11) system. The port 11 input interrupt circuit is configured by selecting a port (signal) to be used for generating an interrupt from P93, INT_DCSIO, P83, and P73. When using the DCSIO interrupt, set SPTB[1:0] (D[7:6]/0x3003C4) to 10 to select INT_DCSIO. This setting enables the INT_DCSIO signal to be sent to the ITC as the port 11 input interrupt signal. Furthermore, SEPTB (D3/0x3003C7) for selecting a trigger mode of the FPT11 input signal should be set to 0 (level) and SPPTB (D3/0x3003C6) for selecting a polarity of the FPT11 input signal should be set to 1 (high level).

* **SPTB[1:0]:** FPT11 Interrupt Input Port Select Bits in the Port Input Interrupt Select Register 3 (D[7:6]/0x3003C4)

* **SPPTB:** FPT11 Input Polarity Select Bit in the Port Input Interrupt Polarity Select Register 2 (D3/0x3003C6)

* **SEPTB:** FPT11 Edge/Level Select Bit in the Port Input Interrupt Edge/Level Select Register 2 (D3/0x3003C7)

Intelligent DMA

The DCSIO interrupt request can be used to invoke intelligent DMA (IDMA). This enables data transfer between memory and the transmit/receive data registers to be performed using DMA.

The IDMA channel numbers set for the cause of DCSIO interrupt is 0x29.

The IDMA request and enable bits shown in Table V.3.6.2 must be set to 1 for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

Table V.3.6.2 Control Bits for IDMA Transfer

Interrupt	IDMA request bit	IDMA enable bit
DCSIO interrupt	RP11(D3/0x3002AC)	DEP11(D3/0x3002AE)

If a cause of interrupt 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 DMA transfer performed.

For details on DMA transfer and how to control interrupts upon completion of DMA transfer, refer to Section II.2, "Intelligent DMA (IDMA)."

High-speed DMA

The DCSIO interrupt (FPT11 interrupt) can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit corresponding to the DCSIO interrupt:

Table V.3.6.3 HSDMA Trigger Set-up Bits

Interrupt	HSDMA Ch.	Trigger set-up bits
DCSIO interrupt	3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2-3 Trigger Set-up Register (0x300299)

For HSDMA to be invoked by a DCSIO interrupt (FPT11 interrupt) request, the trigger set-up bits for HSDMA Ch.3 should be set to "1101."

The HSDMA channel is invoked through generation of the cause of interrupt.

For details on HSDMA transfer, refer to Section II.1, "High-Speed DMA (HSDMA)."

Trap vectors

The default trap-vector address of the cause of DCSIO interrupt is 0xC0015C.

The base address of the trap table can be changed using the TTBR register.

V.3.7 Details of Control Registers

Table V.3.7.1 List of DCSIO Registers

Address	Register name	Size	Function
0x00301800	DCSIO Control Register (pDCSIO_CTL)	32	Sets DCSIO transfer conditions.
0x00301804	DCSIO Data Load Register (pDCSIO_LOAD)	32	Transmit data
0x00301808	DCSIO Receive Data Register (pDCSIO_RCV)	32	Receive data
0x00301814	DCSIO Interrupt Control Register (pDCSIO_INT)	32	Controls DCSIO interrupts.
0x00301818	DCSIO Status Register (pDCSIO_STAT)	32	DCSIO status
0x0030181C	DCSIO Port Direction Control Register (pDCSIO_DIR)	32	Controls data input/output direction.

The following describes each DCSIO control register.

The DCSIO control registers are mapped in the 32-bit device area from 0x301800 to 0x30181C, and can be accessed in units of words, half-words, or bytes except for 0x301804.

- Notes:**
- The DCSIO Data Load Register (0x301804) allow accessing in word size only. Do not read/write the registers in half-word or byte size.
 - When setting the DCSIO control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x301800: DCSIO Control Register (pDCSIO_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
DCSIO control register (pDCSIO_CTL)	00301800 (W)	D31-16	–	reserved		–	–	0 when being read.		
		D15	DIVRT7	DCSIO system clock baud rate setup (DIVRT must be 2 or more.)	Baud rate = $f_{MCLK}/(DIVRT \times 2)$	0	R/W			
		D14	DIVRT6			0				
		D13	DIVRT5			0				
		D12	DIVRT4			0				
		D11	DIVRT3			0				
		D10	DIVRT2			0				
		D9	DIVRT1			0				
		D8	DIVRT0			0				
		D7-4	–	reserved		–	–	0 when being read.		
		D3	ADVRATE	Non-base line transfer rate setup	1	Base × 1/8	0	= Base line	0	R/W
		D2	BASESEL	Base line selection	1	A	0	B	0	R/W
D1	LSBFIRST	LSB first selection	1	LSB first	0	MSB first	0	R/W		
D0	DCSIOEN	DCSIO enable	1	Enabled	0	Disabled	0	R/W		

D[31:16] Reserved

D[15:8] DIVRT[7:0]: DCSIO System Clock Baud Rate Setup Bits

Sets the transfer rate for the base line. (Default: 0x0)

$$DIVRT = \frac{f_{DCSIO_CLK}}{2 \times bps}$$

DIVRT: DIVRT[7:0] set value (cannot be set to less than two)

f_{DCSIO_CLK}: DCSIO input clock frequency (= MCLK Hz)

bps: Transfer rate (bits/second)

D[7:4] Reserved

D3 ADVRATE: Non-Base Line Transfer Rate Setup Bit

Sets the transfer rate for the non-base line.

1 (R/W): Base line × 1/8

0 (R/W): Same as base line (default)

When ADVRATE is set to 1 (Base line × 1/8), the base line transmits the same 8-bit data 8 times while the non-base line performs an 8-bit data transfer.

D2 BASESEL: Base Line Select Bit

Selects the base line.

1 (R/W): Line A (DCSIO0)

0 (R/W): Line B (DCSIO1) (default)

Normally, select the line to be used for clock output as the base line.

D1 LSBFIRST: LSB First Select Bit

Selects either MSB first or LSB first as the data direction.

1 (R/W): LSB first

0 (R/W): MSB first (default)

This setting applies to both Line A and Line B.

D0 DCSIOEN: DCSIO Enable Bit

Enables/disables operation of the DCSIO module.

1 (R/W): Enable (on)

0 (R/W): Disable (off) (default)

When DCSIOEN is set to 1, the DCSIO module starts operating and data transfer is enabled.

When DCSIOEN is set to 0, the DCSIO module goes off.

Make sure that this bit is 0 before setting up data transfer conditions using the DCSIO registers.

0x301804: DCSIO Data Load Register (pDCSIO_LOAD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
DCSIO data load register (pDCSIO_LOAD)	00301804 (W)	D31-24	–	reserved	–	–	–	0 when being read.
		D23	TXDA7	Line A transmit data	0x0 to 0xFF	1	R/W	
		D22	TXDA6					
		D21	TXDA5					
		D20	TXDA4					
		D19	TXDA3					
		D18	TXDA2					
		D17	TXDA1					
		D16	TXDA0					
		D15-8	–	reserved	–	–	–	0 when being read.
		D7	TXDB7	Line B transmit data	0x0 to 0xFF	1	R/W	
		D6	TXDB6					
		D5	TXDB5					
		D4	TXDB4					
		D3	TXDB3					
		D2	TXDB2					
		D1	TXDB1					
		D0	TXDB0					

Note: This register allow accessing in word size only. Do not read/write the registers in half-word or byte size.

Sets transmit data.

The data transmission begins by writing data to this register.

TXBE (D1/0x301818) is set to 1 (empty) when the Line A and Line B data written to this register are transferred to the respective shift registers. A cause of transmit data empty interrupt is simultaneously generated. The next transmit data can be written to the register at any time thereafter, even when the DCSIO is sending data.

The serial-converted data is output from the DCSIOx pin beginning with the MSB or LSB selected with LSBFIRST (D1/0x301800), in which the bits set to 1 are output as high-level signals and those set to 0 output as low-level signals.

Writing to this register also starts a data receive operation (transmission and reception are performed simultaneously using the same shift register). The transmit data for the receive line should be set to 0xFF.

D[31:24] Reserved**D[23:16] TXDA[7:0]: Line A Transmit Data Bits**

Sets transmit data for Line A. (Default: 0xFF)

D[15:8] Reserved**D[7:0] TXDB[7:0]: Line B Transmit Data Bits**

Sets transmit data for Line B. (Default: 0xFF)

0x301808: DCSIO Receive Data Register (pDCSIO_RCV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
DCSIO receive data register (pDCSIO_RCV)	00301808 (W)	D31-24	–	reserved		–	–	0 when being read.
		D23	RXDA7	Line A receive data	0x0 to 0xFF	1	R	
		D22	RXDA6			1		
		D21	RXDA5			1		
		D20	RXDA4			1		
		D19	RXDA3			1		
		D18	RXDA2			1		
		D17	RXDA1			1		
		D16	RXDA0			1		
		D15-8	–			reserved		
		D7	RXDB7	Line B receive data	0x0 to 0xFF	1	R	
		D6	RXDB6			1		
		D5	RXDB5			1		
		D4	RXDB4			1		
		D3	RXDB3			1		
		D2	RXDB2			1		
		D1	RXDB1			1		
D0	RXDB0	1						

Stores received data.

When a receive operation is completed and the Line A and Line B data received in the shift registers are loaded to this register, RXBF (D2/0x301818) is set to 1 (data full). At the same time, a cause of receive data full interrupt occurs. 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.

The serial data input from the DCSIOx pin is converted into parallel data, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in this register.

This register is a read-only register, so no data can be written to it.

D[31:24] Reserved

D[23:16] RXDA[7:0]: Line A Receive Data Bits

Line A receive data is stored. (Default: 0xFF)

D[15:8] Reserved

D[7:0] RXDB[7:0]: Line B Receive Data Bits

Line B receive data is stored. (Default: 0xFF)

0x301814: DCSIO Interrupt Control Register (pDCSIO_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
DCSIO interrupt control register (pDCSIO_INT)	00301814 (W)	D31-3	–	reserved		–	–	0 when being read.
		D2	RXBFEN	Receive data full interrupt enable	1	Enabled	0 Disabled	0 R/W
		D1	TXBEEN	Transmit data empty int. enable	1	Enabled	0 Disabled	0 R/W
		D0	INTEN	DCSIO interrupt request enable	1	Enabled	0 Disabled	0 R/W

D[31:3] Reserved**D2 RXBFEN: Receive Data Full Interrupt Enable Bit**

Enables/disables DCSIO interrupt caused by receive data full.

1 (R/W): Enable

0 (R/W): Disable (default)

When RXBFEN is set to 1, DCSIO (receive data full) interrupt requests to the ITC are enabled. A receive data full interrupt request occurs when the data received in the shift register is loaded to the DCSIO Receive Data Register (0x301808) (receive operation completed). At this time the cause-of-interrupt flag FP11 (D3/0x3002A9) in the ITC is set to 1 if both RXBFEN and INTEN (D0) have been set to 1 (enabled).

When RXBFEN is set to 0, DCSIO interrupts caused by receive data full are not generated.

D1 TXBEEN: Transmit Data Empty Interrupt Enable Bit

Enables/disables DCSIO interrupt caused by transmit data empty.

1 (R/W): Enable

0 (R/W): Disable (default)

When TXBEEN is set to 1, DCSIO (transmit data empty) interrupt requests to the ITC are enabled. A transmit data empty interrupt request occurs when the data written to the DCSIO Data Load Register (0x301804) is transferred to the shift register (transmit operation started). At this time the cause-of-interrupt flag FP11 (D3/0x3002A9) in the ITC is set to 1 if both TXBEEN and INTEN (D0) have been set to 1 (enabled).

When TXBEEN is set to 0, DCSIO interrupts caused by transmit data empty are not generated.

D0 INTEN: DCSIO Interrupt Request Enable Bit

Enables/disables DCSIO interrupt requests to the ITC.

1 (R/W): Enable

0 (R/W): Disable (default)

When INTEN is set to 1, DCSIO interrupt requests to the ITC are enabled. An interrupt request is generated and the cause-of-interrupt flag FP11 (D3/0x3002A9) is set to 1 when an enabled cause of DCSIO interrupt occurs.

When INTEN is set to 0, a DCSIO interrupt request to the ITC is not generated even if the DCSIO interrupt for each cause is enabled.

0x301818: DCSIO Status Register (pDCSIO_STAT)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
DCSIO status register (pDCSIO_STAT)	00301818 (W)	D31-3	–	reserved	–			–	–	0 when being read.	
		D2	RXBF	Receive data full flag	1	Full	0	Not full	0	R	
		D1	TXBE	Transmit data empty flag	1	Empty	0	Not empty	0	R	
		D0	BUSY	DCSIO busy flag	1	Busy	0	Idle	0	R	

D[31:3] Reserved

D2 RXBF: Receive Data Full Flag

Indicates the DCSIO Receive Data Register (0x301808) status.

- 1 (R): Data full
- 0 (R): Not full (default)

RXBF is set to 1 when the data received in the shift register is loaded to the DCSIO Receive Data Register (0x301808) (receive operation completed), indicating that the received data can be read out. This bit is reset to 0 when the data is read out.

D1 TXBE: Transfer Data Empty Flag

Indicates the DCSIO Data Load Register (0x301804) status.

- 1 (R): Empty
- 0 (R): Not empty (default)

TXBE is cleared to 0 when transmit data is written to the DCSIO Data Load Register (0x301804) 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 DCSIO Data Load Register (0x301804) when this bit = 1.

D0 BUSY: DCSIO Busy Flag

Indicates the DCSIO transmit/receive operation status.

- 1 (R): Busy
- 0 (R): Idle (default)

BUSY is set to 1 when the DCSIO starts data transmission/reception and stays 1 while data transmission/reception is in progress. BUSY is cleared to 0 upon completion of transmit/receive operation.

0x30181C: DCSIO Port Direction Control Register (pDCSIO_DIR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
DCSIO port direction control register (pDCSIO_DIR)	0030181C (W)	D31-4	–	reserved	–	–	–	0 when being read.	
		D3 D2	DIRA1 DIRA0	Line A direction select	DIRA[1:0]	I/O	0	R/W	
					1*	Input only	0		
					01	Input/push-pull output			
		00	Input/open-drain output						
		D1 D0	DIRB1 DIRB0	Line B direction select	DIRB[1:0]	I/O	0	R/W	
1*	Input only				0				
01	Input/push-pull output								
00	Input/open-drain output								

This register is used to select the data input/output direction for Line A and Line B individually as shown in Table V.3.7.2.

Table V.3.7.2 Selecting Data Input/Output Direction

DIRA1/DIRB1	DIRA0/DIRB0	Input/output mode
1	*	Input only
0	1	Input/push-pull output
0	0	Input/open-drain output

(Default: 0b00 = Input/open-drain output)

D[31:4] Reserved

D[3:2] DIRA[1:0]: Line A Direction Select Bits

Selects the data input/output direction for Line A.

D[1:0] DIRB[1:0]: Line B Direction Select Bits

Selects the data input/output direction for Line B.

V.3.8 Precautions

- Be sure to use 32-bit access instructions for reading/writing from/to the DCSIO Data Load Register (0x301804). Transmit data for Line A and Line B must be written simultaneously.
- To prevent malfunctions, write 0x0 to the DCSIO Interrupt Control Register (0x301814) to disable all the DCSIO interrupt requests, before disabling the DCSIO circuit (before setting DCSIOEN (D0/0x301800) to 0).
 - * **DCSIOEN**: DCSIO Enable Bit in the DCSIO Control Register (D0/0x301800)
- When using the DCSIO ports as input/open-drain output ports (default configuration), be sure to enable the internal pull-up resistor (refer to Section III.4.4, “Pin Control Registers”) or to connect an external pull-up resistor to the pin.

V.4 Card Interface (CARD)

V.4.1 Outline of the Card Interface

The Card Interface (CARD) generates control signals for the card interfaces listed below.

- #SMRD and #SMWR signals for SmartMedia (NAND flash)
- #CFCE1 and #CFCE2 signals for CompactFlash
- #IORD, #IOWR, #OE and #WE signals for PC Card

Each device may be located in any of the #CE4, #CE7, #CE9, or #CE11 areas. The data and address signals of each device can be connected directly to the external bus of the SRAMC.

Use general-purpose input/output ports to control the signals specific to each card.

The PC card interface can accommodate up to two channels by locating each channel in different areas.

This interface incorporates an ECC (Error Correction Code) generator for 8- and 16-bit card devices.

V.4.2 Card Interface Pins

The Card Interface has six ports (CARD0 to CARD5) to output each interface signal listed above.

Note, however, that these pins are shared with general-purpose I/O ports or other peripheral circuit input/output pins. In the initial state, these pins are set for functions other than the card interface. Before the pins can be used as CARD0 to CARD5 ports, the pin functions must be switched over by setting the corresponding Port Function Select Register.

For details on how to switch over the pin functions, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Selecting the card interface pin functions

The CARD0 to CARD5 ports each are assigned two card interface signals, either of which can be selected for output from the respective ports. Table V.4.2.1 lists the signals assigned to each port.

Table V.4.2.1 Relationship between Ports and Card Interface Signals

Pin name	Function 0 (default)	Function 1	Function select bit
CARD0	#SMRD	#CFCE1	CARDIO0 (D0/0x300302)
CARD1	#SMWR	#CFCE2	CARDIO1 (D1/0x300302)
CARD2	#IORD	#SMRD	CARDIO2 (D2/0x300302)
CARD3	#IOWR	#SMWR	CARDIO3 (D3/0x300302)
CARD4	#OE	#CFCE1	CARDIO4 (D4/0x300302)
CARD5	#WE	#CFCE2	CARDIO5 (D5/0x300302)

* **CARDIOx**: CARDx Port Function Select Bit in the Card I/F Output Port Configuration Register (Dx/0x300302)

CARDIOx (Dx/0x300302) is initially set to 0, with function 0 selected. To use function 1, set CARDIOx (Dx/0x300302) to 1.

V.4.3 Card Area

V.4.3.1 Selecting the Area

The following #CE areas can be used as memory areas for cards.

#CE4 area: Area 4 (0x100000 to 0x1FFFFFF, 1MB) or Area 14 (0x3000000 to 0x3FFFFFF, 16MB)

#CE7 area: Area 7 (0x400000 to 0x5FFFFFF, 2MB) or Area 19 (0x10000000 to 0x1FFFFFFF, 256MB)

#CE9 area: Area 9 (0x800000 to 0xBFFFFFF, 4MB) or Area 22 (0x80000000 to 0xFFFFFFFF, 2GB)

#CE11 area: Area 11 (0x1000000 to 0x17FFFFFF, 8MB) or Area 12 (0x1800000 to 0x1FFFFFF, 8MB)

One of the four areas above can be selected for each card interface.

The following describes the control bits used to select any area and the bit settings.

SmartMedia (NAND flash)

- * **CARDSMT[1:0]**: SmartMedia/NAND Flash Area Configuration Bits in the Card I/F Area Configuration Register (D[1:0]/0x300300)

CompactFlash

- * **CARDCF[1:0]**: CF Area Configuration Bits in the Card I/F Area Configuration Register (D[3:2]/0x300300)

PC Card 1

- * **CARDPC1[1:0]**: PC Card 1 Area Configuration Bits in the Card I/F Area Configuration Register (D[5:4]/0x300300)

PC Card 2

- * **CARDPC2[1:0]**: PC Card 2 Area Configuration Bits in the Card I/F Area Configuration Register (D[7:6]/0x300300)

Table V.4.3.1.1 Area Selection

Control bit settings	Area
11	#CE11 (area 11, 12)
10	#CE9 (area 9, 22)
01	#CE7 (area 7, 19)
00	#CE4 (area 4, 14)

(Default: 0b00 = #CE4)

V.4.3.2 Setting Area Access Conditions

Depending on the device used, access conditions must be set for the area assigned to the card. Use the SRAMC registers to set these conditions as suitable for the specifications of the connected card. For details of access conditions and the SRAMC registers, see Section II.3, "SRAM Controller (SRAMC)."

V.4.4 Card Interface Control Signals

This section describes the logic used to generate each card interface signal, and shows an example of connecting a card.

V.4.4.1 SmartMedia Interface

Figure V.4.4.1.1 shows the logic used to generate SmartMedia interface signals. Figure V.4.4.1.2 shows an example of connecting the S1C33E08 and a SmartMedia card (NAND flash).

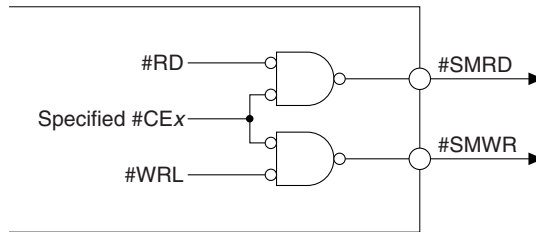


Figure V.4.4.1.1 SmartMedia Interface Signal Generation Circuit

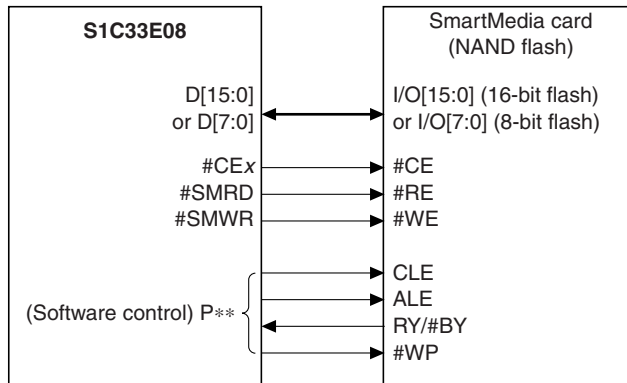


Figure V.4.4.1.2 Example of Connecting a SmartMedia Card

For SmartMedia and CompactFlash, a middleware is available for use in developing S1C33 Family applications. For details of the system and applications that can be implemented, refer to the manual included with the middleware.

The S1C33E08 supports system boot from a NAND Flash. See Appendix for details on booting.

V.4.4.2 CompactFlash Interface

Figure V.4.4.2.1 shows the logic used to generate CompactFlash interface signals. Figure V.4.4.2.2 shows an example of connecting the S1C33E08 and a CompactFlash card.

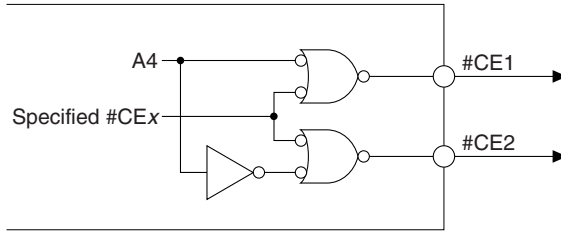


Figure V.4.4.2.1 CompactFlash Interface Signal Generation Circuit

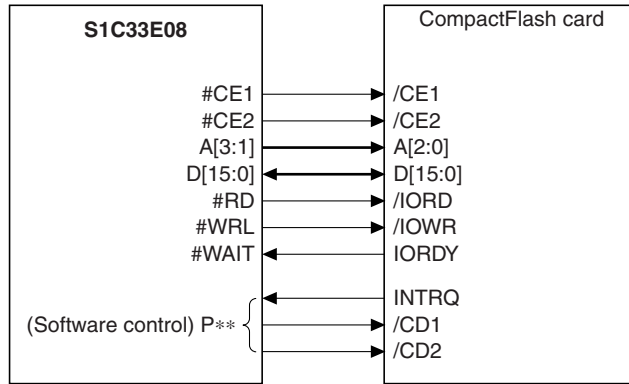


Figure V.4.4.2.2 Example of Connecting a CompactFlash Card

For SmartMedia and CompactFlash, a middleware is available for use in developing S1C33 Family applications. For details of the system and applications that can be implemented, refer to the manual included with the middleware.

V.4.4.3 PC Card Interface

Figure V.4.4.3.1 shows the logic used to generate PC Card interface signals. Figure V.4.4.3.2 shows an example of connecting the S1C33E08 and a PC Card.

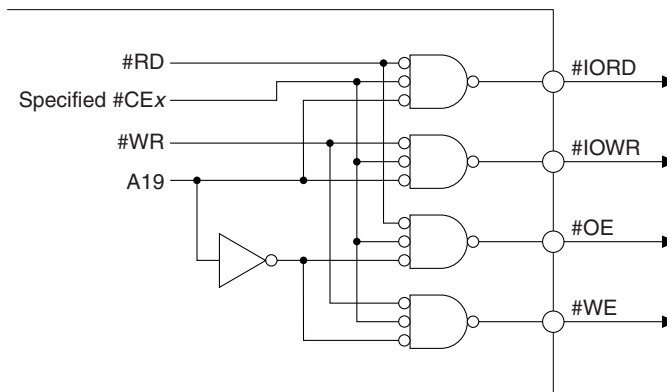
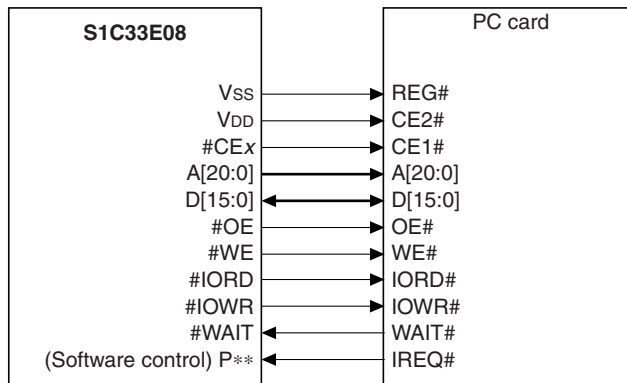


Figure V.4.4.3.1 PC Card Interface Signal Generation Circuit

V PERIPHERAL MODULES 3 (INTERFACE): CARD INTERFACE (CARD)

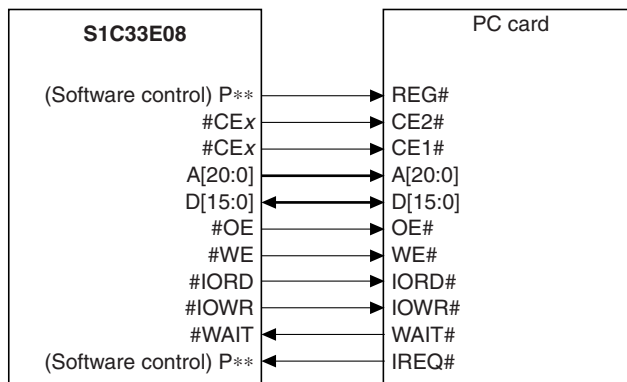
Example connection 1

Area used: Area 7
 Device size: 8 bits (only accessible in units of bytes)
 Attribute memory space: 0x480000 to 0x4FFFFF, 512KB (common memory not used)
 I/O space: 0x400000 to 0x47FFFF, 512KB



Example connection 2

Area used: Area 7
 Device size: 16 bits (only accessible in units of half-words)
 Attribute memory space: 0x480000 to 0x4FFFFF, 512KB (common memory not used)
 I/O space: 0x400000 to 0x47FFFF, 512KB



* Since this example connection only supports access in units of half-words, PC cards with only 16-bit registers can be used. PC cards with 8-bit registers cannot be used.

Figure V.4.4.3.2 Example of Connecting a PC Card

Precautions on using the PC Card interface

- This interface supports 16-bit PC cards, such as ATA (CF), LAN (Ethernet, wireless), and modem that are connected as an I/O card. SRAM cards, etc. are not supported.
- Live or hot-line card insertion and removal are not supported. Power must be turned off before inserting or removing a card. Automatic recognition of cards is also not supported.
- DMA, ZV, and CardBus are also not supported.
- To accommodate differences in power supply voltage between the PC card (5 V or 3.3 V) and the S1C33E08, use a buffer IC (e.g., S1C37120).

V.4.5 Card Interface Operating Clock

The card interface module is clocked by the `CARD_CLK` clock (= `MCLK`) supplied by the CMU. When initially reset, this clock is selected for supply to the card interface. However, when the card interface is idle or not in use, clock supply from the CMU may be turned off to reduce current consumed on the chip. Use `CARD_CKE` (`D4/0x301B04`) of the CMU for this control.

* **CARD_CKE**: CARD I/F Clock Control Bit in the Gated Clock Control Register 1 (`D4/0x301B04`)

Setting `CARD_CKE` (`D4/0x301B04`) to 0 stops clock supply from the CMU to the card interface.

For details of the generation and control of the clock, see Section III.1, “Clock Management Unit (CMU).”

Note: The Gated Clock Control Register 1 (`0x301B04`) is write-protected. Write protection of this and other CMU control registers at addresses `0x301B00` to `0x301B14` to be rewritten must be removed by writing `0x96` to the Clock Control Protect Register (`0x301B24`). Since unnecessary rewrites to addresses `0x301B00` to `0x301B14` could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (`0x301B24`) is other than `0x96`, unless re-writing said registers.

V.4.6 ECC Generator

In order to improve the card 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 card data area.

When reading data from a card, software can compare the ECC data read from the card 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 card, software can write the ECC data generated by the ECC generator to the redundant area of the card.

ECC algorithm

The card data is treated as bit streams, and ECC parity data is generated as follows:

'^' denotes the XOR (exclusive OR) operation.

ECC for 8-bit card 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 card 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 V.4.6.1 shows a card data read flowchart.

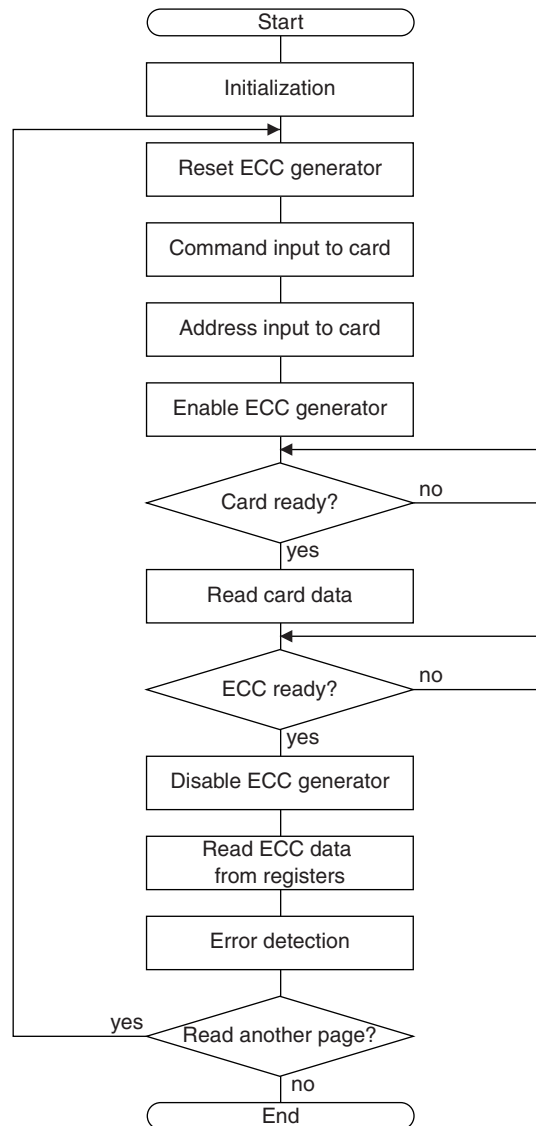


Figure V.4.6.1 Data Read Flowchart

Notes: • ECCRST (D0/0x300311) should be set before using the ECC function.

* **ECCRST**: ECC Circuit Reset Bit in the ECC Reset/Ready Register (D0/0x300311)

- The ECCEN (D0/0x300312) should be set only while reading or writing card data. It should be disabled during command input, address input or status reading.

* **ECCEN**: ECC Circuit Enable Bit in the ECC Enable Register (D0/0x300312)

Data write flowchart

Figure V.4.6.2 shows a card data write flowchart.

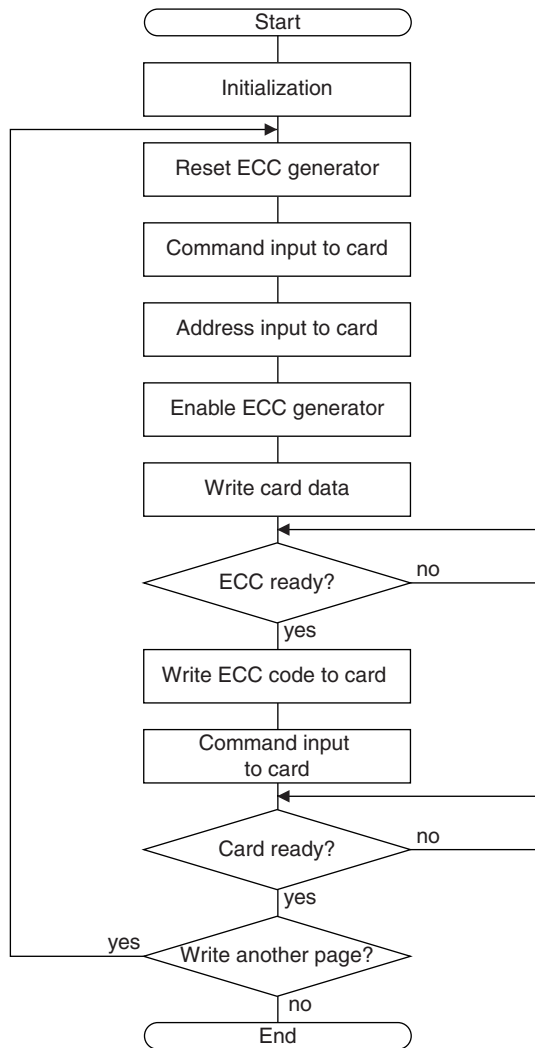


Figure V.4.6.2 Data Write Flowchart

- Notes:**
- ECCRST (D0/0x300311) should be set before using the ECC function.
 - The ECCEN (D0/0x300312) should be set only while reading or writing card data. It should be disabled during command input, address input or status reading.

V.4.7 Details of Control Registers

Table V.4.7.1 Card Interface Control Register List

Address	Register name	Size	Function
0x00300300	Card I/F Area Configuration Register (pCARDSETUP)	8	Allocates card space.
0x00300302	Card I/F Output Port Configuration Register (pCARDFUNCSEL05)	8	Selects the function of signal output ports for card control.
0x00300310	ECC Trigger Area Select Register (pECCTRIGSEL)	8	Selects ECC trigger area.
0x00300311	ECC Reset/Ready Register (pECCRSTRDY)	8	Resets ECC circuits and indicates ECC ready status.
0x00300312	ECC Enable Register (pECCENA)	8	Enables ECC circuits.
0x00300313	ECC Mode Register (pECCMD)	8	Selects device size.
0x00300314	Area 0 ECC Column Parity Data Register (pECC0CP)	8	Area 0 column parity data
0x00300316	Area 0 ECC Line Parity Register 0 (pECC0LPL)	8	Area 0 line parity data (low-order byte)
0x00300317	Area 0 ECC Line Parity Register 1 (pECC0LPH)	8	Area 0 line parity data (high-order byte)
0x00300318	Area 1 ECC Column Parity Data Register (pECC1CP)	8	Area 1 column parity data
0x0030031A	Area 1 ECC Line Parity Register 0 (pECC1LPL)	8	Area 1 line parity data (low-order byte)
0x0030031B	Area 1 ECC Line Parity Register 1 (pECC1LPH)	8	Area 1 line parity data (high-order byte)

The following describes each card interface control register.

The card interface control registers are mapped to the 8-bit device area at addresses 0x300300 to 0x30031B, and can be accessed in units of bytes.

Note: When setting the card interface control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x300300: Card I/F Area Configuration Register (pCARDSETUP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Card I/F area configuration register (pCARDSETUP)	00300300 (B)	D7	CARDPC21	PC card 2 area configuration	bit[1:0]	#CE area	0	R/W	
		D6	CARDPC20				0		
		D5	CARDPC11	PC card 1 area configuration	11	#CE11 (Area 11, 12)	0	R/W	
		D4	CARDPC10				0		
		D3	CARDCF1	CF area configuration	10	#CE9 (Area 9, 22)	0	R/W	
		D2	CARDCF0				0		
		D1	CARDSMT1	SmartMedia/NAND flash area configuration	01	#CE7 (Area 7, 19)	0	R/W	
		D0	CARDSMT0				0		
					00	#CE4 (Area 4, 14)	0		

This register sets the area in which to allocate each card space. Table V.4.7.2 lists the settings of each card interface control bit used to select this area.

Table V.4.7.2 Area Selection

Control bit settings	Area
11	#CE11 (area 11, 12)
10	#CE9 (area 9, 22)
01	#CE7 (area 7, 19)
00	#CE4 (area 4, 14)

(Default: 0b00 = #CE4)

D[7:6] CARDPC2[1:0]: PC Card 2 Area Configuration Bits

These bits select the area used for PC Card 2.

D[5:4] CARDPC1[1:0]: PC Card 1 Area Configuration Bits

These bits select the area used for PC Card 1.

D[3:2] CARDCF[1:0]: CF Area Configuration Bits

These bits select the area used for the CompactFlash card.

D[1:0] CARDSMT[1:0]: SmartMedia Area Configuration Bits

These bits select the area used for the SmartMedia card (NAND flash).

0x300302: Card I/F Output Port Configuration Register (pCARDFUNCSEL05)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Card I/F output port configuration register (pCARDFUNCSEL05)	00300302 (B)	D7-6	–	reserved		–	–	–	0 when being read.	
		D5	CARDIO5	CARD5 port function select	1	#CFCE2	0	#WE	0	R/W
		D4	CARDIO4	CARD4 port function select	1	#CFCE1	0	#OE	0	R/W
		D3	CARDIO3	CARD3 port function select	1	#SMWR	0	#IOWR	0	R/W
		D2	CARDIO2	CARD2 port function select	1	#SMRD	0	#IORD	0	R/W
		D1	CARDIO1	CARD1 port function select	1	#CFCE2	0	#SMWR	0	R/W
		D0	CARDIO0	CARD0 port function select	1	#CFCE1	0	#SMRD	0	R/W

Table V.4.7.3 Relationship between Ports and Card Interface Signals

Pin name	Function 0 (default)	Function 1
CARD0	#SMRD	#CFCE1
CARD1	#SMWR	#CFCE2
CARD2	#IORD	#SMRD
CARD3	#IOWR	#SMWR
CARD4	#OE	#CFCE1
CARD5	#WE	#CFCE2

D[7:6] Reserved**D5 CARDIO5: CARD5 Port Function Select Bit**

This bit selects the signal to output from the CARD5 pin.

1 (R/W): #CFCE2

0 (R/W): #WE (default)

D4 CARDIO4: CARD4 Port Function Select Bit

This bit selects the signal to output from the CARD4 pin.

1 (R/W): #CFCE1

0 (R/W): #OE (default)

D3 CARDIO3: CARD3 Port Function Select Bit

This bit selects the signal to output from the CARD3 pin.

1 (R/W): #SMWR

0 (R/W): #IOWR (default)

D2 CARDIO2: CARD2 Port Function Select Bit

This bit selects the signal to output from the CARD2 pin.

1 (R/W): #SMRD

0 (R/W): #IORD (default)

D1 CARDIO1: CARD1 Port Function Select Bit

This bit selects the signal to output from the CARD1 pin.

1 (R/W): #CFCE2

0 (R/W): #SMWR (default)

D0 CARDIO0: CARD0 Port Function Select Bit

This bit selects the signal to output from the CARD0 pin.

1 (R/W): #CFCE1

0 (R/W): #SMRD (default)

0x300310: ECC Trigger Area Select Register (pECCTRIGSEL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
ECC trigger area select register (pECCTRIGSEL)	00300310 (B)	D7-3	-	reserved	-	-	-	0 when being read.	
		D2	ECCSEL2	External memory area select for triggering ECC	ECCSEL[2:0]	#CE area	0	R/W	
		D1	ECCSEL1		111	#CE11	0		
		D0	ECCSEL0		110	#CE10	0		
					101	#CE9			
					100	#CE8			
					011	#CE7			
					010	#CE6			
					001	#CE5			
					000	#CE4			

D[7:3] Reserved

D[2:0] ECCSEL[2:0]: External Memory Area Select Bits for Triggering ECC

Selects a card area in which the ECC function is used.

Table V.4.7.4 Selecting an ECC Trigger Area

ECCSEL2	ECCSEL1	ECCSEL0	Area
1	1	1	#CE11
1	1	0	#CE10
1	0	1	#CE9
1	0	0	#CE8
0	1	1	#CE7
0	1	0	#CE6
0	0	1	#CE5
0	0	0	#CE4

(Default: 0b000 = #CE4)

0x300311: ECC Reset/Ready Register (pECCRSTRDY)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
ECC reset/ready register (pECCRSTRDY)	00300311 (B)	D7-2	–	reserved				–	–	0 when being read.	
		D1	ECCRDY1	Area 1 parity data ready status	1	Ready	0	Busy	0	R	
		D0	ECCRST	ECC circuit reset	1	Reset	0	Invalid	–	W	
			ECCRDY0	Area 0 parity data ready status	1	Ready	0	Busy	0	R	

D[7:2] Reserved**D1 ECCRDY1: Area 1 Parity Data Ready Status Bit**

Indicates whether the parity data for area 1 is ready for reading or not.

1 (R): Ready for read

0 (R): Not ready (default)

When 1 is read from ECCRDY1, the ECC generator has generated the column and line parity data for area 1 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 for area 1.

D0 ECCRST: ECC Circuit Reset Bit (W)**ECCRDY0: Area 0 Parity Data Ready Status Bit (R)**

Resets the ECC generator and indicates whether the parity data for area 0 is ready for reading or not.

1 (W): Reset

0 (W): Invalid

1 (R): Ready for read

0 (R): Not ready (default)

When writing (ECCRST):

Writing 1 to this bit resets the ECC generator. This operation is required in every page before reading or writing data (before setting ECCEN (D0/0x300312) to 1).

When reading (ECCRDY0):

When 1 is read from this bit (ECCRDY0), the ECC generator has generated the column and line parity data for area 0 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 for area 0.

* Area 0 means the first transferred 256-byte (in 8-bit device) or 128-word (in 16-bit device) card data after the ECC circuit is enabled. Area 1 means the second transferred 256-byte (in 8-bit device) or 128-word (in 16-bit device) card data.

0x300312: ECC Enable Register (pECCENA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
ECC enable register (pECCENA)	00300312 (B)	D7-1	–	reserved	–		–	–	0 when being read.
		D0	ECCEN	ECC circuit enable	1	Enabled	0	Disabled	0

D[7:1] Reserved**D0 ECCEN: ECC Circuit Enable Bit**

Enables the ECC generator.

1 (R/W): Enable

0 (R/W): Disable (default)

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.

0x300313: ECC Mode Register (pECCMD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
ECC mode register (pECCMD)	00300313 (B)	D7-1	-	reserved	-		-	-	0 when being read.
		D0	MODE	Card device mode	1	16 bits	0	8 bits	0

D[7:1] Reserved

D0 MODE: Card Device Mode Select Bit

Selects the card device size.

1 (R/W): 16-bit device

0 (R/W): 8-bit device (default)

Set MODE to 1 when using a 16-bit device or set to 0 when using an 8-bit device.

0x300314: Area 0 ECC Column Parity Data Register (pECC0CP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Area 0 ECC column parity data register (pECC0CP)	00300314 (B)	D7	CP05	Area 0 column parity data		0x0 to 0x3F	1	R
		D6	CP04					
		D5	CP03					
		D4	CP02					
		D3	CP01					
		D2	CP00					
		D1	–	Unused bit	–	1	R	1 when being read.
		D0	–	Unused bit	–	1	R	

D[7:2] CP0[5:0]: Area 0 Column Parity Data Bits

The column parity data that is generated by the ECC generator from the 256-byte card data in area 0 is stored in this register.

The data read from this register is valid when ECCRDY0 (D0/0x300311) is read as 1.

D[1:0] Reserved

0x300316: Area 0 ECC Line Parity Register 0 (pECC0LPL)**0x300317: Area 0 ECC Line Parity Register 1 (pECC0LPH)**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Area 0 ECC line parity register 0 (pECC0LPL)	00300316 (B)	D7	LP07	Area 0 ECC line parity LP00 = LSB	0x0 to 0xFF (low-order 8 bits)	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 (pECC0LPH)	00300317 (B)			D7		
D6	LP014			1				
D5	LP013			1				
D4	LP012			1				
D3	LP011			1				
D2	LP010			1				
D1	LP09			1				
D0	LP08			1				

D[7:0]/0x300316 LP0[7:0]: Area 0 ECC Line Parity (Low-order 8 bits)

D[7:0]/0x300317 LP0[15:8]: Area 0 ECC Line Parity (High-order 8 bits)

The line parity data that is generated by the ECC generator from the 256-byte card data in area 0 is stored in these registers.

The data read from these registers are valid when ECCRDY0 (D0/0x300311) is read as 1.

0x300318: Area 1 ECC Column Parity Data Register (pECC1CP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Area 1 ECC column parity data register (pECC1CP)	00300318 (B)	D7	CP15	Area 1 column parity data	0x0 to 0x3F	1	R	
		D6	CP14					
		D5	CP13					
		D4	CP12					
		D3	CP11					
		D2	CP10					
		D1	–	Unused bit	–	1	R	1 when being read.
		D0	–	Unused bit	–	1	R	

D[7:2] CP1[5:0]: Area 1 Column Parity Data Bits

The column parity data that is generated by the ECC generator from the 256-byte card data in area 1 is stored in this register.

The data read from this register is valid when ECCRDY1 (D1/0x300311) is read as 1.

D[1:0] Reserved

0x30031A: Area 1 ECC Line Parity Register 0 (pECC1LPL)
0x30031B: Area 1 ECC Line Parity Register 1 (pECC1LPH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Area 1 ECC line parity register 0 (pECC1LPL)	0030031A (B)	D7	LP17	Area 1 ECC line parity LP10 = LSB	0x0 to 0xFF (low-order 8 bits)	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 (pECC1LPH)	0030031B (B)			D7		
D6	LP114			1				
D5	LP113			1				
D4	LP112			1				
D3	LP111			1				
D2	LP110			1				
D1	LP19			1				
D0	LP18			1				

D[7:0]/0x30031A LP1[7:0]: Area 1 ECC Line Parity (Low-order 8 bits)

D[7:0]/0x30031B LP1[15:8]: Area 1 ECC Line Parity (High-order 8 bits)

The line parity data that is generated by the ECC generator from the 256-byte card data in area 1 is stored in these registers.

The data read from these registers are valid when ECCRDY1 (D1/0x300311) is read as 1.

V.4.8 Precautions

The PC Card interface (CARD) is subject to the following limitations:

- The interface supports 16-bit PC cards, such as ATA (CF), LAN (Ethernet, wireless), or modem connected as an I/O card. SRAM cards, etc. are not supported.
- Live or hot-line card insertion and removal are not supported. Power must be turned off before inserting or removing a card. The automatic recognition of cards is also not supported.
- DMA, ZV, and CardBus are also not supported.
- To accommodate differences in power supply voltage between the PC card (5 V or 3.3 V) and the S1C33E08, use a buffer IC (e.g., S1C37120).
- The ECC generator supports two 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 card I/O signals must be connected to the D[15:0] pins when using the ECC function.
- ECCRST (D0/0x300311) should be set before using the ECC function.
 - * **ECCRST**: ECC Circuit Reset Bit in the ECC Reset/Ready Register (D0/0x300311)
- The ECCEN (D0/0x300312) should be set only while reading or writing card data. It should be disabled during command input, address input or status reading.
 - * **ECCEN**: ECC Circuit Enable Bit in the ECC Enable Register (D0/0x300312)

V.5 I²S Interface (I²S)

V.5.1 Outline of I²S Module

The S1C33E08 has a built-in I²S module that outputs 16-bit PCM data in the I²S (Inter-IC Sound) format. An audio output circuit can be simply configured by connecting an external DAC to the I²S bus.

The following shows the features of the I²S module:

- 16-bit PCM data output
- An 8-word × 32-bit FIFO is included.
- Stereo, mono (L and R), and mute modes are software selectable.
- Clock polarity is software configurable.
- Data shift direction (MSB first/LSB first) is software selectable.
- FIFO empty can issue a DMA request (dual channel or single channel) or an interrupt request.

Figure V.5.1.1 shows the structure of the I²S module.

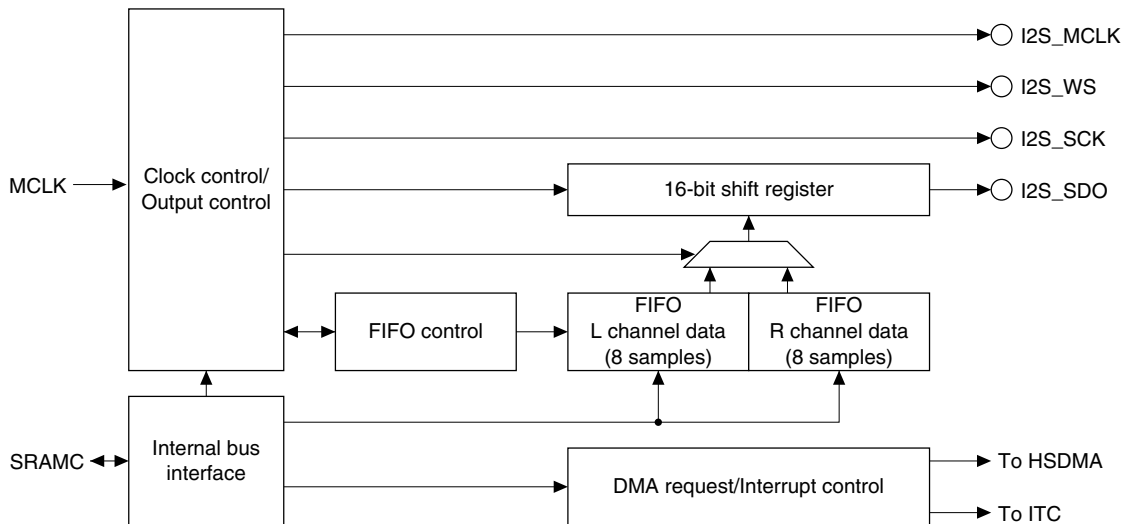


Figure V.5.1.1 Structure of I²S Module

Note: The MP3 decoder BIOS (MP3 decoder module) uses the I²S module, therefore, the I²S module cannot be used while the MP3 decoder is active.

V.5.2 Output Pins of I²S Module

Table V.5.2.1 lists the output pins used by the I²S module.

Table V.5.2.1 I²S Interface Pin Configuration

Pin name	I/O	Function
I2S_SDO	O	Serial data output
I2S_WS	O	Word-select (LRCLK) signal output
I2S_SCK	O	Bit clock output
I2S_MCLK	O	Master clock output

I2S_SDO pin

This pin outputs serial PCM data.

I2S_WS pin

This pin outputs the word-select (LRCLK) signal that indicates the channel (L or R) of the data being output.

I2S_SCK pin

This pin outputs the synchronous clock (bit clock) for serial data.

I2S_MCLK pin

This pin outputs the I²S master clock.

Note: The I²S interface output pins are shared with general-purpose I/O ports or other peripheral circuit inputs/outputs, so that functionality in the initial state is set to other than the I²S interface output. Before the I²S interface output signals assigned to these pins can be used, the function of these pins must be switched for the I²S interface by setting the corresponding Port Function Select Registers.

For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

V.5.3 I²S Module Operating Clock

The I²S module use the I2S_CLK clock (= MCLK) generated by the CMU as the operating clock. The transfer clock is generated in the I²S module by dividing I2S_CLK.

Controlling the supply of the operating clock

I2S_CLK is supplied to the I²S module with default settings. It can be turned off using I2S_CKE (D11/0x301B04) to reduce the amount of power consumed on the chip if the I²S module is not used.

* **I2S_CKE**: I²S Clock Control Bit in the Gated Clock Control Register 1 (D11/0x301B04)

Setting I2S_CKE (D11/0x301B04) to 0 (1 by default) turns off the clock supply to the I²S module. When the clock supply is turned off, the I²S module control registers cannot be accessed.

For details on how to set and control the clock, refer to Section III.1, “Clock Management Unit (CMU).”

Note: The Gated Clock Control Register 1 (0x301B04) is write-protected. Write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless re-writing said registers.

Clock state in standby mode

The clock supply to the I²S module stops depending on type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

SLEEP mode: The operating clock supply stops.

Therefore, the I²S module also stops operating in SLEEP mode.

V

I²S

V.5.4 Setting I²S Module

When performing data transfers via the I²S bus, the following settings must be made before data transfer is actually begun:

1. Setting output pins
2. Setting I²S interface clocks
3. Setting the output data format and timing
4. Setting HSDMA
5. Setting interrupts

The following explains the content of each setting. For details on interrupt settings, refer to Section V.5.6, “I²S Interrupt.”

Note: Always make sure the I²S module is inactive (I2SEN (D7/0x301C00) = 0) before these settings are made. A change of settings during operation may cause a malfunction.

* **I2SEN:** I²S Module Enable Bit in the I²S Control Register (D7/0x301C00)

Setting output pins

Configure the Port Function Select Registers to enable the I²S output functions. For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Setting the I²S interface clocks

The I²S module outputs the following three clocks to the slave device (DAC) (some DACs use two clocks):

1. I2S_MCLK (master clock)
2. I2S_WS (LRCLK clock)
3. I2S_SCK (bit clock)

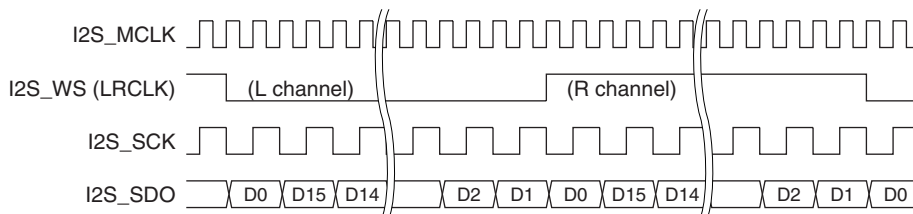


Figure V.5.4.1 I²S Interface Clocks

The following shows how to configure the clocks:

Divide ratio for I2S_MCLK (master clock)

The I²S module generates the I2S_MCLK by dividing the operating clock (MCLK generated by the CMU). Specify the divide ratio using I2SMCLK[5:0] (D[5:0]/0x301C04).

* **I2SMCLK[5:0]:** I2S_MCLK Divide Ratio Select Bits in the I²S MCLK Divide Ratio Register (D[5:0]/0x301C04)

Table V.5.4.1 Setting I2S_MCLK (Master Clock)

I2SMCLK[5:0]	I2S_MCLK
111111	MCLK _(CMU) /64
111110	MCLK _(CMU) /63
111101	MCLK _(CMU) /62
:	:
000010	MCLK _(CMU) /3
000001	MCLK _(CMU) /2
000000	MCLK _(CMU) /1

(Default: 0b000000 = MCLK/1)

Divide ratio for I2S_WS (LRCLK clock)

LRCLK is the clock output as the word-select signal and is generated by dividing I2S_MCLK. Specify the divide ratio using LRCLK[2:0] (D[2:0]/0x301C08).

* **LRCLK[2:0]**: LRCLK Divide Ratio Select Bits in the I²S Audio Clock Divide Ratio Register (D[2:0]/0x301C08)

Table V.5.4.2 Setting LRCLK

LRCLK2	LRCLK1	LRCLK0	LRCLK
1	1	1	I2S_MCLK/512
1	1	0	I2S_MCLK/448
1	0	1	I2S_MCLK/384
1	0	0	I2S_MCLK/320
0	1	1	I2S_MCLK/256
0	1	0	I2S_MCLK/192
0	0	1	I2S_MCLK/128
0	0	0	I2S_MCLK/64

(Default: 0b000 = I2S_MCLK/64)

Configure LRCLK to the PCM sampling frequency.

The bit clock is configured to LRCLK × 32[Hz]. For example, the bit clock frequency is 1.536 MHz when LRCLK is set to 48 kHz.

Selecting the word clock

The I2S_WS (LRCLK) signal represents the current output channel (L or R) with its level (low or high). Use WCLKMD (D3/0x301C00) to select the relationship between the signal level and the channel.

* **WCLKMD**: I²S Output Word Clock Mode Select Bit in the I²S Control register (D3/0x301C00)

WCLKMD = 0 (default)



WCLKMD = 1



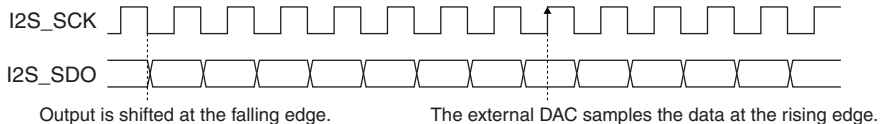
Figure V.5.4.2 Selecting Word Clock Mode

I2S_SCK (bit clock) polarity

Use BCLKPOL (D2/0x301C00) to select the bit clock polarity.

* **BCLKPOL**: I²S Output Bit Clock Polarity Select Bit in the I²S Control register (D2/0x301C00)

BCLKPOL = 0 (default)



BCLKPOL = 1

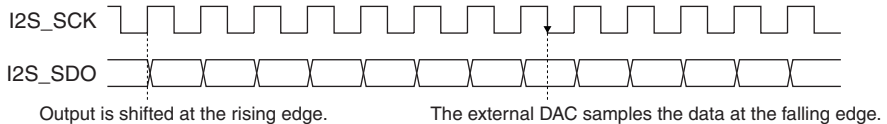


Figure V.5.4.3 Selecting the Bit Clock Polarity

Setting the output data format and timing

Use DTFORM (D0/0x301C00) to select either MSB first or LSB first as the data output direction. Setting DTFORM (D0/0x301C00) to 0 (default) selects MSB first and setting 1 selects LSB first. The output channel mode and HSDMA mode do not affect the output data format and it is always treated in 16 bit units.

* **DTFORM:** I²S Output Data Format Select Bit in the I²S Control Register (D0/0x301C00)

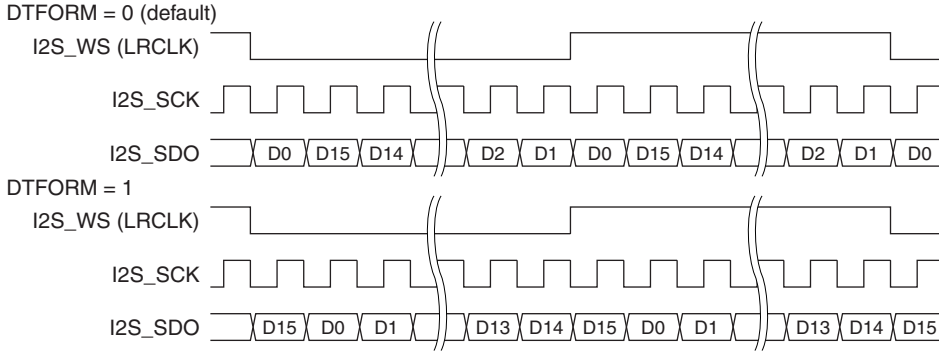


Figure V.5.4.4 Selecting Output Data Format

In the default setting, the first bit of each data is output after one I2S_SCK clock delay from the I2S_WS signal edge. This timing can be changed by writing 1 to DTTMG (D1/0x301C00) so that each data output will start at the I2S_WS signal edge.

* **DTTMG:** I²S Output Data Timing Select Bit in the I²S Control Register (D1/0x301C00)

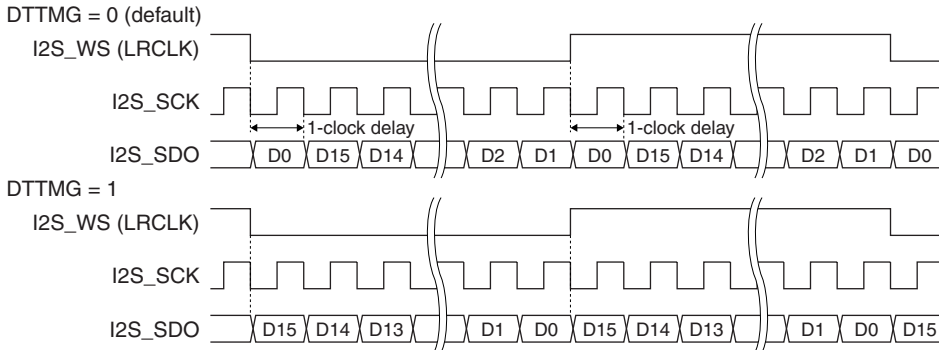


Figure V.5.4.5 Selecting Data Output Timing

Setting HSDMA

The I²S module has an embedded 8-word FIFO for storing PCM data and is able to output DMA requests to write data when the FIFO is not full. The I²S module has two DMA request signal paths corresponding to L and R channels and these signals are directly input to HSDMA. When transferring L and R data together, one DMA request signal only is used, or when transferring L and R data individually, two signals are used. I2SHSDMA (D0/0x301C10) is used for this selection.

* **I2SHSDMA**: I²S HSDMA Mode Select Bit in the I²S HSDMA Mode Select Register (D0/0x301C10)

When one DMA request signal is used (I2SHSDMA = 0, default)

When one data in the FIFO is loaded to the shift register, the L channel DMA request signal to the HSDMA Ch.0 is asserted. Program HSDMA so that up to eight 32-bit data will be transferred to the FIFO (0x301C20) using the requests as triggers. The 32-bit data consists of an L channel data of upper 16 bits and an R channel data of lower 16 bits.

The following shows standard settings for HSDMA Ch.0. For details of HSDMA, see Section II.1, “High-Speed DMA (HSDMA).”

1. HSDMA STD/ADV Mode Select Register (0x30119C) = 0x1
Set HSDMA in advanced mode to perform 32-bit data transfer.
2. HSDMA Ch.0 Control Register for ADV Mode (0x301162) = 0x1
Set the transfer data size to word (32 bits).
3. HSDMA Ch.0 Source Address Setup Registers for ADV Mode (0x301164, 0x301166) = PCM data address
Specify the PCM data start address in the memory as the source address.
4. HSDMA Ch.0 Destination Address Setup Registers for ADV Mode (0x301168, 0x30116A) = 0x301C20
Specify the data write address of the I²S FIFO as the destination address.
5. HSDMA Ch.0 Transfer Counter Register (0x301120) = 0x8 to PCM data size
Configure the transfer counter with the word length to be transferred. Be sure not to set a value of less than 8. The I²S module fills the FIFO with output data until it becomes full before outputting. After that, every time word data is read out from the FIFO, the I²S module outputs a DMA request until the FIFO becomes full again.
6. HSDMA Ch.0 Control Register (0x301122) = 0x80
Select dual-address mode.
7. HSDMA Ch.0 High-Order Source Address Setup Register (0x301126) = 0x3000
Select “address incremented without initialization” for the source address control.
8. HSDMA Ch.0 High-Order Destination Address Setup Register (0x30112A) = 0x0000
Set HSDMA in single transfer mode to transfer one word per transfer by one trigger.
Select “address fixed” for the destination address control.
9. HSDMA Ch.0–1 Trigger Set-up Register (0x300298) = 0x*9
Select I²S as the trigger source for HSDMA Ch.0. A trigger source for HSDMA Ch.1 should be set to * (upper four bits).

When two DMA request signals are used (I²SHSDMA = 1)

This mode should be selected when PCM data in the memory are separated into L and R channels. In this mode, an L channel DMA request and an R channel DMA request occur when data in the FIFO is loaded to the shift register and the FIFO becomes less than full.

In this mode, 16-bit data for each channel can be transferred and written to the FIFO individually.

The L channel DMA request signal is output to HSDMA Ch.0 and the R channel DMA request signal is output to HSDMA Ch.1. Therefore, HSDMA Ch.0 and Ch.1 should be programmed as shown below. For details of HSDMA, see Section II.1, “High-Speed DMA (HSDMA).”

1. HSDMA STD/ADV Mode Select Register (0x30119C) = 0x1
Set HSDMA in advanced mode. Standard mode can be set for 16-bit data transfer.
2. HSDMA Ch.0 Source Address Setup Registers for ADV Mode (0x301164, 0x301166) = L channel data address
HSDMA Ch.1 Source Address Setup Registers for ADV Mode (0x301174, 0x301176) = R channel data address
Specify the L and R channel PCM data start addresses in the memory as the source addresses.
3. HSDMA Ch.0 Destination Address Setup Registers for ADV Mode (0x301168, 0x30116A) = 0x301C20
HSDMA Ch.1 Destination Address Setup Registers for ADV Mode (0x301178, 0x30117A) = 0x301C22
Specify the data write address of the I²S FIFO for each channel.
4. HSDMA Ch.0 Transfer Counter Register (0x301120) = 0x8 to left channel PCM data size
HSDMA Ch.1 Transfer Counter Register (0x301130) = 0x8 to right channel PCM data size
Configure the transfer counters with the data length to be transferred. Be sure not to set a value of less than 8 as the I²S FIFO size is 8 words.
5. HSDMA Ch.0 Control Register (0x301122) = 0x80
HSDMA Ch.1 Control Register (0x301132) = 0x80
Select dual-address mode.
6. HSDMA Ch.0 High-Order Source Address Setup Register (0x301126) = 0x7000
HSDMA Ch.1 High-Order Source Address Setup Register (0x301136) = 0x7000
Set the transfer data size to half-word (16 bits).
Select “address incremented without initialization” for the source address control.
7. HSDMA Ch.0 High-Order Destination Address Setup Register (0x30112A) = 0x0000
HSDMA Ch.1 High-Order Destination Address Setup Register (0x30113A) = 0x0000
Set HSDMA in single transfer mode to transfer 16 bits per transfer by one trigger.
Select “address fixed” for the destination address control.
8. HSDMA Ch.0–1 Trigger Set-up Register (0x300298) = 0x99
Select I²S as the trigger source for HSDMA Ch.0 and Ch.1.

V.5.5 Data Output Control

The following shows PCM data output procedure:

1. Set up the I²S and HSDMA conditions as described in the previous section.
2. When HSDMA Ch.0 and Ch.1 cannot be used for I²S, set up the interrupt conditions using the ITC registers (explained later). When using the I²S interrupt, the cause-of-interrupt flag in the ITC must be cleared before enabling the interrupt.
3. Write 1 to the I2SEN (D7/0x301C00) to turn the I²S circuit on.
The I²S circuit starts frequency division of the source clock.

* **I2SEN**: I²S Module Enable Bit in the I²S Control Register (D7/0x301C00)

4. Set the output channel mode using CHMD[1:0] (D[5:4]/0x301C00).

* **CHMD[1:0]**: I²S Output Channel Mode Select Bits in the I²S Control Register (D[5:4]/0x301C00)

Table V.5.5.1 Selecting Output Channel Mode

CHMD1	CHMD0	Output channel mode	L channel	R channel
1	1	Mute	0	0
1	0	Mono (L)	Data output	0
0	1	Mono (R)	0	Data output
0	0	Stereo	Data output	Data output

(Default: 0b00 = Stereo)

The output channel mode can be switched even if data is being output. In this case, the mode changes after the current word output has finished.

5. Write the first PCM word to the FIFO.

When writing a 32-bit data for L and R channels together, perform word write to 0x301C20.

When writing data for each channel individually, write a 16-bit L channel data to 0x301C20 and a 16-bit R channel data to 0x301C22 using a half-word access instruction.

Note that the newest data of the FIFO are overwritten if two or more words are written to the FIFO before 1 is written to I2SSTART (D0/0x301C0C).

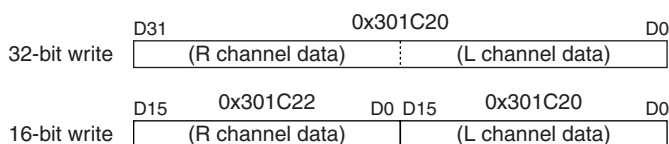


Figure V.5.5.1 Writing to FIFO

6. Write 1 to I2SOUTEN (D6/0x301C00) to enable I²S output.

* **I2SOUTEN**: I²S Output Enable Bit in the I²S Control Register (D6/0x301C00)

When I2SOUTEN (D6/0x301C00) = 0, the I2S_MCLK, I2S_WS, and I2S_SDO pins are fixed at 0. The I2S_SCK pin is fixed at 0 (when BCLKPOL (D2/0x301C00) = 0) or 1 (when BCLKPOL (D2/0x301C00) = 1).

When I2SOUTEN (D6/0x301C00) is set to 1, all output pins enter standby status.

* **BCLKPOL**: I²S Output Bit Clock Polarity Select Bit in the I²S Control register (D2/0x301C00)

7. Write 1 to I2SSTART (D0/0x301C0C) to start output.

* **I2SSTART**: I²S Start/Stop Control Bit in the I²S Start Register (D0/0x301C0C)

The I²S circuit loads one word of data in the FIFO to the shift register and it starts serial output in sync with the I2S_WS signal. The data in the shift register is shifted at the I2S_SCK clock edge and is output from the L channel first. When an output of one word has finished, the next data is read out from the FIFO and the same operation repeats.

When the I²S module has finished outputting the last data in the FIFO, I2SFIFOEF (D0/0x301C14) is set to 1.

* **I2SFIFOEF**: I²S FIFO Empty Flag in the I²S FIFO Status Register (D0/0x301C14)

V PERIPHERAL MODULES 3 (INTERFACE): I²S INTERFACE (I²S)

Every time word data is read out from the FIFO and the FIFO turns lack from full, a DMA request is output to HSDMA. If the DMA transfer conditions described in the previous section has been set and DMA is enabled, the following data is written to the FIFO in a DMA transfer. When data is written to the FIFO, I2SFIFOEF (D0/0x301C14) is reset to 0 and the data output continues.

The FIFO size is 8 words so up to 8 samples of PCM data can be written to it. If writing data exceeds 8 words, the newest data of the FIFO are overwritten.

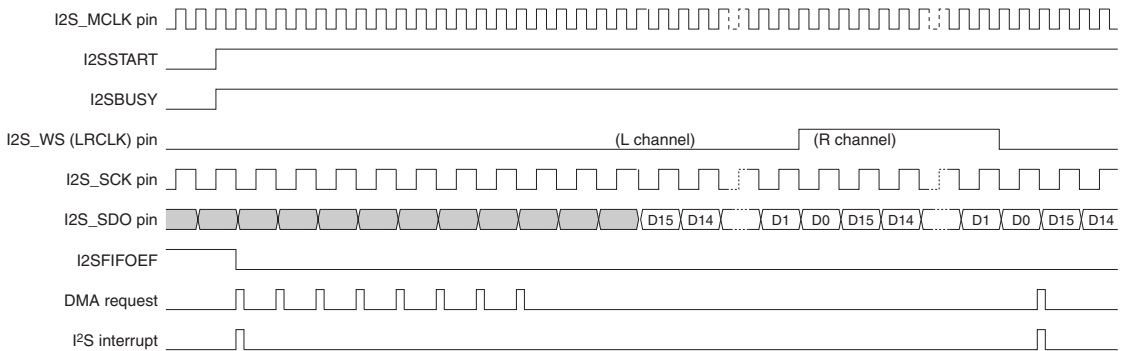
Every time the FIFO changes from full into lack, an interrupt can be generated. By using this interrupt, data can be written to the FIFO in an interrupt handler routine. For more information on the interrupt, see Section V.5.6, “I²S Interrupt.”

I2SBUSY (D7/0x301C0C) is set to 1 while data is being output. This flag can be used to check the output status.

* **I2SBUSY**: I²S Module Busy Flag in the I²S Start Register (D7/0x301C0C)

8. To stop output, write 0 to I2SSTART (D0/0x301C0C).
When I2SSTART (D0/0x301C0C) is set to 0, the I²S module will stop data output after the remaining data stored in the FIFO are all output. When the I²S stops, I2SBUSY (D7/0x301C0C) is reset to 0.
9. To disable output, write 0 to I2SOUTEN (D6/0x301C00) after writing 0 to I2SSTART (D0/0x301C0C) to stop the current output. Finally, write 0 to I2SEN (D7/0x301C00) to turn the I²S circuit off.

When I2SSTART is set to 1



When I2SSTART is set to 0

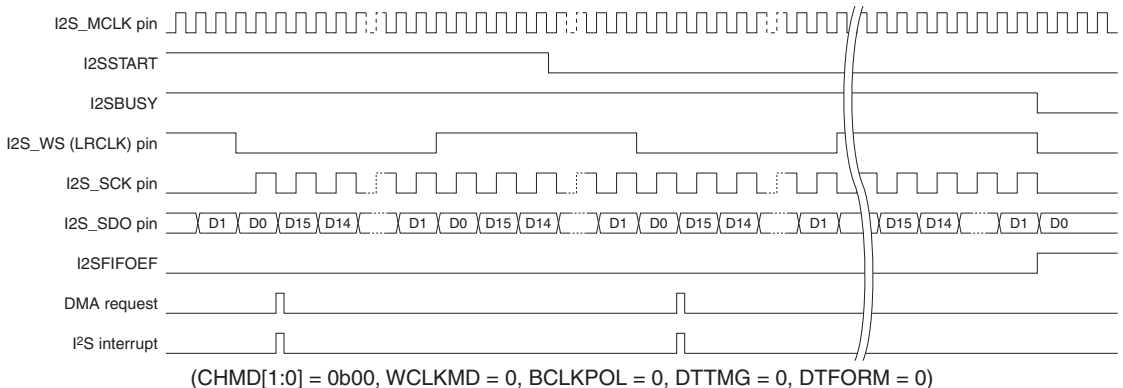


Figure V.5.5.2 Data Output Timing Chart

* **Output when mute or mono mode is selected**

When mute mode is selected using CHMD[1:0] (D[5:4]/0x301C00), the I2S_SDO pin is fixed at 0. However, the FIFO and shift register run the same as stereo mode and three clock signals are output normally. Also in mono mode, the I2S_SDO pin is fixed at 0 during the output period for the unselected channel.

The FIFO data is read out normally, therefore a DMA request or interrupt caused by a FIFO empty occurs.

If CHMD[1:0] (D[5:4]/0x301C00) is changed when data is being output, the mode changes after the current word output has finished.

V.5.6 I²S Interrupt

The I²S module can generate an interrupt every time the FIFO is not full by loading one data in the FIFO to the shift register.

Control registers of the interrupt controller

Table V.5.6.1 shows the interrupt controller's control registers provided for the I²S interrupt.

Table V.5.6.1 Control Register of Interrupt Controller

Interrupt	Cause-of-interrupt flag	Interrupt enable register	Interrupt priority register
I ² S interrupt	FI2S(D2/0x3002AA)	EI2S(D2/0x3002A7)	PI2S[2:0](D[2:0]/0x3002A4)

When a cause of FIFO empty interrupt occurs, the cause-of-interrupt flag listed above is set to 1. If the interrupt enable register bit for that cause of interrupt has been set to 1, an interrupt request is generated.

Interrupts can be disabled by leaving the interrupt enable register bit for that cause of interrupt set to 0. The cause-of-interrupt 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 Section III.2, "Interrupt Controller (ITC)."

Intelligent DMA

The I²S interrupt request can be used to invoke intelligent DMA (IDMA). This enables data transfer between memory and the FIFO to be performed using DMA.

The IDMA channel numbers set for the cause of I²S interrupt is 0x2E.

The IDMA request and enable bits shown in Table V.5.6.2 must be set to 1 for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

Table V.5.6.2 Control Bits for IDMA Transfer

Interrupt	IDMA request bit	IDMA enable bit
I ² S interrupt	RI2S(D0/0x3002AD)	DEI2S(D0/0x3002AF)

If a cause of interrupt 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 DMA transfer performed.

For details on DMA transfer and how to control interrupts upon completion of DMA transfer, refer to Section II.2, "Intelligent DMA (IDMA)."

High-speed DMA

The cause of I²S interrupt cannot invoke high-speed DMA (HSDMA).

DMA transfer can be performed using the DMA request signals that are directly output to HSDMA from the I²S module. For the HSDMA settings, see Section V.5.4, "Setting I²S Module."

Trap vectors

The default trap-vector address of the cause of I²S interrupt is 0xC00178.

The base address of the trap table can be changed using the TTBR register.

V.5.7 Details of Control Registers

Table V.5.7.1 List of I²S Registers

Address	Register name	Size	Function
0x00301C00	I ² S Control Register (pl2S_CONTROL)	32	Sets I ² S signals and controls module.
0x00301C04	I ² S MCLK Divide Ratio Register (pl2S_DV_MCLK)	32	Sets I2S_MCLK clock.
0x00301C08	I ² S Audio Clock Divide Ratio Register (pl2S_DV_LRCLK)	32	Sets LRCLK clock.
0x00301C0C	I ² S Start Register (pl2S_START)	32	Output control and status
0x00301C10	I ² S HSDMA Mode Select Register (pl2S_HSDMAMD)	32	Sets HSDMA mode.
0x00301C14	I ² S FIFO Status Register (pl2S_FIFO_EMPTY)	32	FIFO empty status
0x00301C20	I ² S FIFO Register (pl2S_FIFO)	32	FIFO data write address

The following describes each I²S control register.

The I²S control registers are mapped in the 32-bit device area from 0x301C00 to 0x301C20. Addresses 0x301C00 to 0x301C14 can be accessed in units of words, half-words, or bytes and address 0x301C20 can be accessed in units of words or half-words.

Note: When setting the I²S control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x301C00: I²S Control Register (pI²S_CONTROL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
I ² S control register (pI ² S_CTRL)	00301C00 (W)	D31–8	–	reserved	–		–	–	0 when being read.	
		D7	I2SEN	I ² S module enable	1	Enabled	0	Disabled	0	R/W
		D6	I2SOUTEN	I ² S output enable	1	Enabled	0	Disabled	0	R/W
		D5	CHMD1 CHMD0	I ² S output channel mode	CHMD[1:0]		Mode		0	R/W
		11			Mute		0			
		10			Mono left					
		01			Mono right					
		D4	00		Stereo					
		D3	WCLKMD	I ² S output word clock mode	1	L: high R: low	0	L: low R: high	0	R/W
D2	BCLKPOL	I ² S output bit clock polarity	1	Negative	0	Positive	0	R/W		
D1	DTTNG	I ² S output data timing	1	Not delayed	0	Delayed	0	R/W		
D0	DTFORM	I ² S output data format	1	LSB first	0	MSB first	0	R/W		

D[31:8] Reserved**D7 I2SEN: I²S Enable Bit**

Enables/disables operation of the I²S module.

1 (R/W): Enable (on)

0 (R/W): Disable (off) (default)

When I2SEN is set to 1, the I²S module starts operating and data transfer is enabled.

When I2SEN is set to 0, the I²S module goes off.

Make sure that this bit is 0 before setting up data transfer conditions using the I²S registers.

D6 I2SOUTEN: I²S Output Enable Bit

Enables/disables output of the I²S signals.

1 (R/W): Enable (on)

0 (R/W): Disable (off) (default)

When I2SOUTEN = 0, the I2S_MCLK, I2S_WS, and I2S_SDO pins are fixed at 0. The I2S_SCK pin is fixed at 0 (when BCLKPOL (D2) = 0) or 1 (when BCLKPOL (D2) = 1).

When I2SOUTEN is set to 1, all output pins enter standby status.

D[5:4] CHMD[1:0]: I²S Output Channel Mode Select Bits

Selects the I²S output channel mode.

Table V.5.7.2 Selecting Output Channel Mode

CHMD1	CHMD0	Output channel mode	L channel	R channel
1	1	Mute	0	0
1	0	Mono (L)	Data output	0
0	1	Mono (R)	0	Data output
0	0	Stereo	Data output	Data output

(Default: 0b00 = Stereo)

The output channel mode can be switched even if data is being output. In this case, the mode changes after the current word output has finished.

When mute mode is selected, the I2S_SDO pin is fixed at 0. However, the FIFO and shift register run the same as stereo mode and three clock signals are output normally. Also in mono mode, the I2S_SDO pin is fixed at 0 during the output period for the unselected channel.

The FIFO data is read out normally, therefore a DMA request or interrupt occurs.

D3 WCLKMD: I²S Output Word Clock Mode Select Bit

Selects the I2S_WS (LRCLK) signal level for indicating a channel.

1 (R/W): High = L channel, Low = R channel

0 (R/W): High = R channel, Low = L channel (default)

WCLKMD = 0 (default)



WCLKMD = 1



Figure V.5.7.1 Selecting Word Clock Mode

D2 BCLKPOL: I²S Output Bit Clock Polarity Select Bit

Selects the bit clock polarity.

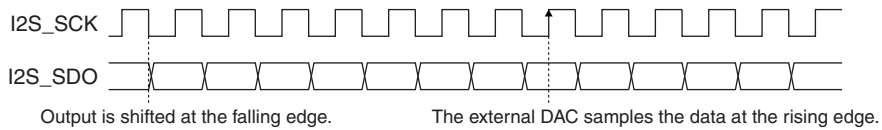
1 (R/W): Active low

0 (R/W): Active high (default)

When BCLKPOL is 0, the I2S_SDO output changes at the falling edge of I2S_SCK and the external DAC samples the data bit at the rising edge of I2S_SCK.

When BCLKPOL is set to 1, the I2S_SDO output changes at the rising edge of I2S_SCK and the external DAC samples the data bit at the falling edge of I2S_SCK.

BCLKPOL = 0 (default)



BCLKPOL = 1

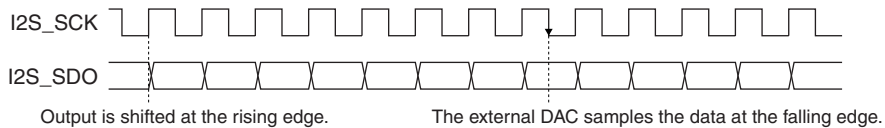


Figure V.5.7.2 Selecting Bit Clock Polarity

D1 DTTMG: I²S Output Data Timing Select Bit

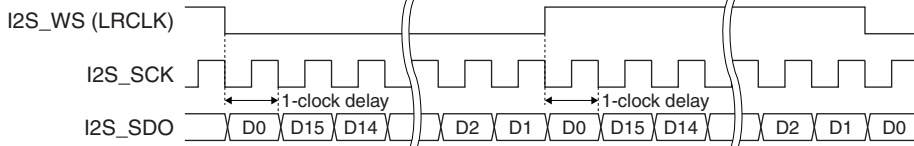
Selects the data bit output timing.

1 (R/W): I2S_WS edge

0 (R/W): 1 clock delay from I2S_WS edge (default)

In the default setting, the first bit of each data is output after one I2S_SCK clock delay from the I2S_WS signal edge. This timing can be changed by writing 1 to DTTMG so that each data output will start at the I2S_WS signal edge.

DTTMG = 0 (default)



DTTMG = 1

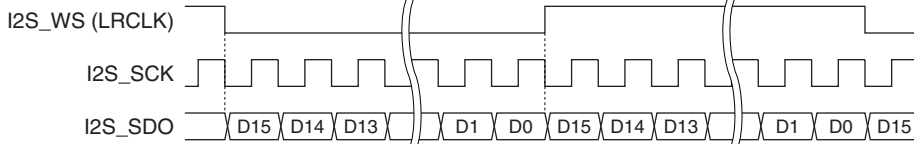


Figure V.5.7.3 Selecting Data Output Timing

D0 DTFORM: I²S Output Data Format Select Bit

Selects either MSB first or LSB first as the data output direction.

1 (R/W): LSB first

0 (R/W): MSB first (default)

The output channel mode and HSDMA mode do not affect the output data format and it is always treated in 16 bit units.

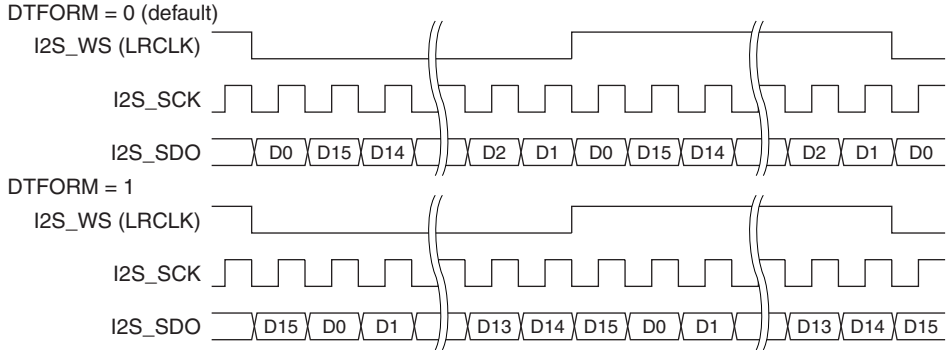


Figure V.5.7.4 Selecting Data Format

0x301C04: I²S MCLK Divide Ratio Register (pI2S_DV_MCLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S MCLK divide ratio register (pI2S_DV_MCLK)	00301C04 (W)	D31-6	–	reserved	–	–	–	0 when being read.	
		D5	I2SMCLK5	I2S_MCLK divide ratio selection	I2SMCLK[5:0]	I2S_MCLK	0	R/W	
		D4	I2SMCLK4		111111	MCLK _(CMU) /64	0		
		D3	I2SMCLK3		111110	MCLK _(CMU) /63	0		
		D2	I2SMCLK2				0		
		D1	I2SMCLK1		000001	MCLK _(CMU) /2	0		
		D0	I2SMCLK0		000000	MCLK _(CMU) /1	0		

D[31:6] Reserved

D[5:0] I2SMCLK[5:0]: I2S_MCLK Divide Ratio Select Bits

Configures the I²S master clock (I2S_MCLK).

The I²S module generates the I2S_MCLK by dividing the operating clock (MCLK generated by the CMU). Specify the divide ratio using I2SMCLK[5:0].

Table V.5.7.3 Setting I2S_MCLK (Master Clock)

I2SMCLK[5:0]	I2S_MCLK
111111	MCLK _(CMU) /64
111110	MCLK _(CMU) /63
111101	MCLK _(CMU) /62
:	:
000010	MCLK _(CMU) /3
000001	MCLK _(CMU) /2
000000	MCLK _(CMU) /1

(Default: 0b000000 = MCLK/1)

0x301C08: I²S Audio Clock Divide Ratio Register (pi2S_DV_LRCLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S audio clock divide ratio register (pi2S_DV_LRCLK)	00301C08 (W)	D31–3	–	reserved	–	–	–	0 when being read.	
		D2	LRCLK2	LRCLK divide ratio selection	LRCLK[2:0]	LRCLK	0	R/W	
		D1	LRCLK1		111	I2S_MCLK/512	0		
		D0	LRCLK0		110	I2S_MCLK/448	0		
					101	I2S_MCLK/384			
					100	I2S_MCLK/320			
					011	I2S_MCLK/256			
					010	I2S_MCLK/192			
					001	I2S_MCLK/128			
					000	I2S_MCLK/64			

D[31:3] Reserved

D[2:0] LRCLK[2:0]: LRCLK Divide Ratio Select Bits

Configures the LRCLK clock (I2S_WS signal).

LRCLK is generated by dividing I2S_MCLK. Specify the divide ratio using LRCLK[2:0].

Table V.5.7.4 Setting LRCLK

LRCLK2	LRCLK1	LRCLK0	LRCLK
1	1	1	I2S_MCLK/512
1	1	0	I2S_MCLK/448
1	0	1	I2S_MCLK/384
1	0	0	I2S_MCLK/320
0	1	1	I2S_MCLK/256
0	1	0	I2S_MCLK/192
0	0	1	I2S_MCLK/128
0	0	0	I2S_MCLK/64

(Default: 0b000 = I2S_MCLK/64)

Configure LRCLK to the PCM sampling frequency.

The bit clock is configured to $LRCLK \times 32[\text{Hz}]$. For example, the bit clock frequency is 1.536 MHz when LRCLK is set to 48 kHz.

0x301C0C: I²S Start Register (pl2S_START)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S start register (pl2S_START)	00301C0C (W)	D31-8	–	reserved	–	–	–	0 when being read.
		D7	I2SBUSY	I ² S module busy flag	1 Busy 0 Idle	0	R	
		D6-1	–	reserved	–	–	–	0 when being read.
		D0	I2SSTART	I ² S start/stop	1 Start 0 Stop	0	R/W	

D[31:8] Reserved**D7 I2SBUSY: I²S Busy Flag**

Indicates the data output status of the I²S module.

1 (R): Busy

0 (R): Idle (default)

I2SBUSY is set to 1 when the I²S module starts data output and stays 1 while data is being output. This flag is cleared to 0 upon completion of the output operation.

D[6:1] Reserved**D0 I2SSTART: I²S Start/Stop Control Bit**

Starts/stops the I²S output.

1 (R/W): Start

0 (R/W): Stop (default)

Writing 1 to I2SSTART starts serial output.

Writing 0 to I2SSTART stops output operation after the current word output has finished.

0x301C10: I²S HSDMA Mode Select Register (pI2S_HSDMAMD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S HSDMA mode select register (pI2S_HSDMAMD)	00301C10 (W)	D31-1	–	reserved	–	–	–	0 when being read.
		D0	I2SHSDMA	I ² S HSDMA mode selection	1 16 bits × 2 ch. 0 32 bits × 1 ch.	0	R/W	

D[31:1] Reserved**D0 I2SHSDMA: I²S HSDMA Mode Select Bit**

Sets the HSDMA mode.

1 (R/W): 16 bits × 2 channels

0 (R/W): 32 bits × 1 channel (default)

Select 32 bits × 1 channel mode when 32-bit PCM data that contains L and R channel samples is prepared in the memory. Select 16 bits × 2 channels mode when 16-bit PCM data for each channel is prepared individually.

The I²S module has two DMA request signal outputs to HSDMA (used to request L and R channel data).

In the default setting (I2SHSDMA = 0), the DMA request signal for the L channel is enabled and it is asserted when the FIFO is not full. This signal invokes HSDMA Ch.0 to transfer 32-bit L&R data to the FIFO in the set conditions.

When I2SHSDMA is set to 1, two DMA request signals separated into L and R channels are enabled.

In this case, both L and R channel DMA request signals are asserted when the FIFO is not full. These signals invoke HSDMA Ch.0 and Ch.1. HSDMA Ch.0 writes 16-bit L data to the L data field of the FIFO in the set conditions and HSDMA Ch.1 writes 16-bit R data to the R data field of the FIFO.

HSDMA must be configured so that DMA transfer above can be performed.

V

I²S

0x301C14: I²S FIFO Status Register (pl2S_FIFO_EMPTY)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S FIFO status register (pl2S_FIFO_EMPTY)	00301C14 (W)	D31-1	–	reserved	–	–	–	0 when being read.
		D0	I2SFIFOEF	I ² S FIFO empty flag	1 Empty 0 Not empty	0	R	

D[31:1] Reserved**D0 I2SFIFOEF: I²S FIFO Empty Flag**

Indicates the FIFO status.

1 (R): Empty

0 (R): Not empty (default)

I2SFIFOEF is reset to 0 when PCM data is written to the FIFO and is set to 1 when the I²S module has finished outputting the last data in the FIFO.

0x301C20: I²S FIFO Register (pl2S_FIFO)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S FIFO register (pl2S_FIFO)	00301C20 (W)	D31-0	I2SFIFO[31:0]	PCM data for writing to FIFO		0	W	0 when being read.

Note: This register must be accessed using a word (32-bit) access instruction or a half-word (16-bit) access instruction.

D[31:0] I2SFIFO[31:0]: PCM Data for Writing to FIFO

Write data to the FIFO through this address.

When writing a 32-bit data for L and R channels together, perform word write to 0x301C20.

When writing data for each channel individually, write a 16-bit L channel data to 0x301C20 and a 16-bit R channel data to 0x301C22 using a half-word access instruction.

The FIFO size is 8 words so up to 8 samples of PCM data can be written to it. If writing data exceeds 8 words, the newest data of the FIFO are overwritten.

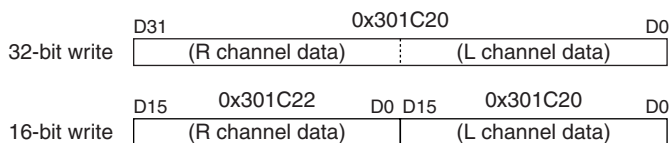


Figure V.5.7.5 Writing to FIFO

THIS PAGE IS BLANK.

S1C33E08 Technical Manual

VI PERIPHERAL MODULES 4 (PORTS)

VI.1 General-Purpose I/O Ports (GPIO)

VI.1.1 Structure of I/O Port

The S1C33E08 contains 74 I/O ports (P0[7:0], P1[7:0], P2[7:0], P3[6:0], P4[7:0], P5[7:0], P6[7:0], P7[4:0], P8[5:0], and P9[7:0]) that can be directed for input or output through the use of a program. 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.

Figure VI.1.1.1 shows the structure of a typical I/O port.

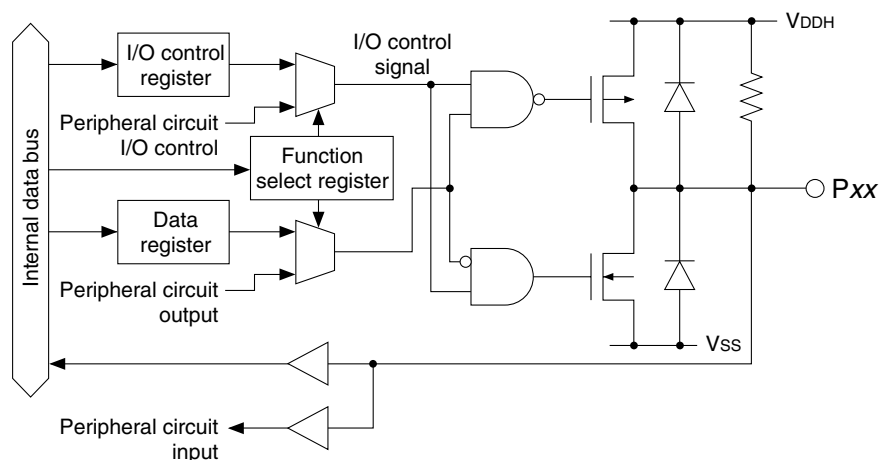


Figure VI.1.1.1 Structure of I/O Port

Note: A pull-up resistor is provided for each pin and it can be enabled/disabled by software control. Refer to Section III.4.4, “Pin Control Registers,” for how to control the pull-up resistor.

When the port is in output mode, the port pin is not pulled up regardless of how the pull-up control bit is set.

VI.1.2 Selecting the I/O Pin Functions

The I/O ports concurrently serve as the input/output pins for peripheral circuits or bus signals. Whether they are used as I/O ports or for peripheral circuits/bus signals can be selected bit-for-bit using the port function select registers. All pins not used for peripheral circuits/bus signals can be used as general-purpose I/O ports.

Each I/O port pin (Pxx) is initialized for a default function at initial reset.

For the pin that has two or more functions assigned, the port extended function select bits (CFPxx[1:0]) provided for each I/O port pin can be used to select the desired function.

For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

The subsequent sections explain the port functions assuming that the pin has been set as a general-purpose I/O port.

Notes:

- To use the P15–P17 and P34–P36 pins that are configured as the debug interface pins by default for general-purpose inputs/outputs, clear TRCMUX (D0/0x300014) to 0.

* **TRCMUX:** P15–17, P34–36 Debug Function Select Bit in the Debug Port MUX Register (D0/0x300014)

Note, however, that the PC trace function of the debugger cannot be used when TRCMUX (D0/0x300014) is set to 0.

- The MP3 decoder BIOS (MP3 decoder module) uses the P04, P05, P06, and P07 ports for the I²S signal input/outputs. Therefore, these ports cannot be used in the user program.

VI.1.3 I/O Control Register and I/O Modes

The I/O ports are directed for input or output modes by writing data to IOC_x corresponding to each port bit.

- * **IOC0[7:0]**: P07–P00 I/O Control Bits in the P0 I/O Control Register (D[7:0]/0x300381)
- * **IOC1[7:0]**: P17–P10 I/O Control Bits in the P1 I/O Control Register (D[7:0]/0x300383)
- * **IOC2[7:0]**: P27–P20 I/O Control Bits in the P2 I/O Control Register (D[7:0]/0x300385)
- * **IOC3[6:0]**: P36–P30 I/O Control Bits in the P3 I/O Control Register (D[6:0]/0x300387)
- * **IOC4[7:0]**: P47–P40 I/O Control Bits in the P4 I/O Control Register (D[7:0]/0x300389)
- * **IOC5[7:0]**: P57–P50 I/O Control Bits in the P5 I/O Control Register (D[7:0]/0x30038B)
- * **IOC6[7:0]**: P67–P60 I/O Control Bits in the P6 I/O Control Register (D[7:0]/0x30038D)
- * **IOC8[5:0]**: P85–P80 I/O Control Bits in the P8 I/O Control Register (D[5:0]/0x300391)
- * **IOC9[7:0]**: P97–P90 I/O Control Bits in the P9 I/O Control Register (D[7:0]/0x300393)

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. The port pin is pulled up when the pull-up resistor is enabled using the pin control register.

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_{DDH} 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_{DDH} level); when the data is 0, the port outputs a low level (V_{SS} level). When the port is in output mode, the port pin is not pulled up even if the pull-up resistor is enabled.

Note: The P7_x port has no I/O control register, as it is input only port.

VI.1.4 Input Interrupt

The I/O ports support 16 system of port input interrupts and two systems of key input interrupts.

VI.1.4.1 Port Input Interrupt

The port input interrupt circuit has 16 interrupt systems (FPT15–FPT0) and a port can be selected for generating each cause of interrupt.

The interrupt condition can also be selected from between input signal edge and input signal level.

Figure VI.1.4.1.1 shows the configuration of the port input interrupt circuit.

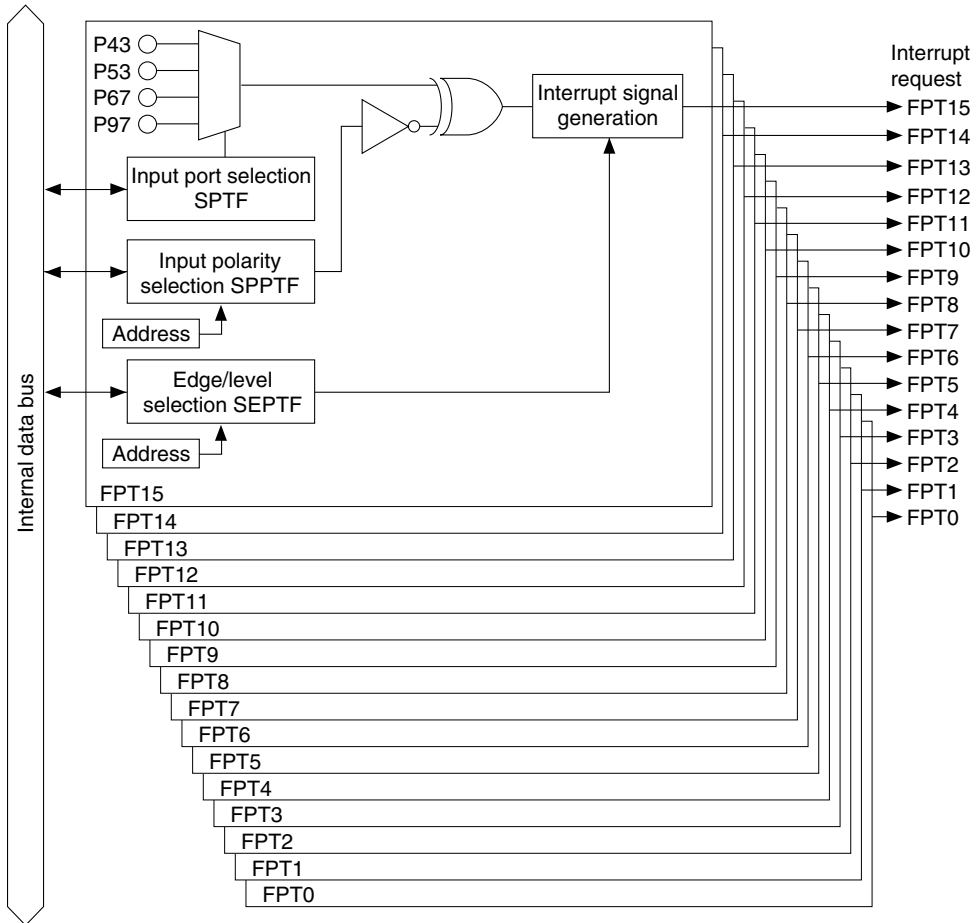


Figure VI.1.4.1.1 Configuration of Port Input Interrupt Circuit

Selecting input pins

The causes of interrupt allow selection of an input pin from the four predefined pins independently. Table VI.1.4.1.1 shows the control bits and the selectable pins for each cause of interrupt.

Table VI.1.4.1.1 Selecting Pins for Port Input Interrupts

Cause of interrupt	Control bit	SPT settings			
		11	10	01	00
FPT15	SPTF[1:0] (D[7:6])/Port input interrupt select register 4 (0x3003C5)	P97	P67	P53	P43
FPT14	SPTE[1:0] (D[5:4])/Port input interrupt select register 4 (0x3003C5)	P96	P66	P52	P42
FPT13	SPTD[1:0] (D[3:2])/Port input interrupt select register 4 (0x3003C5)	P95	P65	P51	P41
FPT12	SPTC[1:0] (D[1:0])/Port input interrupt select register 4 (0x3003C5)	P94	P64	P50	P40
FPT11	SPTB[1:0] (D[7:6])/Port input interrupt select register 3 (0x3003C4)	P93	INT_DCSIO	P83	P73
FPT10	SPTA[1:0] (D[5:4])/Port input interrupt select register 3 (0x3003C4)	P92	INT_USB	P82	P72
FPT9	SPT9[1:0] (D[3:2])/Port input interrupt select register 3 (0x3003C4)	P91	USB_PDREQ	P81	P71
FPT8	SPT8[1:0] (D[1:0])/Port input interrupt select register 3 (0x3003C4)	P90	INT_SPI	P80	P70
FPT7	SPT7[1:0] (D[7:6])/Port input interrupt select register 2 (0x3003C1)	P63	P17	P27	P07
FPT6	SPT6[1:0] (D[5:4])/Port input interrupt select register 2 (0x3003C1)	P62	P16	P26	P06
FPT5	SPT5[1:0] (D[3:2])/Port input interrupt select register 2 (0x3003C1)	P61	P15	P25	P05
FPT4	SPT4[1:0] (D[1:0])/Port input interrupt select register 2 (0x3003C1)	P60	P14	P24	P04
FPT3	SPT3[1:0] (D[7:6])/Port input interrupt select register 1 (0x3003C0)	P33	P13	P23	P03
FPT2	SPT2[1:0] (D[5:4])/Port input interrupt select register 1 (0x3003C0)	P32	P12	P22	P02
FPT1	SPT1[1:0] (D[3:2])/Port input interrupt select register 1 (0x3003C0)	P31	P11	P21	P01
FPT0	SPT0[1:0] (D[1:0])/Port input interrupt select register 1 (0x3003C0)	P30	P10	P20	P00

Note: The FPT8, FPT9, FPT10, and FPT11 interrupt systems are shared with the SPI, USB, and DCSIO interrupts. When using the SPI, USB, or DCSIO interrupts, set the SPT bits to 0b10. In this case, the port input interrupt control registers and signals are used for the SPI, USB, or DCSIO interrupts.

Conditions for port input-interrupt generation

Each port input interrupt can be generated by the edge or level of the input signal. SEPT_x (D_x/0x3003C3, D_x - 8/0x3003C7) is used for this selection. When SEPT_x is set to 1, the FPT_x interrupt will be generated at the signal edge. When SEPT_x is set to 0, the FPT_x interrupt will be generated by the input signal level.

- * **SEPT[7:0]:** FPT_x Edge/Level Select Bits in the Port Input Interrupt Edge/Level Select Register 1 (D_x/0x3003C3)
- * **SEPT[F:8]:** FPT_x Edge/Level Select Bits in the Port Input Interrupt Edge/Level Select Register 2 (D_x - 8/0x3003C7)

Furthermore, the signal polarity can be selected using SPPT_x (D_x/0x3003C2, D_x - 8/0x3003C6).

- * **SPPT[7:0]:** FPT_x Input Polarity Select Bits in the Port Input Interrupt Polarity Select Register 1 (D_x/0x3003C2)
- * **SPPT[F:8]:** FPT_x Input Polarity Select Bits in the Port Input Interrupt Polarity Select Register 2 (D_x - 8/0x3003C6)

With these registers, the port input interrupt condition is decided as shown in Table VI.1.4.1.2.

Table VI.1.4.1.2 Port Input Interrupt Condition

SEPT _x	SPPT _x	FPT _x 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 cause-of-interrupt flag FP_x is set to 1 and, if other interrupt conditions set by the interrupt controller are met, an interrupt is generated.

VI.1.4.2 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 cause of interrupt.

The interrupt condition can also be set by software.

Figure VI.1.4.2.1 shows the configuration of the key input interrupt circuit.

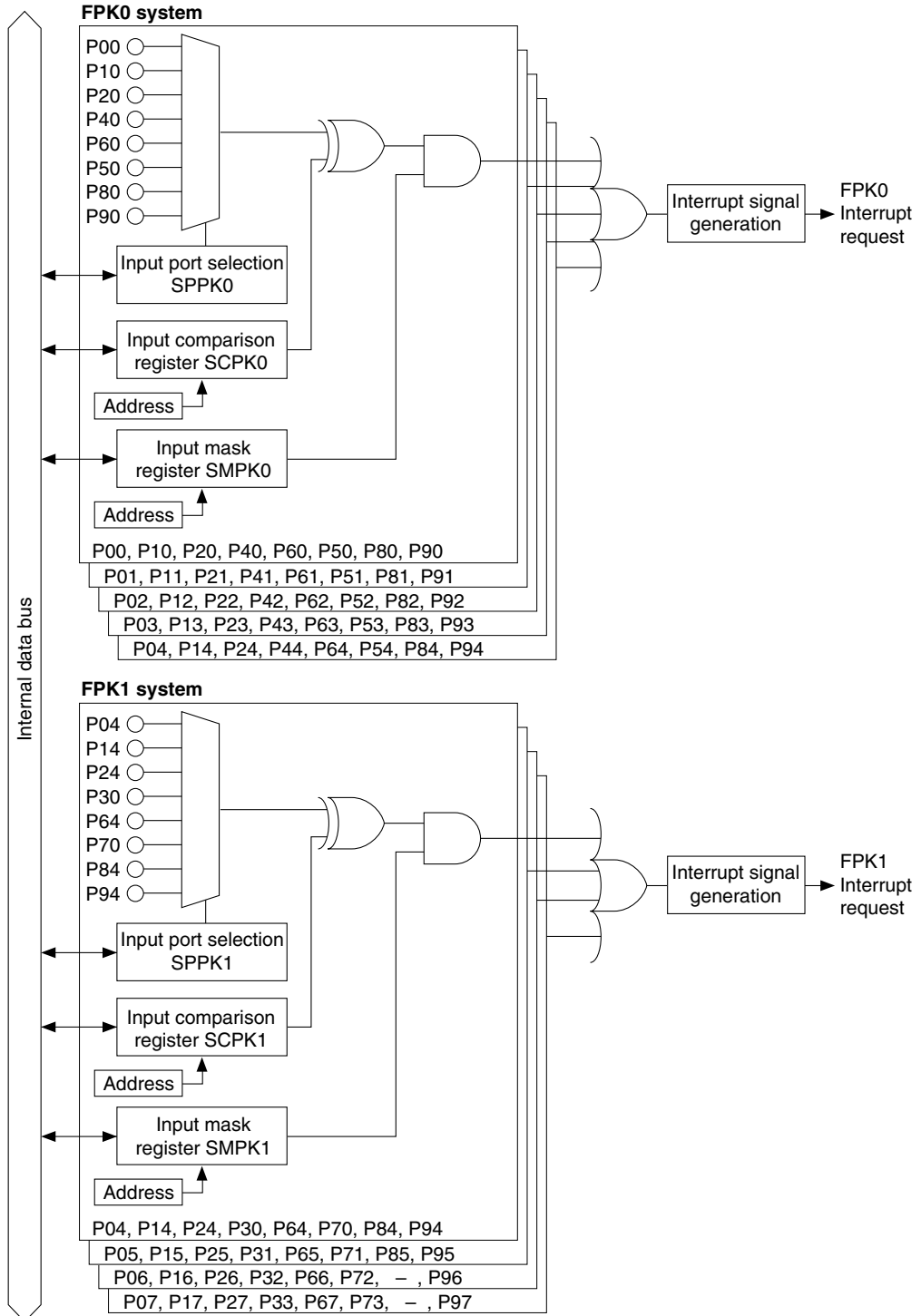


Figure VI.1.4.2.1 Configuration of Key Input Interrupt Circuit

Selecting input pins

For the FPK1 interrupt system, a four-bit input pin group (two input pins when using P8[5:4]) can be selected from the eight predefined groups. For the FPK0 system, a five-bit input pin group can be selected. Table VI.1.4.2.1 shows the control bits and the selectable groups for each cause of interrupt.

Table VI.1.4.2.1 Selecting Pins for Key Input Interrupts

Cause of interrupt	Control bit		SPPK settings							
			111	110	101	100	011	010	001	000
FPK1	SPPK1[2:0] (D[6:4])	Key input interrupt select register (0x3003D0)	P9[7:4]	P8[5:4]	P7[3:0]	P6[7:4]	P3[3:0]	P2[7:4]	P1[7:4]	P0[7:4]
FPK0	SPPK0[2:0] (D[2:0])		P9[4:0]	P8[4:0]	P5[4:0]	P6[4:0]	P4[4:0]	P2[4:0]	P1[4:0]	P0[4:0]

Conditions for key input-interrupt generation

The key input interrupt circuit has the input mask bits SMPK0[4:0] (D[4:0]/0x3003D4) for FPK0 and SMPK1[3:0] (D[3:0]/0x3003D5) for FPK1, and the input comparison bits SCPK0[4:0] (D[4:0]/0x3003D2) for FPK0 and SCPK1[3:0] (D[3:0]/0x3003D3) for FPK1 to set input-interrupt conditions.

- * **SMPK0[4:0]**: FPK0 Input Mask Bits in the Key Input Interrupt (FPK0) Input Mask Register (D[4:0]/0x3003D4)
- * **SMPK1[3:0]**: FPK1 Input Mask Bits in the Key Input Interrupt (FPK1) Input Mask Register (D[3:0]/0x3003D5)
- * **SCPK0[4:0]**: FPK0 Input Comparison Bits in the Key Input Interrupt (FPK0) Input Comparison Register (D[4:0]/0x3003D2)
- * **SCPK1[3:0]**: FPK1 Input Comparison Bits in the Key Input Interrupt (FPK1) Input Comparison Register (D[3:0]/0x3003D3)

The input mask bit (SMPK) is used to mask the input pin that is not used for an interrupt. This bit masks each input pin, whereas the interrupt enable bit of the interrupt controller masks the cause of interrupt for each interrupt group.

The input comparison bit (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 bit (SMPK) and the content of the input comparison bit (SCPK) become unmatched after being matched, the cause-of-interrupt flag (FK) is set to 1 and, if other interrupt conditions are met, an interrupt is generated.

Figure VI.1.4.2.2 shows cases in which a FPK0 interrupt is generated. Here, it is assumed that the P0[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.

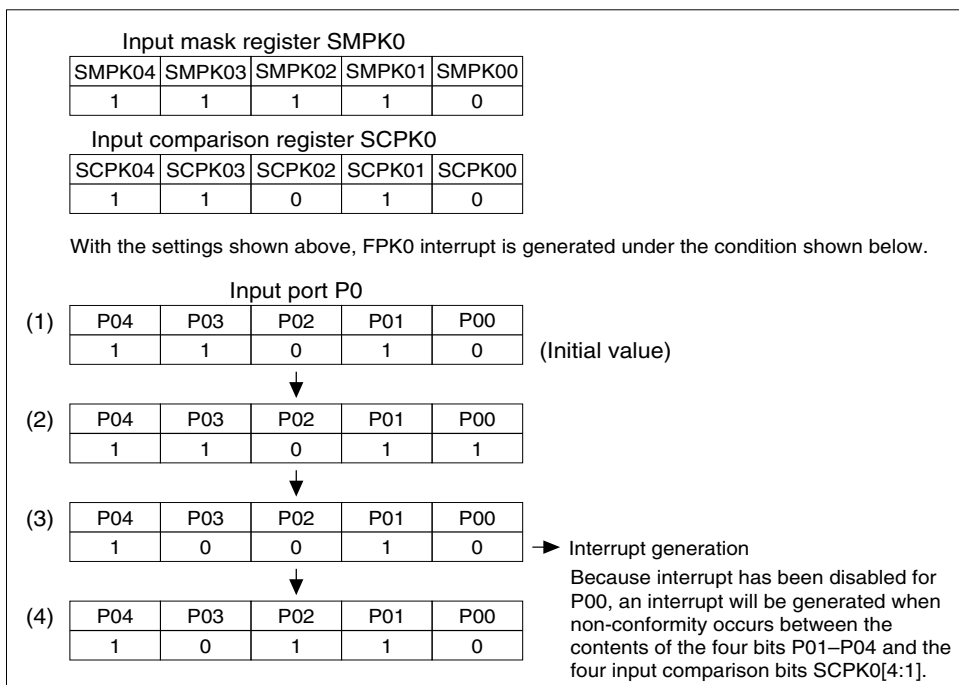


Figure VI.1.4.2.2 FPK0 Interrupt Generation Example (when P0[4:0] is selected by SPPK0[2:0])

Since P00 is masked from interrupt by SMPK00 (D0/0x3003D4), no interrupt occurs at that point (2) above.

Next, because P03 becomes 0 at (3), an interrupt is generated due to the lack of a match between the data of the input pin P0[4:1] that is enabled for interrupt and that of the input comparison register SCPK0[4:1] (D[4:1]/0x3003D2).

Since only a change in states in which the input data and the content of SCPK_x (D[4:0]/0x3003D2, D[3:0]/0x3003D3) 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 SCPK_x, or SCPK_x must be set again. Note that the input pins masked from interrupt by SMPK_x (D[4:0]/0x3003D4, D[3:0]/0x3003D5) do not affect interrupt generation conditions.

An interrupt is generated for FPK1 in the same way as described above.

VI.1.4.3 Control Registers of the Interrupt Controller

Table VI.1.4.3.1 shows the control registers of the interrupt controller that are provided for each input-interrupt system.

Table VI.1.4.3.1 Control Registers of Interrupt Controller

System	Cause-of-interrupt flag	Interrupt enable register	Interrupt priority register
FPT15	FP15(D7/0x3002A9)	EP15(D7/0x3002A6)	PP15L[2:0](D[6:4]/0x3002A3)
FPT14	FP14(D6/0x3002A9)	EP14(D6/0x3002A6)	PP14L[2:0](D[2:0]/0x3002A3)
FPT13	FP13(D5/0x3002A9)	EP13(D5/0x3002A6)	PP13L[2:0](D[6:4]/0x3002A2)
FPT12	FP12(D4/0x3002A9)	EP12(D4/0x3002A6)	PP12L[2:0](D[2:0]/0x3002A2)
FPT11	FP11(D3/0x3002A9)	EP11(D3/0x3002A6)	PP11L[2:0](D[6:4]/0x3002A1)
FPT10	FP10(D2/0x3002A9)	EP10(D2/0x3002A6)	PP10L[2:0](D[2:0]/0x3002A1)
FPT9	FP9(D1/0x3002A9)	EP9(D1/0x3002A6)	PP9L[2:0](D[6:4]/0x3002A0)
FPT8	FP8(D0/0x3002A9)	EP8(D0/0x3002A6)	PP8L[2:0](D[2:0]/0x3002A0)
FPT7	FP7(D6/0x300287)	EP7(D6/0x300277)	PP7L[2:0](D[6:4]/0x30026D)
FPT6	FP6(D5/0x300287)	EP6(D5/0x300277)	PP6L[2:0](D[2:0]/0x30026D)
FPT5	FP5(D4/0x300287)	EP5(D4/0x300277)	PP5L[2:0](D[6:4]/0x30026C)
FPT4	FP4(D3/0x300287)	EP4(D3/0x300277)	PP4L[2:0](D[2:0]/0x30026C)
FPT3	FP3(D3/0x300280)	EP3(D3/0x300270)	PP3L[2:0](D[6:4]/0x300261)
FPT2	FP2(D2/0x300280)	EP2(D2/0x300270)	PP2L[2:0](D[2:0]/0x300261)
FPT1	FP1(D1/0x300280)	EP1(D1/0x300270)	PP1L[2:0](D[6:4]/0x300260)
FPT0	FP0(D0/0x300280)	EP0(D0/0x300270)	PP0L[2:0](D[2:0]/0x300260)
FPK1	FK1(D5/0x300280)	EK1(D5/0x300270)	PK1L[2:0](D[6:4]/0x300262)
FPK0	FK0(D4/0x300280)	EK0(D4/0x300270)	PK0L[2:0](D[2:0]/0x300262)

When the interrupt generation condition described above is met, the corresponding cause-of-interrupt flag is set to 1. If the interrupt enable register bit for that cause of interrupt has been set to 1, an interrupt request is generated.

Interrupts due to a cause of interrupt can be disabled by leaving the interrupt enable register bit for that cause of interrupt set to 0. The cause-of-interrupt 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 Section III.2, "Interrupt Controller (ITC)."

Intelligent DMA

The port input interrupt system can invoke an intelligent DMA (IDMA) through the use of its cause of interrupt. 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 cause of interrupt:

IDMA Ch.	IDMA Ch.
FPT0 input interrupt: 1	FPT8 input interrupt: 38
FPT1 input interrupt: 2	FPT9 input interrupt: 39
FPT2 input interrupt: 3	FPT10 input interrupt: 40
FPT3 input interrupt: 4	FPT11 input interrupt: 41
FPT4 input interrupt: 28	FPT12 input interrupt: 42
FPT5 input interrupt: 29	FPT13 input interrupt: 43
FPT6 input interrupt: 30	FPT14 input interrupt: 44
FPT7 input interrupt: 31	FPT15 input interrupt: 45

For IDMA to be invoked, the IDMA request and IDMA enable bits shown in Table VI.1.4.3.2 must be set to 1 in advance. Transfer conditions, etc. must also be set on the IDMA side in advance.

Table VI.1.4.3.2 Control Bits for IDMA Transfer

System	IDMA request bit	IDMA enable bit
FPT15	RP15(D7/0x3002AC)	DEP15(D7/0x3002AE)
FPT14	RP14(D6/0x3002AC)	DEP14(D6/0x3002AE)
FPT13	RP13(D5/0x3002AC)	DEP13(D5/0x3002AE)
FPT12	RP12(D4/0x3002AC)	DEP12(D4/0x3002AE)
FPT11	RP11(D3/0x3002AC)	DEP11(D3/0x3002AE)
FPT10	RP10(D2/0x3002AC)	DEP10(D2/0x3002AE)
FPT9	RP9(D1/0x3002AC)	DEP9(D1/0x3002AE)
FPT8	RP8(D0/0x3002AC)	DEP8(D0/0x3002AE)
FPT7	RP7(D7/0x300293)	DEP7(D7/0x300297)
FPT6	RP6(D6/0x300293)	DEP6(D6/0x300297)
FPT5	RP5(D5/0x300293)	DEP5(D5/0x300297)
FPT4	RP4(D4/0x300293)	DEP4(D4/0x300297)
FPT3	RP3(D3/0x300290)	DEP3(D3/0x300294)
FPT2	RP2(D2/0x300290)	DEP2(D2/0x300294)
FPT1	RP1(D1/0x300290)	DEP1(D1/0x300294)
FPT0	RP0(D0/0x300290)	DEP0(D0/0x300294)

If the IDMA request and enable bits are set to 1, IDMA is invoked through generation of a cause of interrupt. 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 Section II.2, “Intelligent DMA (IDMA).”

Trap vectors

The trap-vector address of each input default cause of interrupt is set as follows:

FPT0 input interrupt: 0xC00040	FPT7 input interrupt: 0xC0011C
FPT1 input interrupt: 0xC00044	FPT8 input interrupt: 0xC00150
FPT2 input interrupt: 0xC00048	FPT9 input interrupt: 0xC00154
FPT3 input interrupt: 0xC0004C	FPT10 input interrupt: 0xC00158
FPT4 input interrupt: 0xC00050	FPT11 input interrupt: 0xC0015C
FPT5 input interrupt: 0xC00054	FPT12 input interrupt: 0xC00160
FPT6 input interrupt: 0xC00110	FPT13 input interrupt: 0xC00164
FPT7 input interrupt: 0xC00114	FPT14 input interrupt: 0xC00168
FPT8 input interrupt: 0xC00118	FPT15 input interrupt: 0xC0016C

The base address of the trap table can be changed using the TTBR register.

VI.1.5 I/O Port Operating Clock

The GPIO module is clocked by the port operating clocks supplied by the CMU.

The CMU provides the clock paths with a control bit shown below for the GPIO. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) GPIO clock (PORT_CLK)

This clock (MCLK) is used for the GPIO circuit and is required for accessing the GPIO control registers. GPIO_CKE (D8/0x301B04) is used for clock supply control (default: on).

* **GPIO_CKE**: GPIO Normal Clock Control Bit in the Gated Clock Control Register 1 (D8/0x301B04)

(2) GPIO no stop clock (PORT_NOSTOP_CLK)

This clock (MCLK) is used for reading input ports and generating input interrupts. This clock can be automatically turned off in HALT mode (see Section III.1.9.2) by setting GPIONSTP_HCKE (D27/0x301B04) to 0 (default: on).

* **GPIONSTP_HCKE**: GPIO No Stop Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D27/0x301B04)

Note, however, that the GPIO no stop clock is required in HALT mode when using an input interrupt to cancel HALT mode.

For details of the generation and control of the port operating clock, see Section III.1, “Clock Management Unit (CMU).”

Note: The Gated Clock Control Register 1 (0x301B04) is write-protected. Write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless re-writing said registers.

VI.1.6 Details of Control Registers

Table VI.1.6.1 List of I/O Port Registers

Address	Register name	Size	Function
0x00300380	P0 Port Data Register (pP0_P0D)	8	P0 port data read/write register
0x00300381	P0 I/O Control Register (pP0_IOC0)	8	Controls P0 port input/output direction.
0x00300382	P1 Port Data Register (pP1_P1D)	8	P1 port data read/write register
0x00300383	P1 I/O Control Register (pP1_IOC1)	8	Controls P1 port input/output direction.
0x00300384	P2 Port Data Register (pP2_P2D)	8	P2 port data read/write register
0x00300385	P2 I/O Control Register (pP2_IOC2)	8	Controls P2 port input/output direction.
0x00300386	P3 Port Data Register (pP3_P3D)	8	P3 port data read/write register
0x00300387	P3 I/O Control Register (pP3_IOC3)	8	Controls P3 port input/output direction.
0x00300388	P4 Port Data Register (pP4_P4D)	8	P4 port data read/write register
0x00300389	P4 I/O Control Register (pP4_IOC4)	8	Controls P4 port input/output direction.
0x0030038A	P5 Port Data Register (pP5_P5D)	8	P5 port data read/write register
0x0030038B	P5 I/O Control Register (pP5_IOC5)	8	Controls P5 port input/output direction.
0x0030038C	P6 Port Data Register (pP6_P6D)	8	P6 port data read/write register
0x0030038D	P6 I/O Control Register (pP6_IOC6)	8	Controls P6 port input/output direction.
0x0030038E	P7 Port Data Register (pP7_P7D)	8	P7 port data read/write register
0x00300390	P8 Port Data Register (pP8_P8D)	8	P8 port data read/write register
0x00300391	P8 I/O Control Register (pP8_IOC8)	8	Controls P8 port input/output direction.
0x00300392	P9 Port Data Register (pP9_P9D)	8	P9 port data read/write register
0x00300393	P9 I/O Control Register (pP9_IOC9)	8	Controls P9 port input/output direction.
0x003003A0	P00–P03 Port Function Select Register (pP0_03_CFP)	8	Sets P00–P03 port pin function.
0x003003A1	P04–P07 Port Function Select Register (pP0_47_CFP)	8	Sets P04–P07 port pin function.
0x003003A2	P10–P13 Port Function Select Register (pP1_03_CFP)	8	Sets P10–P13 port pin function.
0x003003A3	P14–P17 Port Function Select Register (pP1_47_CFP)	8	Sets P14–P17 port pin function.
0x003003A4	P20–P23 Port Function Select Register (pP2_03_CFP)	8	Sets P20–P23 port pin function.
0x003003A5	P24–P27 Port Function Select Register (pP2_47_CFP)	8	Sets P24–P27 port pin function.
0x003003A6	P30–P33 Port Function Select Register (pP3_03_CFP)	8	Sets P30–P33 port pin function.
0x003003A7	P34–P36 Port Function Select Register (pP3_46_CFP)	8	Sets P34–P36 port pin function.
0x003003A8	P40–P43 Port Function Select Register (pP4_03_CFP)	8	Sets P40–P43 port pin function.
0x003003A9	P44–P47 Port Function Select Register (pP4_47_CFP)	8	Sets P44–P47 port pin function.
0x003003AA	P50–P53 Port Function Select Register (pP5_03_CFP)	8	Sets P50–P53 port pin function.
0x003003AB	P54–P57 Port Function Select Register (pP5_47_CFP)	8	Sets P54–P57 port pin function.
0x003003AC	P60–P63 Port Function Select Register (pP6_03_CFP)	8	Sets P60–P63 port pin function.
0x003003AD	P64–P67 Port Function Select Register (pP6_47_CFP)	8	Sets P64–P67 port pin function.
0x003003AE	P70–P73 Port Function Select Register (pP7_03_CFP)	8	Sets P70–P73 port pin function.
0x003003AF	P74 Port Function Select Register (pP7_4_CFP)	8	Sets P74 port pin function.
0x003003B0	P80–P83 Port Function Select Register (pP8_03_CFP)	8	Sets P80–P83 port pin function.
0x003003B1	P84–P85 Port Function Select Register (pP8_45_CFP)	8	Sets P84–P85 port pin function.
0x003003B2	P90–P93 Port Function Select Register (pP9_03_CFP)	8	Sets P90–P93 port pin function.
0x003003B3	P94–P97 Port Function Select Register (pP9_47_CFP)	8	Sets P94–P97 port pin function.
0x003003C0	Port Input Interrupt Select Register 1 (pPINTSEL_SPT03)	8	Selects ports used for FPT0–FPT3 port input interrupts.
0x003003C1	Port Input Interrupt Select Register 2 (pPINTSEL_SPT47)	8	Selects ports used for FPT4–FPT7 port input interrupts.
0x003003C2	Port Input Interrupt Polarity Select Register 1 (pPINTPOL_SPP07)	8	Selects signal polarity to generate FPT0–FPT7 port input interrupts.
0x003003C3	Port Input Interrupt Edge/Level Select Register 1 (pPINTEL_SEPT07)	8	Selects FPT0–FPT7 port interrupt trigger condition.
0x003003C4	Port Input Interrupt Select Register 3 (pPINTSEL_SPT811)	8	Selects ports used for FPT8–FPT11 port input interrupts.
0x003003C5	Port Input Interrupt Select Register 4 (pPINTSEL_SPT1215)	8	Selects ports used for FPT12–FPT15 port input interrupts.
0x003003C6	Port Input Interrupt Polarity Select Register 2 (pPINTPOL_SPP815)	8	Selects signal polarity to generate FPT8–FPT15 port input interrupts.
0x003003C7	Port Input Interrupt Edge/Level Select Register 2 (pPINTEL_SEPT815)	8	Selects FPT8–FPT15 port interrupt trigger condition.
0x003003D0	Key Input Interrupt Select Register (pKINTSEL_SPPK01)	8	Selects ports used for key input interrupts.
0x003003D2	Key Input Interrupt (FPK0) Input Comparison Register (pKINTCOMP_SCPK0)	8	Sets FPK0 interrupt trigger edge condition.
0x003003D3	Key Input Interrupt (FPK1) Input Comparison Register (pKINTCOMP_SCPK1)	8	Sets FPK1 interrupt trigger edge condition.
0x003003D4	Key Input Interrupt (FPK0) Input Mask Register (pKINTCOMP_SMPK0)	8	Enables/disables ports for generating FPK0 interrupts.
0x003003D5	Key Input Interrupt (FPK1) Input Mask Register (pKINTCOMP_SMPK1)	8	Enables/disables ports for generating FPK1 interrupts.

The following describes each I/O port control register. The I/O port control registers are mapped in the 8-bit device area from 0x300380 to 0x3003D5, and can be accessed in units of bytes.

0x300380–0x300392: Px Port Data Registers (pPx_PxD)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Px port data register (pPx_PxD)	00300380 00300392 (B)	D7	Px7D	Px7 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D6	Px6D	Px6 I/O port data					Ext.	R/W	
		D5	Px5D	Px5 I/O port data					Ext.	R/W	
		D4	Px4D	Px4 I/O port data					Ext.	R/W	
		D3	Px3D	Px3 I/O port data					Ext.	R/W	
		D2	Px2D	Px2 I/O port data					Ext.	R/W	
		D1	Px1D	Px1 I/O port data					Ext.	R/W	
		D0	Px0D	Px0 I/O port data					Ext.	R/W	

Note: The letter 'x' in bit names, etc., denotes a port number from 0 to 6 and 8, 9.

- 0x300380 P0 Port Data Register (pP0_P0D)
- 0x300382 P1 Port Data Register (pP1_P1D)
- 0x300384 P2 Port Data Register (pP2_P2D)
- 0x300386 P3 Port Data Register (pP3_P3D)
- 0x300388 P4 Port Data Register (pP4_P4D)
- 0x30038A P5 Port Data Register (pP5_P5D)
- 0x30038C P6 Port Data Register (pP6_P6D)
- 0x300390 P8 Port Data Register (pP8_P8D)
- 0x300392 P9 Port Data Register (pP9_P9D)

These registers are used to read data from I/O-port pins or to set output data. (Default: external pin status)

1 (R/W): High level

0 (R/W): Low level

When an I/O port is set for output, the data written to the register 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} or V_{DDH} level); if the data is 0, the port pin is set low (V_{SS} level).

Even in input mode, data can be written to the port data register.

When the register is read, 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} or V_{DDH} level), 1 is read out as input data; if the pin voltage is low (V_{SS} level), 0 is read out as input data.

0x30038E: P7 Port Data Register (pP7_P7D)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
P7 port data register (pP7_P7D)	0030038E (B)	D7-5	–	reserved	–		–	–	0 when being read.	
		D4	P74D	P74 input port data	1	High	0	Low	Ext. R	Ext.: The initial value depends on the external pin status.
		D3	P73D	P73 input port data					Ext. R	
		D2	P72D	P72 input port data					Ext. R	
		D1	P71D	P71 input port data					Ext. R	
		D0	P70D	P70 input port data					Ext. R	

This register is used to read data from P7 I/O-port pins. (Default: external pin status)

1 (R): High level

0 (R): Low level

The voltage level on the port pin is read out. If the pin voltage is high (AV_{DD} and V_{DDH} level), 1 is read out as input data; if the pin voltage is low (V_{SS} level), 0 is read out as input data.

Note: The P7 port has no output function, therefore this register is a read only register.

0x300381–0x300393: Px I/O Control Registers (pPx_IOCx)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Px I/O control register (pPx_IOCx)	00300381 00300393 (B)	D7	IOCx7	Px7 I/O control	1	Output	0	Input	0	R/W	
		D6	IOCx6	Px6 I/O control					0	R/W	
		D5	IOCx5	Px5 I/O control					0	R/W	
		D4	IOCx4	Px4 I/O control					0	R/W	
		D3	IOCx3	Px3 I/O control					0	R/W	
		D2	IOCx2	Px2 I/O control					0	R/W	
		D1	IOCx1	Px1 I/O control					0	R/W	
		D0	IOCx0	Px0 I/O control					0	R/W	

Note: The letter 'x' in bit names, etc., denotes a port number from 0 to 6 and 8, 9.

- 0x300381 P0 I/O Control Register (pP0_IOC0)
- 0x300383 P1 I/O Control Register (pP1_IOC1)
- 0x300385 P2 I/O Control Register (pP2_IOC2)
- 0x300387 P3 I/O Control Register (pP3_IOC3)
- 0x300389 P4 I/O Control Register (pP4_IOC4)
- 0x30038B P5 I/O Control Register (pP5_IOC5)
- 0x30038D P6 I/O Control Register (pP6_IOC6)
- 0x300391 P8 I/O Control Register (pP8_IOC8)
- 0x300393 P9 I/O Control Register (pP9_IOC9)

Directs an I/O port for input or output and indicates the I/O control signal value of the port.

1 (R/W): Output mode

0 (R/W): Input mode (default)

Each I/O control register bit corresponds to each I/O port. When IOCx 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.

When the pin is used for a peripheral function, the input/output direction depends on the peripheral function.

When the register is read, the I/O control signal value for the port pin is read out. When I/O port function is selected using the port function select 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 IOCx.

0x3003A0–0x3003B3: Pxx Port Function Select Registers (pPx_xx_CFP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Px0–Px3 port function select register (pPx_03_CFP) or Px4–Px7 port function select register (pPx_47_CFP)	003003A0 003003B3 (B)	D7	CFPx31	Px3/Px7 port extended function	CFPx3/7[1:0]	Function	0	R/W
		D6	CFPx30		11	Pin function 3	0	
			or		10	Pin function 2		
			CFPx71 CFPx70		01 00	Pin function 1 Pin function 0		
	D5	CFPx21	Px2/Px6 port extended function	CFPx2/6[1:0]	Function	0	R/W	
	D4	CFPx20		11	Pin function 3	0		
		or		10	Pin function 2			
		CFPx61 CFPx60		01 00	Pin function 1 Pin function 0			
	D3	CFPx11	Px1/Px5 port extended function	CFPx1/5[1:0]	Function	0	R/W	
	D2	CFPx10		11	Pin function 3	0		
		or		10	Pin function 2			
		CFPx51 CFPx50		01 00	Pin function 1 Pin function 0			
	D1	CFPx01	Px0/Px4 port extended function	CFPx0/4[1:0]	Function	0	R/W	
	D0	CFPx00		11	Pin function 3	0		
		or		10	Pin function 2			
		CFPx41 CFPx40		01 00	Pin function 1 Pin function 0			

Note: The letter ‘x’ in bit names, etc., denotes a port number from 0 to 9.

- 0x3003A0 P00–P03 Port Function Select Register (pP0_03_CFP)
- 0x3003A1 P04–P07 Port Function Select Register (pP0_47_CFP)
- 0x3003A2 P10–P13 Port Function Select Register (pP1_03_CFP)
- 0x3003A3 P14–P17 Port Function Select Register (pP1_47_CFP)
- 0x3003A4 P20–P23 Port Function Select Register (pP2_03_CFP)
- 0x3003A5 P24–P27 Port Function Select Register (pP2_47_CFP)
- 0x3003A6 P30–P33 Port Function Select Register (pP3_03_CFP)
- 0x3003A7 P34–P36 Port Function Select Register (pP3_46_CFP)
- 0x3003A8 P40–P43 Port Function Select Register (pP4_03_CFP)
- 0x3003A9 P44–P47 Port Function Select Register (pP4_47_CFP)
- 0x3003AA P50–P53 Port Function Select Register (pP5_03_CFP)
- 0x3003AB P54–P57 Port Function Select Register (pP5_47_CFP)
- 0x3003AC P60–P63 Port Function Select Register (pP6_03_CFP)
- 0x3003AD P64–P67 Port Function Select Register (pP6_47_CFP)
- 0x3003AE P70–P73 Port Function Select Register (pP7_03_CFP)
- 0x3003AF P74 Port Function Select Register (pP7_4_CFP)
- 0x3003B0 P80–P83 Port Function Select Register (pP8_03_CFP)
- 0x3003B1 P84–P85 Port Function Select Register (pP8_45_CFP)
- 0x3003B2 P90–P93 Port Function Select Register (pP9_03_CFP)
- 0x3003B3 P94–P97 Port Function Select Register (pP9_47_CFP)

These bits select the function of each I/O port pin. (Default: 0b00 = Pin function 0)

The I/O ports concurrently serve as the input/output pins for peripheral circuits or bus signals. Whether they are used as I/O ports or for peripheral circuits/bus signals can be selected bit-for-bit using these registers. All pins not used for peripheral circuits/bus signals can be used as general-purpose I/O ports.

For details of pin functions, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

0x3003C0: Port Input Interrupt Select Register 1 (pPINTSEL_SPT03)**0x3003C1: Port Input Interrupt Select Register 2 (pPINTSEL_SPT47)****0x3003C4: Port Input Interrupt Select Register 3 (pPINTSEL_SPT811)****0x3003C5: Port Input Interrupt Select Register 4 (pPINTSEL_SPT1215)**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input interrupt select register 1 (pPINTSEL_SPT03)	003003C0 (B)	D7	SPT31	FPT3 interrupt input port selection	SPT3[1:0]	Port	0	R/W
		D6	SPT30		11	P33		
					10	P13		
					01	P23		
					00	P03		
		D5	SPT21	FPT2 interrupt input port selection	SPT2[1:0]	Port	0	R/W
		D4	SPT20		11	P32		
					10	P12		
					01	P22		
					00	P02		
		D3	SPT11	FPT1 interrupt input port selection	SPT1[1:0]	Port	0	R/W
		D2	SPT10		11	P31		
					10	P11		
					01	P21		
					00	P01		
		D1	SPT01	FPT0 interrupt input port selection	SPT0[1:0]	Port	0	R/W
D0	SPT00	11	P30					
		10	P10					
		01	P20					
		00	P00					
Port input interrupt select register 2 (pPINTSEL_SPT47)	003003C1 (B)	D7	SPT71	FPT7 interrupt input port selection	SPT7[1:0]	Port	0	R/W
		D6	SPT70		11	P63		
					10	P17		
					01	P27		
					00	P07		
		D5	SPT61	FPT6 interrupt input port selection	SPT6[1:0]	Port	0	R/W
		D4	SPT60		11	P62		
					10	P16		
					01	P26		
					00	P06		
		D3	SPT51	FPT5 interrupt input port selection	SPT5[1:0]	Port	0	R/W
		D2	SPT50		11	P61		
					10	P15		
					01	P25		
					00	P05		
		D1	SPT41	FPT4 interrupt input port selection	SPT4[1:0]	Port	0	R/W
D0	SPT40	11	P60					
		10	P14					
		01	P24					
		00	P04					
Port input interrupt select register 3 (pPINTSEL_SPT811)	003003C4 (B)	D7	SPTB1	FPT11 interrupt input port selection	SPTB[1:0]	Port	0	R/W
		D6	SPTB0		11	P93		
					10	INT_DCSIO		
					01	P83		
					00	P73		
		D5	SPTA1	FPT10 interrupt input port selection	SPTA[1:0]	Port	0	R/W
		D4	SPTA0		11	P92		
					10	INT_USB		
					01	P82		
					00	P72		
		D3	SPT91	FPT9 interrupt input port selection	SPT9[1:0]	Port	0	R/W
		D2	SPT90		11	P91		
					10	USB_PDREQ		
					01	P81		
					00	P71		
		D1	SPT81	FPT8 interrupt input port selection	SPT8[1:0]	Port	0	R/W
D0	SPT80	11	P90					
		10	INT_SPI					
		01	P80					
		00	P70					

VI PERIPHERAL MODULES 4 (PORTS): GENERAL-PURPOSE I/O PORTS (GPIO)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port input interrupt select register 4 (pPINTSEL_SPT1215)	003003C5 (B)	D7 D6	SPTF1 SPTF0	FPT15 interrupt input port selection	SPTF[1:0]	Port	0	R/W
					11	P97	0	
					10	P67		
					01	P53		
		D5 D4	SPTD1 SPTD0	FPT14 interrupt input port selection	SPTD[1:0]	Port	0	R/W
					11	P96	0	
					10	P66		
					01	P52		
		D3 D2	SPTD1 SPTD0	FPT13 interrupt input port selection	SPTD[1:0]	Port	0	R/W
					11	P95	0	
					10	P65		
					01	P51		
		D1 D0	SPTC1 SPTC0	FPT12 interrupt input port selection	SPTC[1:0]	Port	0	R/W
					11	P94	0	
					10	P64		
					01	P50		
					00	P40		

SPTx[1:0]: FPTx Interrupt Input Port Select Bits

Selects an input pin used to generate the FPTx port input interrupt.

Table VI.1.6.2 Selecting Pins for Port Input Interrupts

Interrupt system	SPT settings			
	11	10	01	00
FPT15	P97	P67	P53	P43
FPT14	P96	P66	P52	P42
FPT13	P95	P65	P51	P41
FPT12	P94	P64	P50	P40
FPT11	P93	INT_DCSIO	P83	P73
FPT10	P92	INT_USB	P82	P72
FPT9	P91	USB_PDREQ	P81	P71
FPT8	P90	INT_SPI	P80	P70
FPT7	P63	P17	P27	P07
FPT6	P62	P16	P26	P06
FPT5	P61	P15	P25	P05
FPT4	P60	P14	P24	P04
FPT3	P33	P13	P23	P03
FPT2	P32	P12	P22	P02
FPT1	P31	P11	P21	P01
FPT0	P30	P10	P20	P00

(Default: 0b00)

Note: The FPT8, FPT9, FPT10, and FPT11 interrupt systems are shared with the SPI, USB, and DCSIO interrupts. When using the SPI, USB, or DCSIO interrupts, set the SPT bits to 0b10. In this case, the port input interrupt control registers and signals are used for the SPI, USB, or DCSIO interrupts.

0x3003C2: Port Input Interrupt Polarity Select Register 1 (pPINTPOL_SPP07)

0x3003C6: Port Input Interrupt Polarity Select Register 2 (pPINTPOL_SPP815)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Port input interrupt polarity select register 1 (pPINTPOL_SPP07)	003003C2 (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 polarity select register 2 (pPINTPOL_SPP815)	003003C6 (B)	D7	SPPTF	FPT15 input polarity selection	1 High level or Rising edge	0 Low level or Falling edge	1	R/W	
		D6	SPPTE	FPT14 input polarity selection			1	R/W	
		D5	SPPTD	FPT13 input polarity selection			1	R/W	
		D4	SPPTC	FPT12 input polarity selection			1	R/W	
		D3	SPPTB	FPT11 input polarity selection			1	R/W	
		D2	SPPTA	FPT10 input polarity selection			1	R/W	
		D1	SPPT9	FPT9 input polarity selection			1	R/W	
		D0	SPPT8	FPT8 input polarity selection			1	R/W	

These registers are used to select the input signal polarity for generating port input interrupts.

1 (R/W): High level or Rising edge (default)

0 (R/W): Low level or Falling edge

SPPT x is the input polarity select bit corresponding to the FPT x interrupt. When SPPT x is set to 1, the FPT x interrupt will be generated by a high level input or at the rising edge. When SPPT x 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 SEPT x (0x3003C3, 0x3003C7).

D[7:0]/0x3003C2 SPPT[7:0]: FPT[7:0] Interrupt Polarity Select Bits

Selects input signal polarity to generate an FPT[7:0] interrupt.

D[7:0]/0x3003C6 SPPT[F:8]: FPT[15:8] Interrupt Polarity Select Bits

Selects input signal polarity to generate an FPT[15:8] interrupt.

0x3003C3: Port Input Interrupt Edge/Level Select Register 1 (pPINTEL_SEPT07)

0x3003C7: Port Input Interrupt Edge/Level Select Register 2 (pPINTEL_SEPT815)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Port input interrupt edge/level select register 1 (pPINTEL_SEPT07)	003003C3 (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	
Port input interrupt edge/level select register 2 (pPINTEL_SEPT815)	003003C7 (B)	D7	SEPTF	FPT15 edge/level selection	1	Edge	0	Level	1	R/W	
		D6	SEPT E	FPT14 edge/level selection					1	R/W	
		D5	SEPTD	FPT13 edge/level selection					1	R/W	
		D4	SEPTC	FPT12 edge/level selection					1	R/W	
		D3	SEPTB	FPT11 edge/level selection					1	R/W	
		D2	SEPTA	FPT10 edge/level selection					1	R/W	
		D1	SEPT9	FPT9 edge/level selection					1	R/W	
		D0	SEPT8	FPT8 edge/level selection					1	R/W	

These registers are used to select an edge trigger or a level sense condition for generating port input interrupts.

1 (R/W): Edge (default)

0 (R/W): Level

SEPT_x is the edge/level select bit corresponding to the FPT_x interrupt. When SEPT_x is set to 1, the FPT_x interrupt will be generated at the signal edge. Either falling edge or rising edge can be selected by SPPT_x (0x3003C2, 0x3003C6). When SEPT_x is set to 0, the interrupt will be generated by the level (high or low) specified with SPPT_x.

D[7:0]/0x3003C3 SEPT[7:0]: FPT[7:0] Edge/Level Select Bits

Selects an edge trigger or a level sense for the FPT[7:0] interrupt.

D[7:0]/0x3003C7 SEPT[F:8]: FPT[15:8] Edge/Level Select Bits

Selects an edge trigger or a level sense for the FPT[15:8] interrupt.

0x3003D0: Key Input Interrupt Select Register (pKINTSEL_SPPK01)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Key input interrupt select register (pKINTSEL_SPPK01)	003003D0 (B)	D7	–	reserved		–	–	0 when being read.	
		D6	SPPK12	FPK1 interrupt input port selection	SPPK1[2:0]	Port	0	R/W	
		D5	SPPK11		111	P9[7:4]	0		
		D4	SPPK10		110	P8[5:4]	0		
					101	P7[3:0]	0		
					100	P6[7:4]			
					011	P3[3:0]			
					010	P2[7:4]			
					001	P1[7:4]			
		000	P0[7:4]						
		D3	–	reserved		–	–	0 when being read.	
		D2	SPPK02	FPK0 interrupt input port selection	SPPK0[2:0]	Port	0	R/W	
		D1	SPPK01		111	P9[4:0]	0		
		D0	SPPK00		110	P8[4:0]	0		
101	P5[4:0]								
100	P6[4:0]								
011	P4[4:0]								
010	P2[4:0]								
001	P1[4:0]								
000	P0[4:0]								

This register is used to select an input-pin group for generating key interrupts.

Table VI.1.6.3 Selecting Pins for Key Input Interrupts

Interrupt system	SPPK settings							
	111	110	101	100	011	010	001	000
FPK1	P9[7:4]	P8[5:4]	P7[3:0]	P6[7:4]	P3[3:0]	P2[7:4]	P1[7:4]	P0[7:4]
FPK0	P9[4:0]	P8[4:0]	P5[4:0]	P6[4:0]	P4[4:0]	P2[4:0]	P1[4:0]	P0[4:0]

(Default: 0b000)

D7 **Reserved**

D[6:4] **SPPK1[2:0]: FPK1 Interrupt Input Port Select Bits**

Selects an input-pin group for the FPK1 interrupt.

D3 **Reserved**

D[2:0] **SPPK0[2:0]: FPK0 Interrupt Input Port Select Bits**

Selects an input-pin group for the FPK0 interrupt.

**0x3003D2: Key Input Interrupt (FPK0) Input Comparison Register
(pKINTCOMP_SCPK0)****0x3003D3: Key Input Interrupt (FPK1) Input Comparison Register
(pKINTCOMP_SCPK1)**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Key input interrupt (FPK0) input comparison register (pKINTCOMP_SCPK0)	003003D2 (B)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	SCPCK04	FPK04 input comparison	1 High 0 Low	0	R/W	
		D3	SCPCK03	FPK03 input comparison		0	R/W	
		D2	SCPCK02	FPK02 input comparison		0	R/W	
		D1	SCPCK01	FPK01 input comparison		0	R/W	
D0	SCPCK00	FPK00 input comparison	0	R/W				
Key input interrupt (FPK1) input comparison register (pKINTCOMP_SCPK1)	003003D3 (B)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	SCPCK13	FPK13 input comparison	1 High 0 Low	0	R/W	
		D2	SCPCK12	FPK12 input comparison		0	R/W	
		D1	SCPCK11	FPK11 input comparison		0	R/W	
		D0	SCPCK10	FPK10 input comparison		0	R/W	

D[4:0]/0x3003D2 SCPK0[4:0]: FPK0[4:0] Input Comparison Bits

Sets the conditions for generating FPK0 key-input interrupts (timing of interrupt generation).

1 (R/W): Falling edge

0 (R/W): Rising edge (default)

SCPCK0[4:0] is compared with the input state of five bits of the FPK0 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 SMPK0[4:0] (D[4:0]/0x3003D4)).

D[3:0]/0x3003D3 SCPK1[3:0]: FPK1[3:0] Input Comparison Bits

Sets the conditions for generating FPK1 key-input interrupts (timing of interrupt generation).

1 (R/W): Falling edge

0 (R/W): Rising edge (default)

SCPCK1[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 SMPK1[3:0] (D[3:0]/0x3003D5)).

0x3003D4: Key Input Interrupt (FPK0) Input Mask Register (pKINTCOMP_SMPK0)

0x3003D5: Key Input Interrupt (FPK1) Input Mask Register (pKINTCOMP_SMPK1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Key input interrupt (FPK0) input mask register (pKINTCOMP_SMPK0)	003003D4 (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 (pKINTCOMP_SMPK1)	003003D5 (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

D[4:0]/0x3003D4 SMPK0[4:0]: FPK0[4:0] Input Mask Bits

Sets conditions for generating FPK0 key-input interrupts (interrupt enabled/disabled).

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

SMPK0 x is an input mask bit for each FPK0 key-input interrupt port. 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.

D[3:0]/0x3003D5 SMPK1[3:0]: FPK1[3:0] Input Mask Bits

Sets conditions for generating FPK1 key-input interrupts (interrupt enabled/disabled).

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

SMPK1 x is an input mask bit for each FPK1 key-input interrupt port. 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.

VI.1.7 Precautions

- After an initial reset, the cause-of-interrupt flags become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset the flags in a program.
- To prevent regeneration of interrupts due to the same cause of interrupt following the occurrence of an interrupt, always be sure to reset the cause-of-interrupt flag before resetting the PSR or executing the reti instruction.
- When using an port input interrupt as the trigger to restart from the 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 SLEEP mode using a port input interrupt.

- To use the P15–P17 and P34–P36 pins that are configured as the debug interface pins by default for general-purpose inputs/outputs, clear TRCMUX (D0/0x300014) to 0.

* **TRCMUX**: P15–17, P34–36 Debug Function Select Bit in the Debug Port MUX Register (D0/0x300014)

Note, however, that the PC trace function of the debugger cannot be used when TRCMUX (D0/0x300014) is set to 0.

- Even if the port input interrupt condition is set to falling edge, the input pulse width must be longer than 1 cycle of the port operating clock (= MCLK) to be certain an interrupt will be generated.

VI.2 Extended General-Purpose I/O Ports (EGPIO)

VI.2.1 Structure of EGPIO Port

The S1C33E08 contains 17 extended I/O ports (PA[4:0], PB[3:0], and PC[7:0]) to implement extended peripheral functions that cannot be included in the standard GPIO module. When the extended functions assigned to the I/O pins are not used, the I/O pins can be used as general-purpose I/O ports.

Figure VI.2.1.1 shows the structure of a typical I/O port.

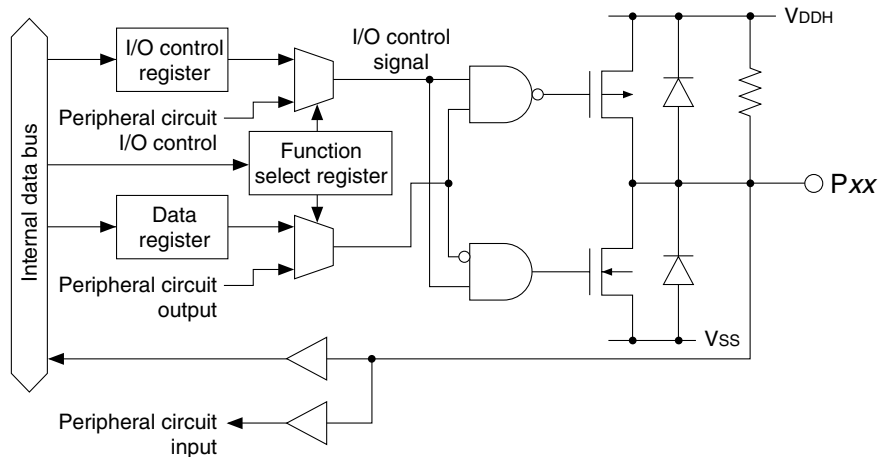


Figure VI.2.1.1 Structure of I/O Port

- Notes:**
- The PA[4:0] and PB[3:0] ports are not available in the QFP24-144pin package model.
 - A pull-up resistor is provided for PA and PB port pins and it can be enabled/disabled by software control. Refer to Section III.4.4, “Pin Control Registers,” for how to control the pull-up resistor.
When the port is in output mode, the port pin is not pulled up regardless of how the pull-up control bit is set.

VI.2.2 Selecting the I/O Pin Functions

The I/O ports concurrently serve as the input/output pins for peripheral circuits or bus signals. Whether they are used as I/O ports or for peripheral circuits/bus signals can be selected bit-for-bit using the port function select registers. All pins not used for peripheral circuits/bus signals can be used as general-purpose I/O ports.

Each I/O port pin (Pxx) is initialized for a default function at initial reset.

For the pin that has two or more functions assigned, the port extended function select bits (CFPxx[1:0]) provided for each I/O port pin can be used to select the desired function.

For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

The subsequent sections explain the port functions assuming that the pin has been set as a general-purpose I/O port.

VI.2.3 I/O Control Register and I/O Modes

The I/O ports are directed for input or output modes by writing data to IOC_x corresponding to each port bit.

- * **IOCA[4:0]**: PA4–PA0 I/O Control Bits in the PA I/O Control Register (D[4:0]/0x300C00)
- * **IOCB[3:0]**: PB3–PB0 I/O Control Bits in the PB I/O Control Register (D[3:0]/0x300C02)
- * **IOCC[7:0]**: PC7–PC0 I/O Control Bits in the PC I/O Control Register (D[7:0]/0x300C04)

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. The port pin is pulled up when the pull-up resistor is enabled using the pin control register.

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_{DDH} 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_{DDH} level); when the data is 0, the port outputs a low level (V_{SS} level). When the port is in output mode, the port pin is not pulled up even if the pull-up resistor is enabled.

At initial reset, the I/O control register is set to 0 (input mode).

VI.2.4 EGPIO Operating Clock

The EGPIO module is clocked by the EGPIO_CLK clock (= MCLK) supplied by the CMU. When initially reset, this clock is selected for supply to the EGPIO port. However, when all the EGPIO ports are idle or not in use, clock supply from the CMU may be turned off to reduce current consumed on the chip. Use EGPIO_CKE (D12/0x301B04) of the CMU for this control.

* **EGPIO_CKE**: EGPIO Clock Control Bit in the Gated Clock Control Register 1 (D12/0x301B04)

Setting EGPIO_CKE (D12/0x301B04) to 0 stops clock supply from the CMU to the EGPIO port.

When the CMU stop supplying the clock to the EGPIO port, all the EGPIO port registers are disabled for writing. However, input pin levels can be read correctly.

For details of the generation and control of the clock, see Section III.1, “Clock Management Unit (CMU).”

Note: The Gated Clock Control Register 1 (0x301B04) is write-protected. Write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless rewriting said registers.

VI.2.5 Details of Control Registers

Table VI.2.5.1 List of EGPIO Registers

Address	Register name	Size	Function
0x00300C00	PA I/O Control Register (pPA_IOC)	8	Controls PA port input/output direction.
0x00300C01	PA Port Data Register (pPA_DATA)	8	PA port data read/write register
0x00300C02	PB I/O Control Register (pPB_IOC)	8	Controls PB port input/output direction.
0x00300C03	PB Port Data Register (pPB_DATA)	8	PB port data read/write register
0x00300C04	PC I/O Control Register (pPC_IOC)	8	Controls PC port input/output direction.
0x00300C05	PC Port Data Register (pPC_DATA)	8	PC port data read/write register
0x00300C20	PA0–PA3 Port Function Select Register (pPA_CFP0)	8	Sets PA0–PA3 port pin function.
0x00300C21	PA4 Port Function Select Register (pPA_CFP1)	8	Sets PA4 port pin function.
0x00300C22	PB0–PB3 Port Function Select Register (pPB_CFP0)	8	Sets PB0–PB3 port pin function.
0x00300C24	PC0–PC3 Port Function Select Register (pPC_CFP0)	8	Sets PC0–PC3 port pin function.
0x00300C25	PC4–PC7 Port Function Select Register (pPC_CFP1)	8	Sets PC4–PC7 port pin function.

The following describes each EGPIO control register. The EGPIO control registers are mapped in the 8-bit device area from 0x300C00 to 0x300C25, and can be accessed in units of bytes.

0x300C00–0x300C04: Px I/O Control Registers (pPx_IOC)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Px I/O control register (pPx_IOC)	00300C00 00300C04 (B)	D7	IOCx7	Px7 I/O control	1	Output	0	Input	0	R/W	
		D6	IOCx6	Px6 I/O control					0	R/W	
		D5	IOCx5	Px5 I/O control					0	R/W	
		D4	IOCx4	Px4 I/O control					0	R/W	
		D3	IOCx3	Px3 I/O control					0	R/W	
		D2	IOCx2	Px2 I/O control					0	R/W	
		D1	IOCx1	Px1 I/O control					0	R/W	
		D0	IOCx0	Px0 I/O control					0	R/W	

Note: The letter 'x' in bit names, etc., denotes a port number from A to C.

0x300C00 PA I/O Control Register (pPA_IOC)

0x300C02 PB I/O Control Register (pPB_IOC)

0x300C04 PC I/O Control Register (pPC_IOC)

Directs an I/O port for input or output and indicates the I/O control signal value of the port.

1 (R/W): Output mode

0 (R/W): Input mode (default)

Each I/O control register bit corresponds to each I/O port. When IOC_x 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.

When the pin is used for a peripheral function, the input/output direction depends on the peripheral function.

When the register is read, the I/O control signal value for the port pin is read out. When I/O port function is selected using the port function select 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 IOC_x.

0x300C01–0x300C05: Px Port Data Registers (pPx_DATA)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Px port data register (pPx_DATA)	00300C01	D7	Px7D	Px7 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D6	Px6D	Px6 I/O port data					Ext.	R/W	
	00300C05 (B)	D5	Px5D	Px5 I/O port data					Ext.	R/W	
		D4	Px4D	Px4 I/O port data					Ext.	R/W	
		D3	Px3D	Px3 I/O port data					Ext.	R/W	
		D2	Px2D	Px2 I/O port data					Ext.	R/W	
		D1	Px1D	Px1 I/O port data					Ext.	R/W	
		D0	Px0D	Px0 I/O port data					Ext.	R/W	

Note: The letter 'x' in bit names, etc., denotes a port number from A to C.

0x300C01 PA Port Data Register (pPA_DATA)

0x300C03 PB Port Data Register (pPB_DATA)

0x300C05 PC Port Data Register (pPC_DATA)

These registers are used to read data from I/O-port pins or to set output data. (Default: external pin status)

1 (R/W): High level

0 (R/W): Low level

When an I/O port is set for output, the data written to the register 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_{DDH} level); if the data is 0, the port pin is set low (V_{SS} level).

Even in input mode, data can be written to the port data register.

When the register is read, 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_{DDH} level), 1 is read out as input data; if the pin voltage is low (V_{SS} level), 0 is read out as input data.

0x300C20–0x300C25: P_x Port Function Select Registers (pP_x_CFP0/1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P _x 0–P _x 3 port function select register (pP _x _CFP0) or P _x 4–P _x 7 port function select register (pP _x _CFP1)	00300C20 00300C25 (B)	D7	CFP_x31	P _x 3/P _x 7 port extended function	CFP _x 3/7[1:0]	Function	0	R/W	
		D6	CFP_x30 or CFP_x71 CFP_x70		11	Pin function 3	0		
					10	Pin function 2			
					01	Pin function 1			
					00	Pin function 0			
			D5	CFP_x21	P _x 2/P _x 6 port extended function	CFP _x 2/6[1:0]	Function	0	R/W
	D4	CFP_x20 or CFP_x61 CFP_x60	11	Pin function 3		0			
			10	Pin function 2					
			01	Pin function 1					
					00	Pin function 0			
			D3	CFP_x11	P _x 1/P _x 5 port extended function	CFP _x 1/5[1:0]	Function	0	R/W
	D2	CFP_x10 or CFP_x51 CFP_x50	11	Pin function 3		0			
			10	Pin function 2					
			01	Pin function 1					
					00	Pin function 0			
			D1	CFP_x01	P _x 0/P _x 4 port extended function	CFP _x 0/4[1:0]	Function	0	R/W
D0	CFP_x00 or CFP_x41 CFP_x40	11	Pin function 3	0					
		10	Pin function 2						
		01	Pin function 1						
				00	Pin function 0				

Note: The letter ‘x’ in bit names, etc., denotes a port number from A to C.

0x300C20 PA0–PA3 Port Function Select Register (pPA_CFP0)

0x300C21 PA4 Port Function Select Register (pPA_CFP1)

0x300C22 PB0–PB3 Port Function Select Register (pPB_CFP0)

0x300C24 PC0–PC3 Port Function Select Register (pPC_CFP0)

0x300C25 PC4–PC7 Port Function Select Register (pPC_CFP1)

These bits select the function of each I/O port pin. (Default: 0b00 = Pin function 0)

The I/O ports concurrently serve as the input/output pins for peripheral circuits or bus signals. Whether they are used as I/O ports or for peripheral circuits/bus signals can be selected bit-for-bit using these registers. All pins not used for peripheral circuits/bus signals can be used as general-purpose I/O ports.

For details of pin functions, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

THIS PAGE IS BLANK.

S1C33E08 Technical Manual

VII PERIPHERAL MODULES 5 (ANALOG)

VII.1 A/D Converter (ADC)

VII.1.1 Features and Structure of A/D Converter

The S1C33E08 contains an A/D converter with the following features:

- Conversion method: Successive comparison
- Resolution: 10 bits
- Input channels: 5 channels
- A/D converter input clock: Maximum of 2 MHz, minimum of 16 kHz
- Conversion time: Minimum of 10 μ s (when a 2-MHz input clock is selected)
Maximum of 1250 μ s (when a 16-kHz input clock is selected)
- Conversion range: Between V_{SS} and AV_{DD}
- 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.
- Three 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 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 S1C33E08 has two operating modes, standard mode of which functions are compatible with the C33 STD analog block for the existing models and an advanced mode allowing use of the extended functions.

Figure VII.1.1.1 shows the structure of the A/D converter.

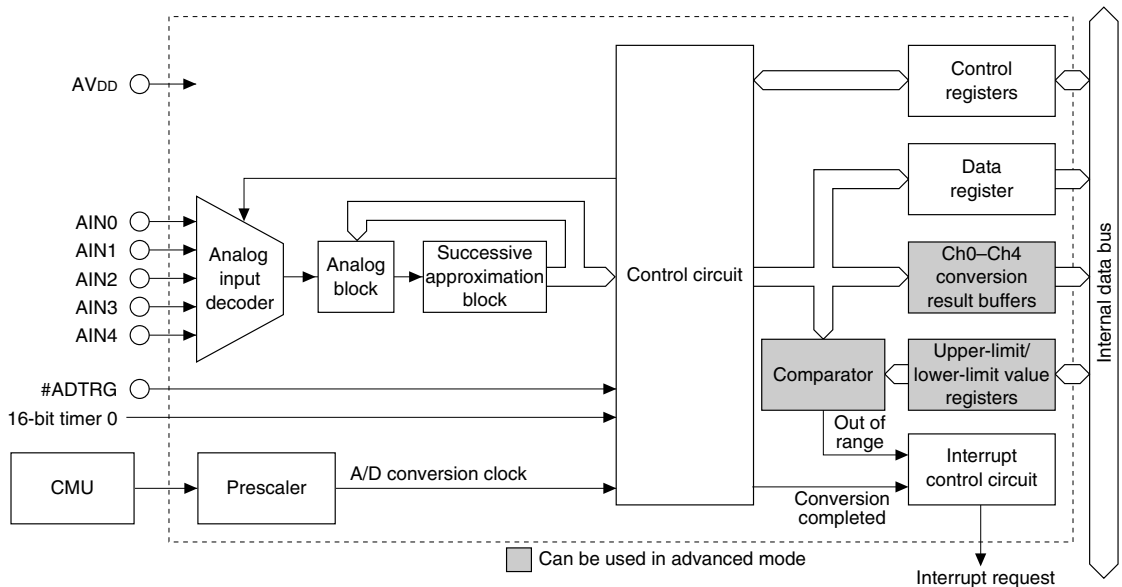


Figure VII.1.1.1 Structure of A/D Converter

VII.1.2 Input Pins of A/D Converter

Table VII.1.2.1 shows the pins used by the A/D converter.

Table VII.1.2.1 Input Pins of A/D Converter

Pin name	I/O	Function
#ADTRG	I	A/D trigger
AIN0	I	A/D converter input 0
AIN1	I	A/D converter input 1
AIN2	I	A/D converter input 2
AIN3	I	A/D converter input 3
AIN4	I	A/D converter input 4
AVDD	–	Analog power supply voltage (+)

AVDD (analog power-supply pin)

AVDD is the power-supply pin for the analog circuit.

Note: When the A/D converter is enabled, a current flows between AVDD 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 disabled (default 0 setting of ADE (D2/0x300544)).

* **ADE:** A/D Enable Bit in the A/D Control/Status Register (D2/0x300544)

AIN[4:0] (analog-signal input pins)

The analog input pins AIN4 (Ch.4) through AIN0 (Ch.0) are shared with I/O ports. 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 initial reset, all these pins are set for I/O ports.

The analog input voltage AVIN can be input in the range of $V_{SS} \leq AV_{IN} \leq AV_{DD}$.

#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 I/O port, 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 initial reset, this pin is set for I/O port.

Note: The A/D input pins are shared with general-purpose I/O ports or other peripheral circuit inputs/outputs, so that functionality in the initial state is set to other than the A/D input. Before the A/D converter can be used, the function of these pins must be switched for the analog input by setting the corresponding Port Function Select Registers.

For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

VII.1.3 A/D Converter Operating Clock

The A/D converter use the ADC_CLK clock (= MCLK) generated by the CMU as the operating clock. The conversion clock is generated in the A/D converter module.

Controlling the supply of the operating clock

ADC_CLK is supplied to the A/D converter with default settings. It can be turned off using ADC_CKE (D3/0x301B04) to reduce the amount of power consumed on the chip if the A/D converter is not used.

* **ADC_CKE:** A/D Converter Clock Control Bit in the Gated Clock Control Register 1 (D3/0x301B04)

Setting ADC_CKE (D3/0x301B04) to 0 (1 by default) turns off the clock supply to the A/D converter. When the clock supply is turned off, the A/D converter control registers cannot be accessed.

For details on how to set and control the clock, refer to Section III.1, “Clock Management Unit (CMU).”

Note: The Gated Clock Control Register 1 (0x301B04) is write-protected. Write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless rewriting said registers.

Clock state in standby mode

The clock supply to the A/D converter stops depending on type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

SLEEP mode: The operating clock supply stops.

Therefore, the A/D converter also stops operating in SLEEP mode.

VII.1.4 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 ... See Sections VII.1.2 and I.3.3.
2. Setting the operating mode (standard 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 ... See Section VII.1.6.

Note: Before making these settings, make sure the A/D converter is disabled (ADE (D2/0x300544) = 0). Changing the settings while the A/D converter is enabled could cause a malfunction.

* **ADE:** A/D Enable Bit in the A/D Control/Status Register (D2/0x300544)

Setting the operating mode (standard mode / advanced mode)

The A/D converter of the S1C33E08 has two operating modes, standard mode of which functions are compatible with the C33 STD analog block for the existing models and an advanced mode allowing use of the extended functions. Table VII.1.4.1 shows differences between the standard mode and the advanced mode.

Table VII.1.4.1 Differences Between Standard Mode and Advanced Mode

Function	Standard 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 (D8/0x30055E) 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 standard mode.

* **ADCADV:** Standard/Advanced Mode Select Bit in the A/D Converter Mode Select/Internal Status Register (D8/0x30055E)

The following descriptions unless otherwise specified are common contents for both modes.

The extended functions in the advanced mode are explained assuming that ADCADV (D8/0x30055E) has been set to 1.

Setting the input clock

The A/D converter contains a prescaler and the A/D conversion clock can be selected from among the eight types shown in Table VII.1.4.2 below. Use PSAD[2:0] (D[2:0]/0x300520) for this selection.

* **PSAD[2:0]**: A/D Converter Clock Division Ratio Setup Bits in the A/D Clock Control Register (D[2:0]/0x300520)

Table VII.1.4.2 Input Clock Selection

PSAD2	PSAD1	PSAD0	Division ratio
1	1	1	MCLK/256
1	1	0	MCLK/128
1	0	1	MCLK/64
1	0	0	MCLK/32
0	1	1	MCLK/16
0	1	0	MCLK/8
0	0	1	MCLK/4
0	0	0	MCLK/2

(Default: 0b000 = MCLK/2)

The selected clock is output from the prescaler by writing 1 to PSONAD (D3/0x300520).

* **PSONAD**: A/D Converter Clock Control Bit in the A/D Clock Control Register (D3/0x300520)

- Notes:**
- 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 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 using CS[2:0] (D[10:8]/0x300542) and CE[2:0] (D[13:11]/0x300542) respectively.

* **CS[2:0]**: A/D Converter Start Channel Setup Bits in the A/D Trigger/Channel Select Register (D[10:8]/0x300542)

* **CE[2:0]**: A/D Converter End Channel Setup Bits in the A/D Trigger/Channel Select Register (D[13:11]/0x300542)

Table VII.1.4.3 Relationship between CS/CE and Input Channel

CS2/CE2	CS1/CE1	CS0/CE0	Channel selected
1	0	0	AIN4
0	1	1	AIN3
0	1	0	AIN2
0	0	1	AIN1
0	0	0	AIN0
Other			Reserved

Example: Operation of one A/D conversion

CS[2:0] = 0, CE[2:0] = 0: Converted only in AIN0

CS[2:0] = 0, CE[2:0] = 3: Converted in the following order: AIN0→AIN1→AIN2→AIN3

CS[2:0] = 3, CE[2:0] = 1: Converted in the following order: AIN3→AIN4→AIN0→AIN1

Note: Only conversion-channel input pins that have been set for use with the A/D converter can be set using CS[2:0] (D[10:8]/0x300542) and CE[2:0] (D[13:11]/0x300542).

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 (D5/0x300542).

* **MS:** A/D Conversion Mode Select Bit in the A/D Trigger/Channel Select Register (D5/0x300542)

1. Normal mode (MS = 0)

All inputs in the range of channels set using CS[2:0] (D[10:8]/0x300542) and CE[2:0] (D[13:11]/0x300542) are A/D converted once and then stopped.

2. Continuous mode (MS = 1)

A/D conversions in the range of channels set using CS[2:0] and CE[2:0] 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]/0x300542) to select a trigger to start A/D conversion from among the three types shown in Table VII.1.4.4.

* **TS[1:0]:** A/D Conversion Trigger Select Bits in the A/D Trigger/Channel Select Register (D[4:3]/0x300542)

Table VII.1.4.4 Trigger Selection

TS1	TS0	Trigger
1	1	External trigger (#ADTRG)
1	0	Reserved
0	1	16-bit 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 #ADTRG pin must be set in advance using the port function select register. A/D conversion is started when a low level of the #ADTRG signal is detected.

2. 16-bit timer

The comparison match B signal of the 16-bit timer 0 is used as a trigger. Since the cycle can be programmed using the timer, this trigger is effective when cyclic A/D conversions are required.

For details on how to set the timer, refer to the explanation of the 16-bit timer in this manual.

3. Software trigger

Writing 1 to ADST (D1/0x300544) in the software serves as a trigger to start A/D conversion.

* **ADST:** A/D Conversion Control/Status Bit in the A/D Control/Status Register (D1/0x300544)

Setting the sampling time

The A/D converter contains ST[1:0] (D[9:8]/0x300544) that allows the analog-signal input sampling time to be set in four steps (3, 5, 7, or 9 times the conversion clock period).

However, this register should be used as set by default (ST[1:0] = 11; x9 clock periods).

* **ST[1:0]:** Input Signal Sampling Time Setup Bits in the A/D Control/Status Register (D[9:8]/0x300544)

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 ADCMP[2:0] (D[14:12]/0x300544).

- * **ADCMP[2:0]**: A/D Upper/Lower-limit Comparison Channel Select Bits in the A/D Control/Status Register (D[14:12]/0x300544)

Table VII.1.4.5 Selecting the Channel for Checking Conversion Results

ADCMP2	ADCMP1	ADCMP0	Channel selected
1	0	0	AIN4
0	1	1	AIN3
0	1	0	AIN2
0	0	1	AIN1
0	0	0	AIN0
Other			Reserved

2. Setting upper-limit and lower-limit values

Set the upper-limit value to ADUPR[9:0] (D[9:0]/0x300558) and the lower-limit value to ADLWR[9:0] (D[9:0]/0x30055A).

- * **ADUPR[9:0]**: A/D Upper Limit Value Setup Bits in the A/D Upper Limit Value Register (D[9:0]/0x300558)
- * **ADLWR[9:0]**: A/D Lower Limit Value Setup Bits in the A/D Lower Limit Value Register (D[9:0]/0x30055A)

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 ADCMPE (D15/0x300544) to 1 to enable the range check function.

- * **ADCMPE**: A/D Upper/Lower-limit Comparison Enable Bit in the A/D Control/Status Register (D15/0x300544)

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 CNVINTEN (D4/0x300544). 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.

- * **CNVINTEN**: A/D Conversion-Complete Interrupt Enable Bit in the A/D Control/Status Register (D4/0x300544)

2. Enabling/disabling the out-of-range interrupt

The out-of-range interrupt can be enabled/disabled using CMPINTEN (D5/0x300544). 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.

- * **CMPINTEN**: A/D Out-of-Range Interrupt Enable Bit in the A/D Control/Status Register (D5/0x300544)

3. Setting the interrupt signal mode

The S1C33E08 A/D converter has two interrupt request outputs for the interrupt sources above and each interrupt can be handled individually. The A/D converter with advanced mode in the C33 STD Core model uses one signal line for interrupt requests to the ITC. In the initial setting, the out-of-range interrupt signal is ORed with the conversion-complete interrupt signal to send to the ITC. The S1C33E08 A/D converter also supports this interrupt signal mode. So, the cause of conversion-complete interrupt flag in the ITC is set when an A/D conversion has completed or when the conversion results are out of range. This signal mode can be canceled using INTMODE (D6/0x300544). To handle each interrupt individually, set INTMODE (D6/0x300544) to 1. In this setting, the out-of-range interrupt signal is not ORed with the conversion-complete interrupt signal.

- * **INTMODE**: Interrupt Signal Mode Select Bit in the A/D Control/Status Register (D6/0x300544)

4. Masking conversion-complete interrupt for the specified channels

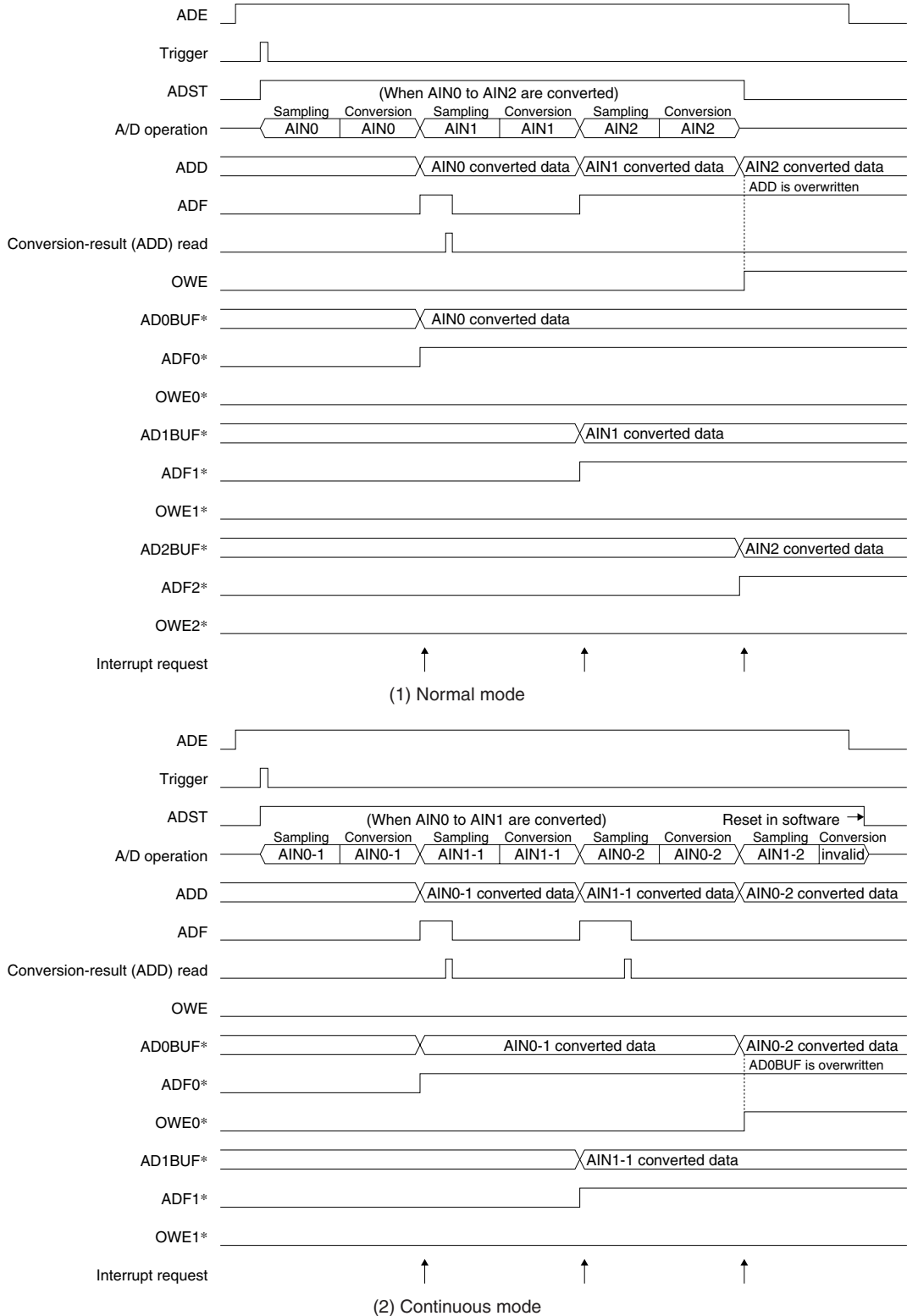
The A/D conversion-complete interrupt mask register is used to mask the conversion-complete interrupts of the specified channels. When INTMASK_x (Dx/0x30055C) for channel 'x' in the register is set to 0, channel 'x' does not generate conversion-complete interrupts. For instance, by masking the conversion-complete interrupt of the channel used for range checking, it is possible to generate out-of range interrupts only.

* **INTMASK_x**: Ch.x Conversion-Complete Interrupt Mask Bit in the A/D Conversion Complete Interrupt Mask Register (Dx/0x30055C)

At initial reset, INTMASK_x are all set to 1 to enable conversion-complete interrupts.

VII.1.5 Control and Operation of A/D Conversion

Figure VII.1.5.1 shows the operation of the A/D converter.



* Extended functions that can be used when ADCADV = 1

Figure VII.1.5.1 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 (D2/0x300544) 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 to be used, set ADE to 0.

- * **ADE:** A/D Enable Bit in the A/D Control/Status Register (D2/0x300544)

Starting A/D conversion

When a trigger is input while ADE (D2/0x300544) = 1, A/D conversion is started. If a software trigger has been selected, A/D conversion is started by writing 1 to ADST (D1/0x300544).

- * **ADST:** A/D Conversion Control/Status Bit in the A/D Control/Status Register (D1/0x300544)

Only the trigger selected using TS[1:0] (D[4:3]/0x300542) are valid; no other trigger is accepted.

- * **TS[1:0]:** A/D Conversion Trigger Select Bits in the A/D Trigger/Channel Select Register (D[4:3]/0x300542)

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] (D[10:8]/0x300542).

- * **CS[2:0]:** A/D Converter Start Channel Setup Bits in the A/D Trigger/Channel Select Register (D[10:8]/0x300542)

ADST (D1/0x300544) 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]/0x300542).

- * **CH[2:0]:** A/D Conversion Channel Status Bits in the A/D Trigger/Channel Select Register (D[2:0]/0x300542)

Reading out A/D conversion results

• Standard 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] (D[9:0]/0x300540), and sets the conversion-complete flag ADF (D3/0x300544) and cause-of-interrupt flag FADE (D1/0x300287). If multiple channels are specified using CS[2:0] (D[10:8]/0x300542) and CE[2:0] (D[13:11]/0x300542), A/D conversions in the subsequent channels are performed in succession.

- * **ADD[9:0]:** A/D Converted Data Bits in the A/D Conversion Result Register (D[9:0]/0x300540)
- * **ADF:** A/D Conversion Completion Flag in the A/D Control/Status Register (D3/0x300544)
- * **FADE:** A/D Conversion Completion Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D1/0x300287)
- * **CE[2:0]:** A/D Converter End Channel Setup Bits in the A/D Trigger/Channel Select Register (D[13:11]/0x300542)

The results of A/D conversion are stored in ADD[9:0] (D[9:0]/0x300540) 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 cause-of-interrupt flag (by writing 0) to prepare the A/D converter for the next operation.

Since the cause of interrupt 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, ADD[9:0] is overwritten with the new conversion results.

If ADD[9:0] is updated when the conversion-complete flag ADF (D3/0x300544) = 1 (before the converted data is read out), the overwrite-error flag OWE (D0/0x300544) 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 OWE 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.

- * **OWE:** Overwrite Error Flag in the A/D Control/Status Register (D0/0x300544)

- **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] (D[9:0]/0x300548 + 2•x) and sets the Ch.x conversion-complete flag ADFx (Dx/0x300546) and the cause-of-interrupt flag FADE (D1/0x300287). If multiple channels are specified using CS[2:0] (D[10:8]/0x300542) and CE[2:0] (D[13:11]/0x300542), A/D conversions in the subsequent channels are performed in succession.

- * **ADxBUF[9:0]**: A/D Ch.x Converted Data Bits in the A/D Ch.x Conversion Result Buffer Register (D[9:0]/0x300548 + 2•x)
- * **ADFx**: A/D Ch.x Conversion-Complete Flag in the A/D Channel Status Flag Register (Dx/0x300546)

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 cause-of-interrupt flag (by writing 0) to prepare the A/D converter for the next operation.

Since the cause of interrupt 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] (D[9:0]/0x300548 + 2•x) is updated when the conversion-complete flag ADFx = 1 (before the converted data is read out), the overwrite-error flag OWEx (Dx + 8/0x300546) is set to 1. ADFx (Dx/0x300546) 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, also read OWEx 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.

- * **OWEx**: A/D Ch.x Overwrite Error Flag in the A/D Channel Status Flag Register (Dx + 8/0x300546)

ADD[9:0] (D[9:0]/0x300540), ADF (D3/0x300544) and OWE (D0/0x300544) used in the standard 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 standard mode. OWE 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 (D15/0x300544) = 1) and an A/D conversion in the channel specified using ADCMP[2:0] (D[14:12]/0x300544) has completed, the conversion results are compared with the contents of ADUPR[9:0] (D[9:0]/0x300558) and ADLWR[9:0] (D[9:0]/0x30055A).

- * **ADCMPE**: A/D Upper/Lower-limit Comparison Enable Bit in the A/D Control/Status Register (D15/0x300544)
- * **ADCMP[2:0]**: A/D Upper/Lower-limit Comparison Channel Select Bits in the A/D Control/Status Register (D[14:12]/0x300544)
- * **ADUPR[9:0]**: A/D Upper Limit Value Setup Bits in the A/D Upper Limit Value Register (D[9:0]/0x300558)
- * **ADLWR[9:0]**: A/D Lower Limit Value Setup Bits in the A/D Lower Limit Value Register (D[9:0]/0x30055A)

If the conversion results exceed the upper-limit value, the upper-limit comparison status bit ADUPRST (D11/0x300544) is set to 1. If the results are less than the lower-limit value, the lower-limit comparison status bit ADLWRST (D10/0x300544) 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 cause-of-interrupt flag FADC (D0/0x300287). Also the same cause-of-interrupt flag FADE (D1/0x300287) as the conversion-complete interrupt is set to 1 when INTMODE (D6/0x300544) has been set to 0.

- * **ADUPRST**: A/D Upper-limit Comparison Status Bit in the A/D Control/Status Register (D11/0x300544)
- * **ADLWRST**: A/D Lower-limit Comparison Status Bit in the A/D Control/Status Register (D10/0x300544)
- * **FADC**: A/D Out-of-Range Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D0/0x300287)
- * **INTMODE**: Interrupt Signal Mode Select Bit in the A/D Control/Status Register (D6/0x300544)

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] (D[10:8]/0x300542) to the conversion end channel specified using CE[2:0] (D[13:11]/0x300542), and is completed after these conversions are executed in one operation. ADST (D1/0x300544) is reset to 0 upon completion of the conversion.

* **MS:** A/D Conversion Mode Select Bit in the A/D Trigger/Channel Select Register (D5/0x300542)

- **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 (D1/0x300544) 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 (D3/0x300544) 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 (D1/0x300544). The results of the conversion then under-way cannot be obtained.

VII.1.6 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 bits available for the A/D converter:

- * **FADE**: A/D Conversion Completion Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D1/0x300287)
- * **FADC**: A/D Out-of-Range Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D0/0x300287)
- * **EADE**: A/D Conversion Completion Interrupt Enable Bit in the Port Input 4–7, RTC, A/D Interrupt Enable Register (D1/0x300277)
- * **EADC**: A/D Out-of-Range Interrupt Enable Bit in the Port Input 4–7, RTC, A/D Interrupt Enable Register (D0/0x300277)
- * **PAD[2:0]**: A/D Interrupt Level Bits in the Serial I/F Ch.1, A/D Interrupt Priority Register (D[6:4]/0x30026A)

The A/D converter sets the cause-of-interrupt flag FADE (D1/0x300287) to 1 when A/D conversion in one channel is completed, and the conversion results are stored in ADD[9:0] and ADxBUF[9:0] (advanced mode).

- * **ADxBUF[9:0]**: A/D Ch.x Converted Data Bits in the A/D Ch.x Conversion Result Buffer Register (D[9:0]/0x300548 + 2•x)
- * **ADD[9:0]**: A/D Converted Data Bits in the A/D Conversion Result Register (D[9:0]/0x300540)

If the out-of-range interrupt is enabled in the advanced mode, the cause-of-interrupt flag FADC (D0/0x300287) is set to 1 when the conversion results in the specified channel are out of range. Also the same cause-of-interrupt flag FADE (D1/0x300287) as the conversion-complete interrupt is set to 1 when INTMODE (D6/0x300544) has been set to 0.

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 cause-of-interrupt 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 Section III.2, "Interrupt Controller (ITC)."

Intelligent DMA

The A/D converter can invoke the intelligent DMA (IDMA) through the use of its cause of interrupt when an A/D conversion has completed. 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 bit RADE (D2/0x300293) and the IDMA enable bit DEADE (D2/0x300297) must be set to 1. Transfer conditions on the IDMA side must also be set in advance.

- * **RADE**: A/D Conversion Completion IDMA Request Bit in the Serial I/F Ch.1, A/D, Port Input 4–7 IDMA Request Register (D2/0x300293)
- * **DEADE**: A/D Conversion Completion IDMA Enable Bit in the Serial I/F Ch.1, A/D, Port Input 4–7 IDMA Enable Register (D2/0x300297)

If a cause of interrupt 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 Section II.2, "Intelligent DMA (IDMA)."

High-speed DMA

The cause of A/D conversion complete interrupt can also invoke high-speed DMA (HSDMA).

The following shows the HSDMA channel number and trigger set-up bit:

Table VII.1.6.1 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 (0x300298)
1	HSD1S[3:0] (D[7:4]) / HSDMA Ch.0–1 trigger set-up register (0x300298)
2	HSD2S[3:0] (D[3:0]) / HSDMA Ch.2–3 trigger set-up register (0x300299)
3	HSD3S[3:0] (D[7:4]) / HSDMA Ch.2–3 trigger set-up register (0x300299)

For HSDMA to be invoked, the trigger set-up bits should be set to 0b1100 in advance. Transfer conditions, etc. must also be set on the HSDMA side. If the A/D cause of interrupt is selected as the HSDMA trigger, the HSDMA channel is invoked through generation of the cause of interrupt.

For details on HSDMA transfer, refer to Section II.1, “High-Speed DMA (HSDMA).”

Trap vector

The A/D converter's interrupt trap-vector default address is set to 0xC00100.

The base address of the trap table can be changed using the TTBR register.

VII.1.7 Details of Control Registers

Table VII.1.7.1 List of A/D Converter Registers

Address	Register name	Size	Function
0x00300520	A/D Clock Control Register (pAD_CLKCTL)	16	Controls A/D converter clock and selects division ratio.
0x00300540	A/D Conversion Result Register (pAD_ADD)	16	A/D conversion data
0x00300542	A/D Trigger/Channel Select Register (pAD_TRIG_CHNL)	16	Sets start/end channels and conversion mode.
0x00300544	A/D Control/Status Register (pAD_EN_SMPSTAT)	16	Controls A/D converter and indicates conversion status.
0x00300546	A/D Channel Status Flag Register (pAD_END)	16	Overwrite error and conversion complete status
0x00300548	A/D Ch.0 Conversion Result Buffer Register (pAD_CH0_BUF)	16	A/D Ch.0 conversion data
0x0030054A	A/D Ch.1 Conversion Result Buffer Register (pAD_CH1_BUF)	16	A/D Ch.1 conversion data
0x0030054C	A/D Ch.2 Conversion Result Buffer Register (pAD_CH2_BUF)	16	A/D Ch.2 conversion data
0x0030054E	A/D Ch.3 Conversion Result Buffer Register (pAD_CH3_BUF)	16	A/D Ch.3 conversion data
0x00300550	A/D Ch.4 Conversion Result Buffer Register (pAD_CH4_BUF)	16	A/D Ch.4 conversion data
0x00300558	A/D Upper Limit Value Register (pAD_UPPER)	16	Specifies A/D conversion upper limit value.
0x0030055A	A/D Lower Limit Value Register (pAD_LOWER)	16	Specifies A/D conversion lower limit value.
0x0030055C	A/D Conversion Complete Interrupt Mask Register (pAD_CH04_INTMASK)	16	Masks A/D conversion complete interrupt.
0x0030055E	A/D Converter Mode Select/Internal Status Register (pAD_ADVMODE)	16	Selects A/D operating mode and indicates internal status and internal counter value.

The following describes each A/D converter control register.

The A/D converter control registers are mapped in the 16-bit device area from 0x300520 to 0x30055E, and can be accessed in units of half-words and bytes.

Note: When setting the A/D converter control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x300520: A/D Clock Control Register (pAD_CLKCTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
A/D clock control register (pAD_CLKCTL)	00300520 (HW)	D15-4	–	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	
		D1	PSAD1		111	MCLK/256	0		
		D0	PSAD0		110	MCLK/128	0		
					101	MCLK/64			
					100	MCLK/32			
					011	MCLK/16			
		010	MCLK/8						
		001	MCLK/4						
		000	MCLK/2						

D[15:4] Reserved

D3 PSONAD: A/D Converter Clock Control Bit

Controls the A/D conversion clock supply to the A/D converter.

1 (R/W): On

0 (R/W): Off (default)

D[2:0] PSAD[2:0]: A/D Converter Clock Division Ratio Setup Bits

Selects a division ratio to generate the A/D converter clock.

Table VII.1.7.2 Selecting Division Ratio

PSAD2	PSAD1	PSAD0	Division ratio
1	1	1	MCLK/256
1	1	0	MCLK/128
1	0	1	MCLK/64
1	0	0	MCLK/32
0	1	1	MCLK/16
0	1	0	MCLK/8
0	0	1	MCLK/4
0	0	0	MCLK/2

(Default: 0b000 = MCLK/2)

0x300540: A/D Conversion Result Register (pAD_ADD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D conversion result register (pAD_ADD)	00300540 (HW)	D15-10	–	reserved		–	–	0 when being read.
		D9	ADD9	A/D converted data	0x0 to 0x3FF	0	R	
		D8	ADD8	ADD9 = MSB		0		
		D7	ADD7	ADD0 = LSB		0		
		D6	ADD6			0		
		D5	ADD5			0		
		D4	ADD4			0		
		D3	ADD3			0		
		D2	ADD2			0		
		D1	ADD1			0		
		D0	ADD0			0		

D[15:10] Reserved**D[9:0] ADD[9:0]: A/D Converted Data Bits**

Stores the results of A/D conversion. (Default: 0x000)

The LSB is stored in ADD0, and the MSB is stored in ADD9.

This is a read-only register, so writing to this register is ignored.

0x300542: A/D Trigger/Channel Select Register (pAD_TRIG_CHNL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
A/D trigger/ channel select register (pAD_TRIG_CHNL)	00300542 (HW)	D15-14	–	reserved	–	–	–	0 when being read.		
		D13	CE2	A/D converter end channel selection	0 to 4	0	R/W			
		D12	CE1							
		D11	CE0							
		D10	CS2	A/D converter start channel selection	0 to 4	0	R/W			
		D9	CS1							
		D8	CS0							
		D7-6	–	reserved	–	–	–	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							11
10	reserved									
01	16-bit timer									
00	Software									
D2	CH2	A/D conversion channel status	0 to 4	0	R					
D1	CH1									
D0	CH0									
							0			

D[15:14] Reserved

D[13:11] CE[2:0]: A/D Converter End Channel Setup Bits

Sets the conversion end channel by selecting a channel number from 0 to 4. (Default: 0b000 = AIN0)
Analog inputs can be A/D-converted successively from the channel set using CS[2:0] (D[10:8]) to the channel set using these bits in one operation. If only one channel is to be A/D converted, set the same channel number in both CS[2:0] and CE[2:0].

D[10:8] CS[2:0]: A/D Converter Start Channel Setup Bits

Sets the conversion start channel by selecting a channel number from 0 to 4. (Default: 0b000 = AIN0)
Analog inputs can be A/D-converted successively from the channel set using these bits to the channel set using CE[2:0] (D[13:11]) in one operation. If only one channel is to be A/D converted, set the same channel number in both CS[2:0] and CE[2:0].

D[7:6] Reserved

D5 MS: A/D Conversion Mode Select Bit

Selects an A/D conversion mode.
1 (R/W): Continuous mode
0 (R/W): Normal mode (default)

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[2:0] (D[10:8]) and CE[2:0] (D[13:11]) 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[2:0] and CE[2:0] are converted in one operation.

D[4:3] TS[1:0]: A/D Conversion Trigger Select Bits

Selects a trigger to start A/D conversion.

Table VII.1.7.3 Trigger Selection

TS1	TS0	Trigger
1	1	External trigger (#ADTRG)
1	0	Reserved
0	1	16-bit timer 0
0	0	Software

(Default: 0b00 = Software trigger)

When an external trigger is used, the #ADTRG pin must be set in advance using the port function select register. A/D conversion is started when a low level of the #ADTRG signal is detected.

When the 16-bit timer is used, since its comparison match B signal serves as a trigger, set the cycle and other parameters for the timer.

D[2:0] CH[2:0]: A/D Conversion Channel Status Bits

Indicates the channel number (0 to 4) currently being A/D-converted. (Default: 0b000 = AIN0)

When A/D conversion is performed in multiple channels, read this bit to identify the channel in which conversion is underway.

0x300544: A/D Control/Status Register (pAD_EN_SMP_L_STAT)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
A/D control/ status register (pAD_EN_SMP_L _STAT)	00300544 (HW)	D15	ADCMPE	Upper/lower-limit comparison enable	1	Enabled	0	Disabled	0	R/W	Can be used when ADCADV = "1".
		D14	ADCMP2	Upper/lower-limit comparison channel selection	0 to 4			0	R/W		
		D13	ADCMP1		0						
		D12	ADCMPO		0						
		D11	ADUPRST	Upper-limit comparison status	1	Out of range	0	Within range	0	R	
		D10	ADLWRST	Lower-limit comparison status	1	Out of range	0	Within range	0	R	
		D9	ST1	Input signal sampling time setup	ST[1:0]	Sampling time	1	R/W	Use with 9 clocks.		
		D8	ST0		11	9 clocks					
					10	7 clocks					
					01	5 clocks					
				00	3 clocks						
		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	Out-of-range 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.		

D15 ADCMPE: A/D Upper/Lower-limit Comparison Enable Bit (for advanced mode)

Enables/disables comparison between converted data and upper-/lower-limit values.

1 (R/W): Enabled

0 (R/W): Disabled (default)

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] (D[14:12]). Set ADCMPE to 1 when using the comparison function or set to 0 when not used.

D[14:12] ADCMP[2:0]: A/D Upper/Lower-limit Comparison Channel Select Bits (for advanced mode)

Set the channel number (0–4) to compare its converted data with the upper-/ lower-limit values. (Default: 0b000 = AIN0)

D11 ADUPRST: A/D Upper-limit Comparison Status Bit (for advanced mode)

Indicates the results of comparison between the A/D converted data and the upper-limit value.

1 (R): Exceeded the upper limit

0 (R): Within the range (default)

When the upper-/lower-limit comparison function is enabled (ADCMPE (D15) = 1), the converted data is compared with the upper-/lower-limit values after A/D conversion of the channel specified using ADCMP[2:0] (D[14:12]) has completed. If the converted data exceeds the upper-limit value set in ADUPR[9:0] (D[9:0]/0x300558), 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.

D10 ADLWRST: A/D Lower-limit Comparison Status Bit (for advanced mode)

Indicates the results of comparison between the A/D converted data and the lower-limit value.

1 (R): Under the lower limit

0 (R): Within the range (default)

When the upper-/lower-limit comparison function is enabled (ADCMPE (D15) = 1), the converted data is compared with the upper-/lower-limit values after A/D conversion of the channel specified using ADCMP[2:0] (D[14:12]) has completed. If the converted data is less than the lower-limit value set in ADLWR[9:0] (D[9:0]/0x30055A), ADLWRST is set to 1. If the converted data is equal to or more than the lower-limit value, ADLWRST is set to 0. An interrupt occurs when ADLWRST is set to 1 if the out-of-range interrupt is enabled.

D[9:8] ST[1:0]: Input Signal Sampling Time Setup Bits

Sets the analog input sampling time.

Table VII.1.7.4 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

(Default: 0b11 = 9-clock period)

The A/D converter conversion clock is used for counting.

To maintain the conversion accuracy, use ST as set by default (9-clock period).

D7 Reserved**D6 INTMODE: Interrupt Signal Mode Select Bit (for advanced mode)**

Configures the conversion-complete interrupt signal delivered to the ITC.

1 (R/W): Conversion-complete signal only

0 (R/W): OR between conversion-complete and out-of-range signals (default)

INTMODE selects whether the conversion-complete 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 when handling the out-of-range interrupt as another interrupt. When using the out-of-range interrupt, set CMPINTEN (D5) to 1.

D5 CMPINTEN: A/D Out-of-Range Interrupt Enable Bit (for advanced mode)

Enables/disables the out-of-range interrupt.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When CMPINTEN is set to 1, upper and lower-limit comparison results become a cause of interrupt. When it is set to 0, an out-of-range interrupt is not generated.

D4 CNVINTEN: A/D Conversion-Complete Interrupt Enable Bit (for advanced mode)

Enables/disables the conversion-complete interrupt.

1 (R/W): Enabled (default)

0 (R/W): Disabled

When CNVINTEN is set to 1, completion of an A/D conversion becomes a cause of interrupt. When it is set to 0, a conversion-complete interrupt is not generated.

D3 ADF: A/D Conversion Completion Flag

Indicates that A/D conversion has been completed.

1 (R): Conversion completed

0 (R): Being converted or standing by (default)

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.

D2 ADE: A/D Enable Bit

Enables the A/D converter (readied for conversion).

1 (R/W): Enabled

0 (R/W): Disabled (default)

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.

D1 ADST: A/D Conversion Control/Status Bit

Controls A/D conversion.

1 (R/W): Software trigger

0 (R/W): A/D conversion is stopped (default)

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 (D2) = 0 (A/D conversion disabled), ADST is fixed to 0, with no trigger accepted.

D0 OWE: Overwrite Error Flag

Indicates that the converted data has been overwritten.

1 (R): Overwritten

0 (R): Normal (default)

1 (W): Has no effect

0 (W): Flag is reset

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 (D3) that has been set through conversion of the preceding channel, OWE is set to 1. When ADF (D3) 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.

0x300546: A/D Channel Status Flag Register (pAD_END)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
A/D channel status flag register (pAD_END)	00300546 (HW)	D15-13	–	reserved			–	–	0 when being read.		
		D12	OWE4	Ch.4 overwrite error flag	1	Error	0	Normal	0	R/W	Can be used when ADCADV = "1". Reset by writing 0.
		D11	OWE3	Ch.3 overwrite error flag					0	R/W	
		D10	OWE2	Ch.2 overwrite error flag					0	R/W	
		D9	OWE1	Ch.1 overwrite error flag					0	R/W	
		D8	OWE0	Ch.0 overwrite error flag					0	R/W	
		D7-5	–	reserved					–	–	0 when being read.
		D4	ADF4	Ch.4 conversion-complete flag	1	Completed	0	Run/Standby	0	R	Can be used when ADCADV = "1". 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	

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 4.

D[15:13] Reserved**D[12:8] OWE[4:0]: A/D Ch.x Overwrite Error Flags (for advanced mode)**

These bits indicate that the conversion result buffer for each channel has been overwritten.

- 1 (R): Overwritten
- 0 (R): Normal (default)
- 1 (W): Has no effect
- 0 (W): Flag is reset

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 ADF_x conversion-complete flag that has been set through the previous conversion, OWE_x is set to 1. When ADF_x is reset, because this means that the converted data has been read out, OWE_x is not set. Once OWE_x is set to 1, it remains set until it is reset by writing 0 in the software.

D[7:5] Reserved**D[4:0] ADF[4:0]: A/D Ch.x Conversion-Complete Flags (for advanced mode)**

These bits indicate that A/D conversion in each channel has been completed.

- 1 (R): Conversion completed
- 0 (R): Being converted or standing by (default)

This flag is set to 1 when A/D conversion of the corresponding channel is completed, and the converted data is stored in the conversion result buffer and is reset to 0 when the conversion result buffer is read out. When A/D conversion is performed in continuous mode, if the next A/D conversion of the same channel is completed while ADF_x = 1 (before the conversion result buffer is read out), the buffer is overwritten with the new conversion results, causing an overrun error to occur. Therefore, ADF_x must be reset by reading out the converted data before the next A/D conversion is completed.

0x300548–0x300550: A/D Ch.x Conversion Result Buffer Registers (pAD_CHx_BUF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Ch.x conversion result buffer register (pAD_CHx_BUF)	00300548 00300550 (HW)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	ADxBUF9	A/D Ch.x converted data	0x0 to 0x3FF	0	R	Can be used when ADCADV = "1".
		D8	ADxBUF8	ADxBUF9 = MSB		0		
		D7	ADxBUF7	ADxBUF0 = LSB		0		
		D6	ADxBUF6			0		
		D5	ADxBUF5			0		
		D4	ADxBUF4			0		
		D3	ADxBUF3			0		
		D2	ADxBUF2			0		
		D1	ADxBUF1			0		
D0	ADxBUF0			0				

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 4.

- 0x300548 A/D Ch.0 Conversion Result Buffer Register (pAD_CH0_BUF)
- 0x30054A A/D Ch.1 Conversion Result Buffer Register (pAD_CH1_BUF)
- 0x30054C A/D Ch.2 Conversion Result Buffer Register (pAD_CH2_BUF)
- 0x30054E A/D Ch.3 Conversion Result Buffer Register (pAD_CH3_BUF)
- 0x300550 A/D Ch.4 Conversion Result Buffer Register (pAD_CH4_BUF)

D[15:10] Reserved

D[9:0] ADxBUF[9:0]: A/D Ch.x Converted Data Bits (for advanced mode)

The conversion results in each channel are stored. (Default: 0x000)
 This is a read-only register, so writing to this register is ignored.

0x300558: A/D Upper Limit Value Register (pAD_UPPER)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D upper limit value register (pAD_UPPER)	00300558 (HW)	D15-10	–	reserved		–	–	0 when being read.
		D9	ADUPR9	A/D conversion upper limit value	0x0 to 0x3FF	0	R/W	Can be used when ADCADV = "1".
		D8	ADUPR8	ADUPR9 = MSB		0		
		D7	ADUPR7	ADUPR0 = LSB		0		
		D6	ADUPR6			0		
		D5	ADUPR5			0		
		D4	ADUPR4			0		
		D3	ADUPR3			0		
		D2	ADUPR2			0		
		D1	ADUPR1			0		
		D0	ADUPR0			0		

D[15:10] Reserved**D[9:0] ADUPR[9:0]: A/D Upper Limit Value Setup Bits (for advanced mode)**

Set the upper-limit value to be compared with the A/D conversion results. (Default: 0x000)

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] (D[14:12]/0x300544). If the converted data exceeds the set value, an interrupt can be generated.

0x30055A: A/D Lower Limit Value Register (pAD_LOWER)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D lower limit value register (pAD_LOWER)	0030055A (HW)	D15-10	–	reserved		–	–	0 when being read.
		D9	ADLWR9	A/D conversion lower limit value	0x0 to 0x3FF	0	R/W	Can be used when ADCADV = "1".
		D8	ADLWR8	ADLWR9 = MSB		0		
		D7	ADLWR7	ADLWR0 = LSB		0		
		D6	ADLWR6			0		
		D5	ADLWR5			0		
		D4	ADLWR4			0		
		D3	ADLWR3			0		
		D2	ADLWR2			0		
		D1	ADLWR1			0		
		D0	ADLWR0			0		

D[15:10] Reserved**D[9:0] ADLWR[9:0]: A/D Lower Limit Value Setup Bits (for advanced mode)**

Set the lower-limit value to be compared with the A/D conversion results. (Default: 0x000)

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] (D[14:12]/0x300544). If the converted data is less than the set value, an interrupt can be generated.

0x30055C: A/D Conversion Complete Interrupt Mask Register (pAD_CH04_INTMASK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
A/D conversion complete interrupt mask register (pAD_CH04_INTMASK)	0030055C (HW)	D15–5	–	reserved		–	–	–	0 when being read.		
		D4	INTMASK4	Ch.4 conversion-complete int. mask	1	Interrupt enabled	0	Interrupt mask	1	R/W	Can be used when ADCADV = "1".
		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	

Note: The letter 'x' in bit names, etc., denotes a channel number from 0 to 4.

D[15:5] Reserved

D[4:0] INTMASK[4:0]: Ch.x Conversion-Complete Interrupt Mask Bits (for advanced mode)

These bits mask the A/D conversion-complete interrupt for each channel individually.

1 (R/W): Interrupt is enabled (default)

0 (R/W): Interrupt is masked

When INTMASK_x is set to 0, the conversion-completed interrupt request of the Ch.x is masked and the cause-of-interrupt flag FADE (D1/0x300287) will not be set to 1 even if A/D conversion is completed. When INTMASK_x is 1, the A/D converter can generate an interrupt upon completion of A/D conversion in Ch.x.

* **FADE:** A/D Conversion Completion Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D1/0x300287)

0x30055E: A/D Converter Mode Select/Internal Status Register (pAD_ADVMODE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
A/D converter mode select/internal status register (pAD_ADVMODE)	0030055E (HW)	D15-9	–	reserved	–	–	–	Do not write 1.	
		D8	ADCADV	Standard/advanced mode selection	1 Advanced 0 Standard	0	R/W		
		D7-6	–	reserved	–	–	–	–	0 when being read.
		D5	ISTATE1	Internal status	ISTATE[1:0]	Status	0	R	
		D4	ISTATE0		11	Converting	0		
					10	reserved			
		01	Sampling						
		00	Idle						
D3	ICOUNTER3	Internal counter value	0 to 15		0	R			
D2	ICOUNTER2		0						
D1	ICOUNTER1		0						
D0	ICOUNTER0		0						

D[15:9] Reserved

D8 ADCADV: Standard/Advanced Mode Select Bit

Selects the A/D converter operating mode.

1 (R/W): Advanced mode

0 (R/W): Standard mode (default)

When ADCADV is set to 1, the A/D converter is set in the advanced mode, and the registers/bits for the extended function can be used.

When ADCADV is set to 0, only the standard C33 A/D converter functions implemented in C33 STD models can be used. In this mode, the extended registers/bits for advanced mode become read only and writing operation is disabled.

D[7:6] Reserved

D[5:4] ISTATE[1:0]: Internal Status Bits

Indicates the A/D converter internal status.

Table VII.1.7.5 Internal Status

ISTATE1	ISTATE0	Status
1	1	Converting
1	0	Reserved
0	1	Sampling
0	0	Idle

(Default: 0b00 = Idle)

D[3:0] ICOUNTER[3:0]: Internal Counter Value Setup Bits

Indicates the internal counter value. (Default: 0b0000)

VII.1.8 Precautions

- Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to disable ADE (D2/0x300544). A change in settings while the A/D converter is enabled could cause it to operate erratically.
 - * **ADE**: A/D Enable Bit in the A/D Control/Status Register (D2/0x300544)
- In consideration of the conversion accuracy, we recommend that the A/D conversion clock be min. 16 kHz to max. 2 MHz.
- 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.
- After an initial reset, FADE (D1/0x300287) and FADC (D0/0x300287) become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset these flags in a program.
 - * **FADE**: A/D Conversion Completion Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D1/0x300287)
 - * **FADC**: A/D Out-of-Range Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D0/0x300287)
- To prevent the regeneration of interrupts due to the same cause of interrupt following the occurrence an interrupt, always be sure to reset the cause-of-interrupt flag before setting the PSR again or executing the reti instruction.
- When the A/D converter is set to enabled state, a current flows between AVDD 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 (D2/0x300544)).
- When the 16-bit timer 0 compare match B signal is used as a trigger factor, the division ratio of the prescaler in the 16-bit timer module must not be set to MCLK/1.
- 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.
- Software controllable pull-up resistors are provided for the input ports. Disable the pull-up resistors of the ports used for analog inputs.
- 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.

S1C33E08 Technical Manual

VIII PERIPHERAL MODULES 6 (LCD)

VIII.1 LCD Controller (LCDC)

VIII.1.1 Overview

The S1C33E08 has a built-in LCD controller (LCDC) that supports 4/8-bit monochrome and color LCD panels, and 12-bit Generic HR-TFT panels. Also the S1C33E08 contains a 12-K byte VRAM (IVRAM) allowing a 320 × 240-dot monochrome screen (1-bpp mode) to be displayed. Furthermore, the SDRAM controller allows the LCDC to access the external SDRAM as a VRAM, thus a 320 × 240-dot screen with 8-bpp color depth (typ.) can be displayed.

The LCDC provides support for Picture-in-Picture Plus (a variable size overlay window). The LCDC can use both the IVRAM and external VRAM, this makes it possible to manage the main and sub (PIP) window display data in different memories.

The features of the LCDC are described below.

Internal bus 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 32-bit accesses are possible.
- The 12K-byte internal VRAM (IVRAM) is mapped at addresses 0x80000 to 0x82FFF.
- The external VRAM map is configurable (in a 2MB or 64MB SDRAM area).
- The LCDC interrupt signal is assigned to interrupt vector No. 73 (trap table base address + 0x124) in the ITC.

Display support

- 4- or 8-bit monochrome LCD interface
- 4- or 8-bit color LCD interface
- Single-panel, single-drive passive displays
- 12-bit Generic HR-TFT interface
320 × 240-dot Sharp HR-TFT panel, SII liquid TFT panel, or some other TFT panels
- Typical resolutions
320 × 240 (8-bpp mode, external VRAM is required) bpp = bits per pixel
320 × 240 (1-bpp mode)
* Note that the panel width must be a multiple of 16 ÷ bits per pixel.

Display modes

- Due to frame rate modulation, grayscale display is possible in up to 16 shades of gray when a monochrome passive LCD panel is used.
Two-shade display in 1-bpp mode
Four-shade display in 2-bpp mode
16-shade display in 4-bpp mode
- A maximum of 64K colors can be simultaneously displayed on a color passive LCD panel.
256-color display in 8-bpp mode
4K-color display in 12-bpp mode
64K-color display in 16-bpp mode
- A maximum of 4096 colors can be simultaneously displayed on a TFT panel.
Two-color display in 1-bpp mode
Four-color display in 2-bpp mode
16-color display in 4-bpp mode
256-color display in 8-bpp mode
4K-color display in 12-bpp mode

Display features

- Picture-in-Picture Plus
Displays a variable size window overlaid over background image.

Clock

- The PCLK (pixel clock) is generated in the CMU by dividing the OSC3 clock by 1 to 16.
- Different clock paths are provided for the AHB bus interface (for accessing the VRAM), SAPB interface (for accessing the control registers), and PCLK, and each clock supply can be controlled individually in the CMU. This makes it possible to reduce current consumption by disabling unnecessary clocks.

Power save

- Software power-save mode
- DOZE mode is supported for RAM built-in or self-refresh-type LCD panels
- Blank display

VIII.1.2 Block Diagram

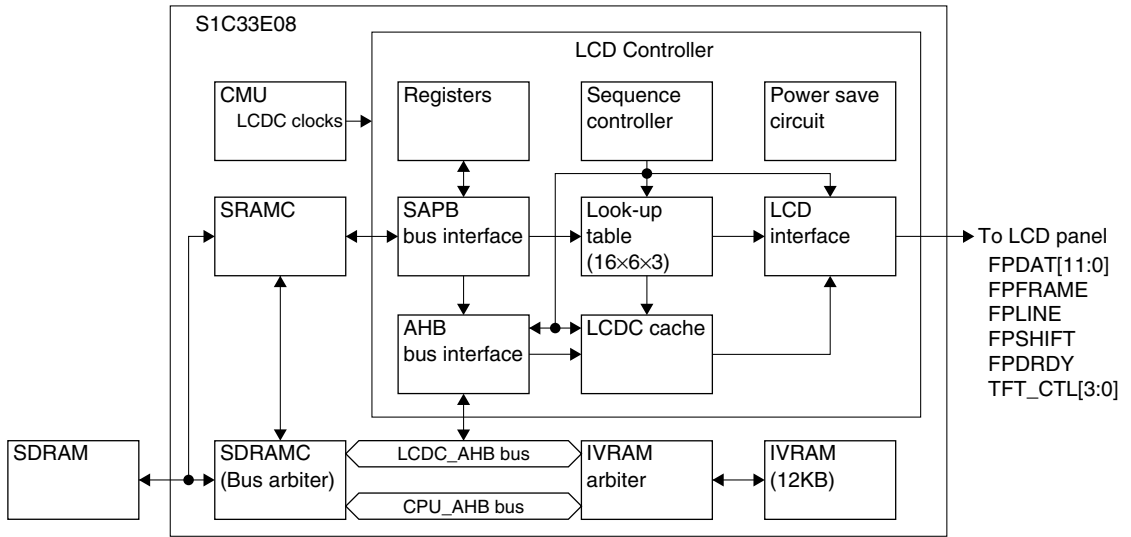


Figure VIII.1.2.1 Block Diagram of the LCD Controller

SAPB bus interface

The C33 PE Core accesses the LCDC registers and look-up table through this interface.

AHB bus interface

The LCDC access the VRAM through this interface.

LCDC cache

This consists of two 32-byte FIFOs used as a display data cache for sending display data to the LCD panel.

Sequence controller

The sequence controller controls data flow from the AHB bus interface 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 16×6 -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.

VIII.1.3 Output Pins of the LCD Controller

Table VIII.1.3.1 lists the output pins of the LCD controller. Table VIII.1.3.2 shows the pin configurations classified by type of LCD panel.

Table VIII.1.3.1 Output Pins of the LCD Controller

Pin name	I/O	Function
FPDAT[11:0]	O	LCD display data outputs
FPFRAME	O	LCD frame clock output
FPLINE	O	LCD line clock output
FPSHIFT	O	LCD shift clock output
FPDRDY	O	LCD DRDY/MOD signal output
TFT_CTL[3:0]	O	TFT I/F control signal outputs

Note: The LCDC output pins are shared with general-purpose I/O ports or other peripheral circuit inputs/outputs, so that functionality in the initial state is set to other than the LCDC output. Before the LCDC output signals assigned to these pins can be used, the function of these pins must be switched for the LCDC output by setting the corresponding Port Function Select Registers. For details of pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Table VIII.1.3.2 Pin Configurations by Type of LCD Panel

Pin name	Monochrome passive panel (Single)		Color passive panel (Single)			Generic HR-TFT panel
	4 bits	8 bits	4 bits	8-bit format 1	8-bit format 2	12 bits
FPFRAME	FPFRAME					SPS
FPLINE	FPLINE					LP
FPSHIFT	FPSHIFT					DCLK
FPDRDY	MOD	MOD	MOD	FPSHIFT2	MOD	Driven 0 *
FPDAT0	Driven 0 *	D0	Driven 0 *	D0	D0	B0
FPDAT1	Driven 0 *	D1	Driven 0 *	D1	D1	B1
FPDAT2	Driven 0 *	D2	Driven 0 *	D2	D2	B2
FPDAT3	Driven 0 *	D3	Driven 0 *	D3	D3	B3
FPDAT4	D0	D4	D0	D4	D4	G0
FPDAT5	D1	D5	D1	D5	D5	G1
FPDAT6	D2	D6	D2	D6	D6	G2
FPDAT7	D3	D7	D3	D7	D7	G3
FPDAT8	Driven 0 *					R0
FPDAT9	Driven 0 *					R1
FPDAT10	Driven 0 *					R2
FPDAT11	Driven 0 *					R3
TFT_CTL0	Driven 0 *					PS
TFT_CTL1	Driven 0 *					CLS
TFT_CTL2	Driven 0 *					REV
TFT_CTL3	Driven 0 *					SPL

* These pins can be used for other peripheral functions.

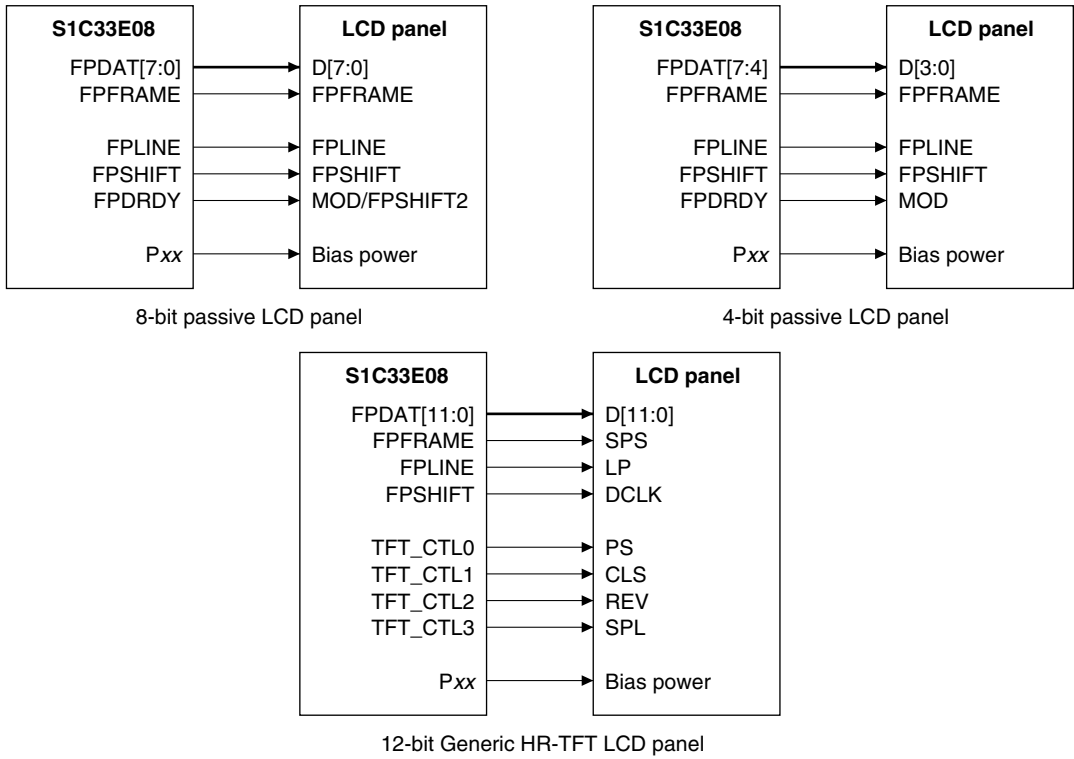


Figure VIII.1.3.1 Typical LCD-Panel Connections

VIII.1.4 System Settings

VIII.1.4.1 Configuration of Display Data Memory (VRAM)

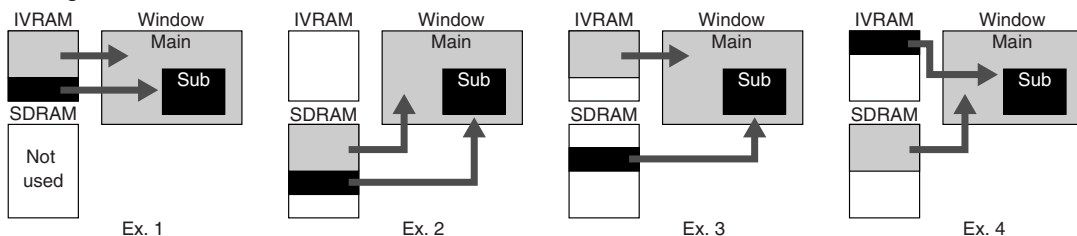
The S1C33E08 has a built-in 12K-byte display data memory (IVRAM). This memory allows selection whether it is used as a VRAM by locating at 0x80000 to 0x82FFF in area 3 or a general-purpose RAM by locating in area 0. Setting IRAM (D0/0x3001A64) to 1 configures the RAM as a general-purpose RAM in area 0; setting to 0 configures it as a VRAM in area 3. At initial reset, this memory is located in area 3 as a VRAM allowing LCDC to access directly.

* **IRAM:** IRAM Assignment Bit in the IRAM Select Register (D0/0x3001A64)

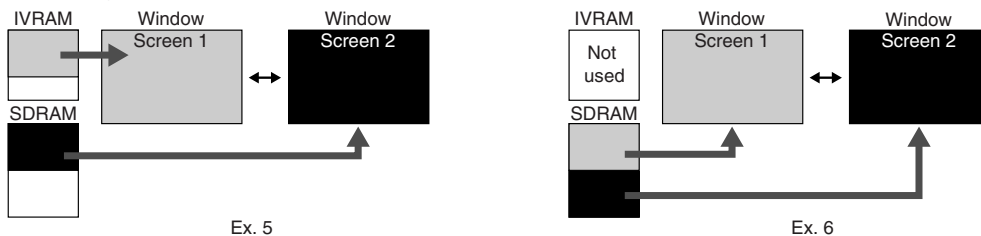
The LCDC can use an external SDRAM as a VRAM in addition to IVRAM (the SDRAM can store general-purpose data as well as display data). There is no special configuration procedure for use of the external SDRAM. Furthermore, both the external SDRAM and IVRAM can be used as VRAM simultaneously. The LCDC handles two screen data for the main window and the sub-window overlaid over the main window to support “Picture-in-Picture Plus.” Also the LCDC can switch the display by selecting a screen from two or more display data prepared in the VRAM. Since the display start memory address is specified using a register, display data can be stored in any location (but it must be a word boundary address) in the memory.

Figure VIII.1.4.1.1 shows memory usage examples.

When using Picture-in-Picture Plus



When switching screens



When using virtual screen area

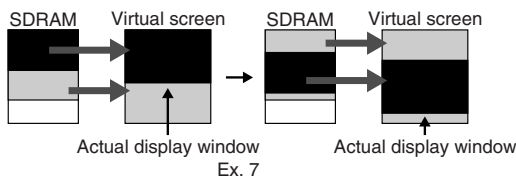


Figure VIII.1.4.1.1 Memory Usage Examples

The memory size required for a screen depends on the screen size and bpp mode. It can be expressed by the following equation:

$$\text{Screen data size} = H_PIXEL \times V_PIXEL \times \text{bpp} / 32 \text{ [words]}$$

(The fractional portion of the number must be rounded up.)

H_PIXEL: Number of horizontal pixels

V_PIXEL: Number of vertical pixels

bpp: Number of bits per pixel (1, 2, 4, 8, 12, 16)

For example, a 320 × 240-pixel screen displayed in 256 colors (8-bpp mode) needs a 19,200 words (75K bytes) of memory area.

VIII.1.4.2 Setting the LCDC Clock

The CMU provides the clock paths with a control bit shown below for the LCDC. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) LCDC AHB bus interface clock (LCDC_AHBIF_CLK)

The LCDC uses this clock (MCLK) to access IVRAM (internal VRAM) or an SDRAM (external VRAM). This clock is required for displaying a screen on the LCD panel. LCDCAHBIF_CKE (D2/0x301B00) is used for clock supply control (default: off).

* **LCDCAHBIF_CKE**: LCDC AHB Bus Interface Clock Control Bit in the Gated Clock Control Register 0 (D2/0x301B00)

(2) Control register clock (LCDC_SAPB_CLK)

This clock (MCLK) is used to control the LCDC registers located in area 6. This clock is required for accessing the LCDC registers and it can be stopped otherwise. LCDCSAPB_CKE (D1/0x301B00) is used for clock supply control (default: off).

* **LCDCSAPB_CKE**: LCDC SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 0 (D1/0x301B00)

(3) IVRAM arbiter clock (IVRAM_ARB_CLK)

This clock (MCLK) is used when the LCDC or CPU accesses IVRAM. When IVRAM is configured as A0RAM accessed by the CPU only, the clock supply can be stopped. IVRAMARB_CKE (D19/0x301B04) is used for clock supply control (default: on).

* **IVRAMARB_CKE**: IVRAM Arbiter Clock Control Bit in the Gated Clock Control Register 1 (D19/0x301B04)

(4) LCD interface clock (LCDC_CLK)

This is the LCD interface clock (LCDC_CLK) generated by dividing the OSC3 clock. The frequency divider generates 16 kinds of clocks from OSC3•1/1 to OSC3•1/16. Select a divided clock according to the frame rate using LCDCDIV[3:0] (D[19:16]/0x301B08).

$$\text{Frame rate} = \frac{f_{\text{LCDC}}}{\text{HT} \times \text{VT}} \text{ [Hz]}$$

f_{LCDC}: LCDC_CLK frequency

HT: Horizontal total period (including non-display period) [pixels]

VT: Vertical total period (including non-display period) [pixels]

* **LCDCDIV[3:0]**: LCDC Clock Divider Select Bits in the System Clock Control Register (D[19:16]/0x301B08)

Table VIII.1.4.2.1 Selecting the LCDC Clock

LCDCDIV3	LCDCDIV2	LCDCDIV1	LCDCDIV0	LCDC_CLK
1	1	1	1	OSC3•1/16
1	1	1	0	OSC3•1/15
1	1	0	1	OSC3•1/14
1	1	0	0	OSC3•1/13
1	0	1	1	OSC3•1/12
1	0	1	0	OSC3•1/11
1	0	0	1	OSC3•1/10
1	0	0	0	OSC3•1/9
0	1	1	1	OSC3•1/8
0	1	1	0	OSC3•1/7
0	1	0	1	OSC3•1/6
0	1	0	0	OSC3•1/5
0	0	1	1	OSC3•1/4
0	0	1	0	OSC3•1/3
0	0	0	1	OSC3•1/2
0	0	0	0	OSC3•1/1

(Default: 0b0111 = OSC3•1/8)

LCDC_CKE (D0/0x301B00) is used for clock supply control (default: off).

* **LCDC_CKE**: LCDC Main Clock Control Bit in the Gated Clock Control Register 0 (D0/0x301B00)

(5) LCDC_AHB bus clock (LCDC_AHBBUS_CLK)

The LCDC_AHB bus clock (MCLK) is always supplied in normal operation. However, it can be automatically turned off in HALT mode (see Section III.1.9.2) by setting LCDCAHB_HCKE (D28/0x301B04) to 0 (default: on).

* **LCDC_AHB_HCKE**: LCDC_AHB Bus Clock Control (HALT) Bit in the Gated Clock Control Register 1 (D28/0x301B04)

- Notes:**
- The LCDC clock supply cannot be stopped while the LCD displays a screen. Before the LCDC clock supply can be stopped, the LCDC must enter power save mode.
 - The Gated Clock Control Registers 0 and 1 (0x301B00, 0x301B04) are write-protected. Write protection of the CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless rewriting said registers.

VIII.1.5 Setting the LCD Panel

VIII.1.5.1 Types of Panels

The LCD controller supports the following types of LCD panels.

- 4- or 8-bit single monochrome passive LCD panels
- 4- or 8-bit single color passive LCD panels
- 12-bit Generic HR-TFT LCD panels

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 STN and HR-TFT

Use TFTSEL (D31/0x301A60) to select the type of LCD panel, either STN or HR-TFT.

TFTSEL = 1: Generic HR-TFT panel selected

TFTSEL = 0: STN panel selected (default)

* **TFTSEL**: HR-TFT Panel Select Bit in the LCDC Display Mode Register (D31/0x301A60)

Selecting between color and monochrome

Use COLOR (D30/0x301A60) to select the type of LCD panel, either color or monochrome.

COLOR = 1: Color panel selected

COLOR = 0: Monochrome panel selected (default)

* **COLOR**: Color/Mono Panel Select Bit in the LCDC Display Mode Register (D30/0x301A60)

Selecting the data width

Use DWD[1:0] (D[27:26]/0x301A60) to select the data width and format.

* **DWD[1:0]**: LCD Panel Data Width Select Bits in the LCDC Display Mode Register (D[27:26]/0x301A60)

Table VIII.1.5.1.1 Selection of the LCD Panel

TFTSEL	COLOR	DWD1	DWD0	LCD panel
1	–	–	–	12-bit Generic HR-TFT LCD
0	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

VIII.1.5.2 STN Panel Timing Parameters

The STN panel timing parameters shown in Figures below can be set using the LCDC control registers.

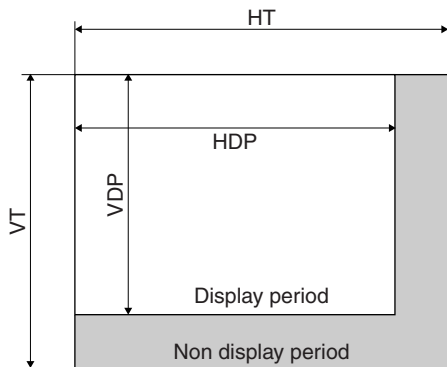


Figure VIII.1.5.2.1 STN Panel Timing Parameters

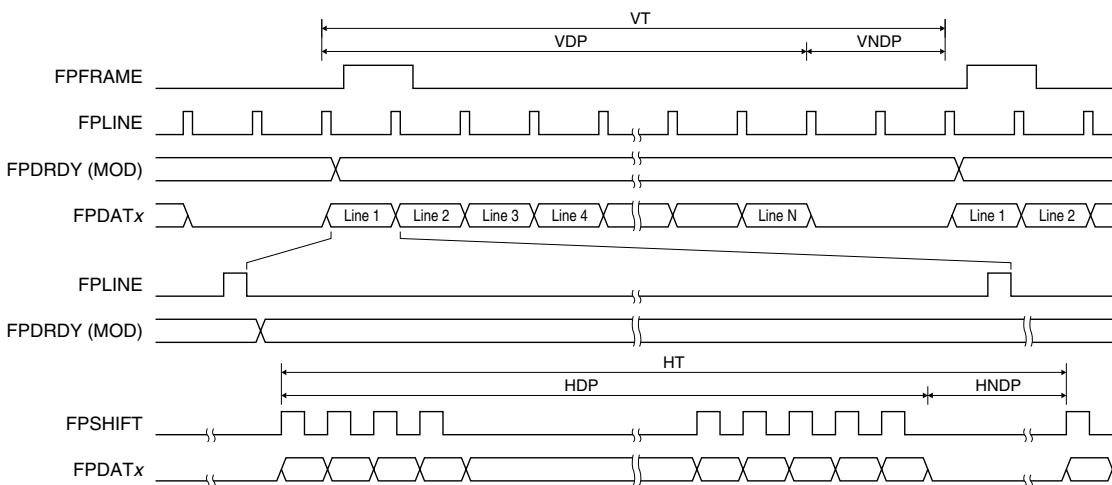


Figure VIII.1.5.2.2 STN Panel Timing Chart (Example)

HT: Horizontal total period

Use HTCNT[6:0] (D[22:16]/0x301A10) to set the horizontal total period.

$$HT = (HTCNT[6:0] + 1) \times 8 [Ts] \quad (Ts: \text{Pixel clock period})$$

* **HTCNT[6:0]**: Horizontal Total Period (HT) Setup Bits in the Horizontal Display Register (D[22:16]/0x301A10)

HTCNT[6:0] (D[22:16]/0x301A10) must be programmed such that the following condition is met:

$$HTCNT[6:0] \geq HDPCNT[6:0] + 3$$

HDP: Horizontal display period

Use HDPCNT[6:0] (D[6:0]/0x301A10) to set the horizontal display period (= horizontal panel resolution).

$$HDP = (HDPCNT[6:0] + 1) \times 8 [Ts]$$

* **HDPCNT[6:0]**: Horizontal Display Period (HDP) Setup Bits in the Horizontal Display Register (D[6:0]/0x301A10)

HDPCNT[6:0] (D[6:0]/0x301A10) must be programmed such that the following condition is met:

$$HDP \geq 16 \quad (HDPCNT[6:0] \geq 1)$$

VT: Vertical total period

Use VTCNT[9:0] (D[25:16]/0x301A14) to set the vertical total period.

$$VT = VTCNT[9:0] + 1 \text{ [lines]}$$

* **VTCNT[9:0]**: Vertical Total Period (VT) Setup Bits in the Vertical Display Register (D[25:16]/0x301A14)

VDP: Vertical display period

Use VDPCNT[9:0] (D[9:0]/0x301A14) to set the vertical display period (= vertical panel resolution).

$$VDP = VDPCNT[9:0] + 1 \text{ [lines]}$$

* **VDPCNT[9:0]**: Vertical Display Period (VDP) Setup Bits in the Vertical Display Register (D[9:0]/0x301A14)

VDPCNT[9:0] (D[9:0]/0x301A14) must be programmed such that the following condition is met:

$$VT \geq VDP + 1$$

FPSHIFT mask for monochrome LCD panel

When color passive panel is selected (COLOR (D30/0x301A60) = 1), the FPSHIFT clock is output during the horizontal display period (HDP) and it stops during the horizontal non-display period (HNDP) as shown in Figure VIII.1.5.2.2.

When monochrome passive panel is selected (COLOR (D30/0x301A60) = 0), the FPSHIFT clock does not stop even in the horizontal non-display period by the default setting. To stop the FPSHIFT clock during the horizontal non-display period, set FPSMASK (D29/0x301A60) to 1.

* **FPSMASK**: FPSHIFT Mask Enable Bit in the LCDC Display Mode Register (D29/0x301A60)

Note: When using an STN panel, the registers for setting the HR-TFT timing parameters must be set to 0x0.

VIII.1.5.3 HR-TFT Panel Timing Parameters

The HR-TFT panel timing parameters shown in Figures below can be set using the LCDC control registers.

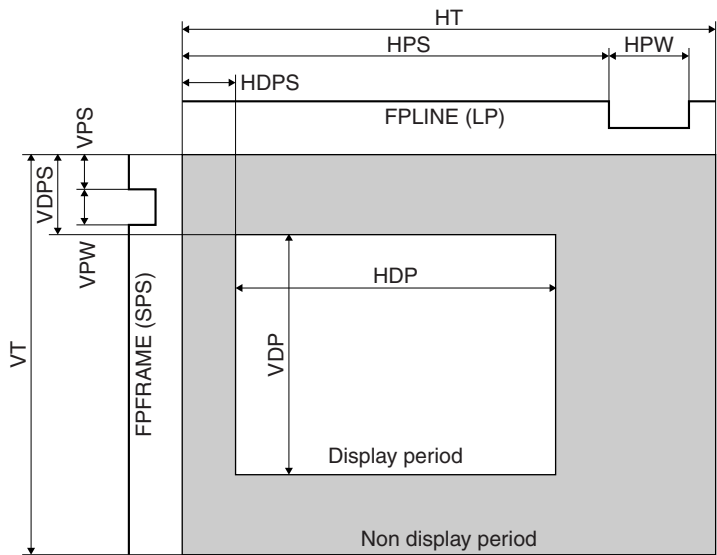


Figure VIII.1.5.3.1 HR-TFT Panel Timing Parameters

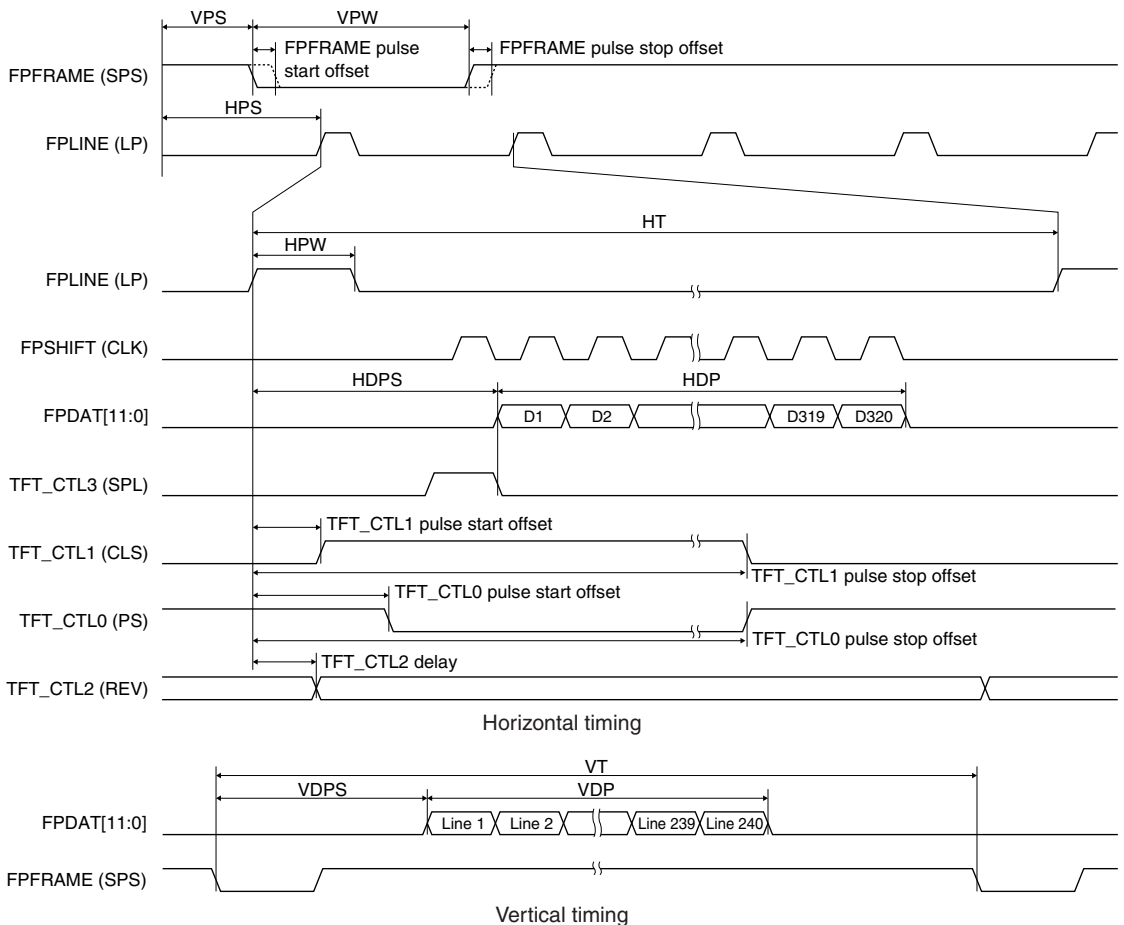


Figure VIII.1.5.3.2 HR-TFT Panel Timing Chart

HT: Horizontal total period

Use HTCNT[6:0] (D[22:16]/0x301A10) to set the horizontal total period.

$$HT = (HTCNT[6:0] + 1) \times 8 [Ts] \quad (Ts: \text{Pixel clock period})$$

* **HTCNT[6:0]**: Horizontal Total Period (HT) Setup Bits in the Horizontal Display Register (D[22:16]/0x301A10)

HTCNT[6:0] (D[22:16]/0x301A10) must be programmed such that the following conditions are met:

$$HTCNT[6:0] \geq HDPCNT[6:0] + 3$$

$$HT > HDP + HDPS$$

HDP: Horizontal display period

Use HDPCNT[6:0] (D[6:0]/0x301A10) to set the horizontal display period (= horizontal panel resolution).

$$HDP = (HDPCNT[6:0] + 1) \times 8 [Ts]$$

* **HDPCNT[6:0]**: Horizontal Display Period (HDP) Setup Bits in the Horizontal Display Register (D[6:0]/0x301A10)

HDPCNT[6:0] (D[6:0]/0x301A10) must be programmed such that the following condition is met:

$$HDP \geq 16 \quad (HDPCNT[6:0] \geq 1)$$

HDPS: Horizontal display period start position

Use HDPCNT[9:0] (D[9:0]/0x301A20) to set the horizontal display period start position for the HR-TFT panel.

$$HDPS = HDPCNT[9:0] + 1 [Ts]$$

* **HDPCNT[9:0]**: Horizontal Display Period Start Position Setup Bits in the Horizontal Display Start Position Register (D[9:0]/0x301A20)

HDPCNT[9:0] (D[9:0]/0x301A20) must be programmed such that the following condition is met:

$$HT > HDP + HDPS$$

HPS: Horizontal sync pulse start position

Use FPLST[9:0] (D[25:16]/0x301A28) to set the horizontal sync pulse (FPLINE or LP) start position for the HR-TFT panel.

$$HPS = FPLST[9:0] + 1 [Ts]$$

* **FPLST[9:0]**: FPLINE Pulse Start Position Setup Bits in the FPLINE Pulse Setup Register (D[25:16]/0x301A28)

HPW: Horizontal sync pulse width

Use FPLWD[6:0] (D[6:0]/0x301A28) to set the horizontal sync pulse width for the HR-TFT panel.

$$HPW = FPLWD[6:0] + 1 [Ts]$$

* **FPLWD[6:0]**: FPLINE Pulse Width Setup Bits in the FPLINE Pulse Setup Register (D[6:0]/0x301A28)

Horizontal sync pulse polarity

Use FPLPOL (D7/0x301A28) to set the horizontal sync pulse polarity for the HR-TFT panel.

FPLPOL = 1: Active high

FPLPOL = 0: Active low (default)

* **FPLPOL**: FPLINE Pulse Polarity Setup Bit in the FPLINE Pulse Setup Register (D7/0x301A28)

VT: Vertical total period

Use VTCNT[9:0] (D[25:16]/0x301A14) to set the vertical total period.

$$VT = VTCNT[9:0] + 1 [\text{lines}]$$

* **VTCNT[9:0]**: Vertical Total Period (VT) Setup Bits in the Vertical Display Register (D[25:16]/0x301A14)

VTCNT[9:0] (D[25:16]/0x301A14) must be programmed such that the following condition is met:

$$VT > VDP + VDPS$$

VDP: Vertical display period

Use VDPCNT[9:0] (D[9:0]/0x301A14) to set the vertical display period (= vertical panel resolution).

$$VDP = VDPCNT[9:0] + 1 \text{ [lines]}$$

* **VDPCNT[9:0]**: Vertical Display Period (VDP) Setup Bits in the Vertical Display Register (D[9:0]/0x301A14)

VDPCNT[9:0] (D[9:0]/0x301A14) must be programmed such that the following condition is met:

$$VT \geq VDP + 1$$

VDPS: Vertical display period start position

Use VDPSCNT[9:0] (D[9:0]/0x301A24) to set the vertical display period start position for the HR-TFT panel.

$$VDPS = VDPSCNT[9:0] \text{ [lines]}$$

* **VDPSCNT[9:0]**: Vertical Display Period Start Position Setup Bits in the Vertical Display Start Position Register (D[6:0]/0x301A24)

VDPSCNT[9:0] (D[9:0]/0x301A24) must be programmed such that the following condition is met:

$$VT > VDP + VDPS$$

VPS: Vertical sync pulse start position

Use FPFST[9:0] (D[25:16]/0x301A2C) to set the vertical sync pulse (FPFRAME or SPS) start position for the HR-TFT panel.

$$VPS = FPFST[9:0] \text{ [lines]} = FPFST[9:0] \times HT \text{ [Ts]}$$

* **FPFST[9:0]**: FPFRAME Pulse Start Position Setup Bits in the FPFRAME Pulse Setup Register (D[25:16]/0x301A2C)

VPW: Vertical sync pulse width

Use FPFWD[2:0] (D[2:0]/0x301A2C) to set the vertical sync pulse width for the HR-TFT panel.

$$VPW = FPFWD[2:0] + 1 \text{ [lines]} = (FPFWD[2:0] + 1) \times HT \text{ [Ts]}$$

* **FPFWD[2:0]**: FPFRAME Pulse Width Setup Bits in the FPFRAME Pulse Setup Register (D[2:0]/0x301A2C)

Vertical sync pulse polarity

Use FPFPOL (D7/0x301A2C) to set the vertical sync pulse polarity for the HR-TFT panel.

FPFPOL = 1: Active High

FPFPOL = 0: Active low (default)

* **FPFPOL**: FPFRAME Pulse Polarity Setup Bit in the FPFRAME Pulse Setup Register (D7/0x301A2C)

Vertical sync pulse offset

The vertical sync pulse position and width that are basically set in line units can be adjusted in pixel clock units.

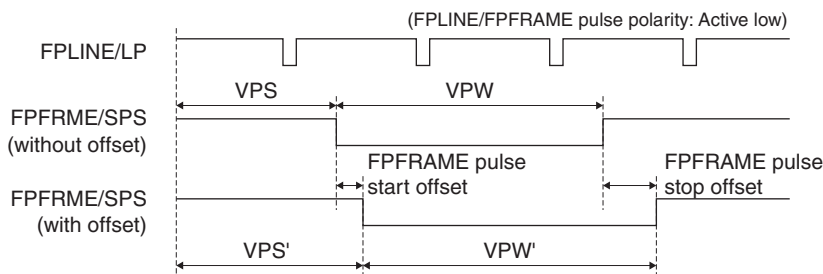


Figure VIII.1.5.3.3 Vertical Sync Pulse Offset

Use FPFSTO[9:0] (D[9:0]/0x301A30) and FPFSTPO[9:0] (D[25:16]/0x301A30) to adjust the vertical sync pulse start and stop positions.

$$VPS' = FPFST[9:0] \times HT + FPFSTO[9:0] \text{ [Ts]}$$

$$VPW' = (FPFWD[2:0] + 1) \times HT - FPFSTO[9:0] + FPFSTPO[9:0] \text{ [Ts]}$$

* **FPFSTO[9:0]**: FPFRAME Pulse Start Offset Bits in the FPFRAME Pulse Offset Register (D[9:0]/0x301A30)

* **FPFSTPO[9:0]**: FPFRAME Pulse Stop Offset Bits in the FPFRAME Pulse Offset Register (D[25:16]/0x301A30)

FPSHIFT (CLK) signal

The FPSHIFT (CLK) signal polarity for HR-TFT panels can be selected using FPSPOL (D1/0x301A40).

- * **FPSPOL**: FPSHIFT Polarity Select Bit in the HR-TFT Special Output Register (D1/0x301A40)

When HR-TFT panel is selected (TFTSEL (D31/0x301A60) = 1), the FPSHIFT (CLK) clock does not stop even in the horizontal non-display period by the default setting. To stop the FPSHIFT clock during the horizontal non-display period, set FPSMASK (D29/0x301A60) to 1.

- * **FPSMASK**: FPSHIFT Mask Enable Bit in the LCDC Display Mode Register (D29/0x301A60)

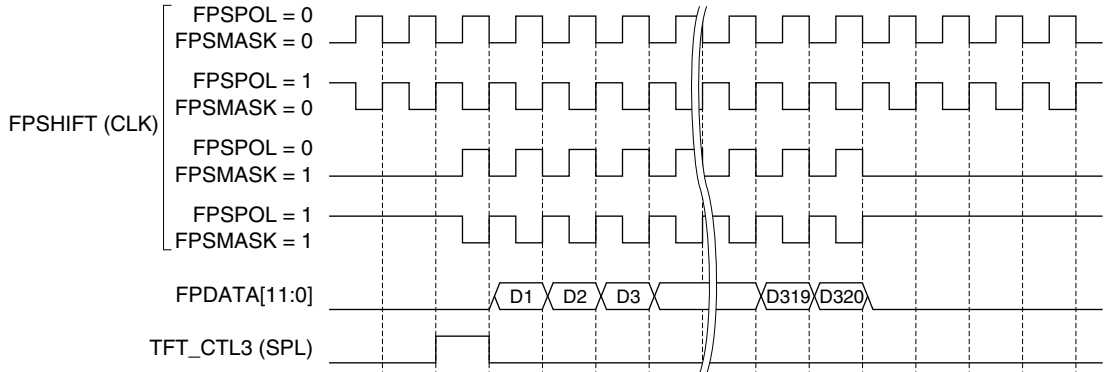


Figure VIII.1.5.3.4 FPSHIFT (CLK) Variations

TFT_CTL1 (CLS) pulse start/stop offset

The TFT_CTL1 (CLS) pulse position and width can be specified in pixel clock cycles. Use CTL1ST[9:0] (D[9:0]/0x301A44) to set the pulse start position and CTL1STP[9:0] (D[25:16]/0x301A44) to set the pulse stop position. These values should be specified an offset from the FPLINE pulse start position.

- * **CTL1ST[9:0]**: TFT_CTL1 Pulse Start Offset Setup Bits in the TFT_CTL1 Pulse Register (D[9:0]/0x301A44)
- * **CTL1STP[9:0]**: TFT_CTL1 Pulse Stop Offset Setup Bits in the TFT_CTL1 Pulse Register (D[25:16]/0x301A44)

By setting this register, the TFT_CTL1 pulse width is set to $CTL1STP[9:0] - CTL1ST[9:0] + 1$ [Ts].

To program the TFT_CTL1 pulse, CTL1CTL (D3/0x301A40) and PRESET (D2/0x301A40) must be set to 1.

- * **CTL1CTL**: TFT_CTL1 Control Bit in the HR-TFT Special Output Register (D3/0x301A40)
- * **PRESET**: TFT_CTL0–2 Preset Enable Bit in the HR-TFT Special Output Register (D2/0x301A40)

When CTL1CTL (D3/0x301A40) is set to 0 (default), the TFT_CTL1 pulse is toggled at the FPLINE pulse start edge.

The TFT_CTL1 and TFT_CTL0 signals can be swapped using CTLSWAP (D0/0x301A40).

TFT_CTL1 pin: CLS output (CTLSWAP = 0), PS output (CTLSWAP = 1)

TFT_CTL0 pin: PS output (CTLSWAP = 0), CLS output (CTLSWAP = 1)

- * **CTLSWAP**: TFT_CTL0/TFT_CTL1 Swap Bit in the HR-TFT Special Output Register (D0/0x301A40)

TFT_CTL0 (PS) pulse start/stop offset

The TFT_CTL0 (PS) pulse position and width can be specified in pixel clock cycles. Use CTL0ST[9:0] (D[9:0]/0x301A48) to set the pulse start position and CTL0STP[9:0] (D[25:16]/0x301A48) to set the pulse stop position. These values should be specified an offset from the FPLINE pulse start position.

- * **CTL0ST[9:0]**: TFT_CTL0 Pulse Start Offset Setup Bits in the TFT_CTL0 Pulse Register (D[9:0]/0x301A48)
- * **CTL0STP[9:0]**: TFT_CTL0 Pulse Stop Offset Setup Bits in the TFT_CTL0 Pulse Register (D[25:16]/0x301A48)

By setting this register, the TFT_CTL0 pulse width is set to $CTL0STP[9:0] - CTL0ST[9:0] + 1$ [Ts].

To program the TFT_CTL0 pulse, PRESET (D2/0x301A40) must be set to 1.

The TFT_CTL1 and TFT_CTL0 signals can be swapped using CTLSWAP (D0/0x301A40).

TFT_CTL2 (REV) delay

Use CTL2DLY[9:0] (D[9:0]/0x301A4C) to set the TFT_CTL2 toggle edge delay time from the FPLINE pulse start edge.

* **CTL2DLY[9:0]**: TFT_CTL2 Delay Setup Bits in the TFT_CTL2 Register (D[9:0]/0x301A4C)

To program the TFT_CTL2 delay time, PRESET (D2/0x301A40) must be set to 1.

VIII.1.5.4 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). Use BPP[2:0] (D[2:0]/0x301A60) to set a display (bpp) mode.

* **BPP[2:0]**: Bit-Per-Pixel Select Bits in the LCDC Display Mode Register (D[2:0]/0x301A60)

Table VIII.1.5.4.1 Specification of Display Modes

BPP2	BPP1	BPP0	Display mode		
			Monochrome (COLOR = 0)	Color (COLOR = 1), STN	Color (COLOR = 1), TFT
1	1	*	Reserved	Reserved	Reserved
1	0	1	Reserved	16 bpp, 64K colors	Reserved
1	0	0	Reserved	12 bpp, 4K colors	12 bpp, 4K colors
0	1	1	Reserved	8 bpp, 256 colors	8 bpp, 256 colors
0	1	0	4 bpp, 16 gray levels	Reserved	4 bpp, 16 colors
0	0	1	2 bpp, 4 gray levels	Reserved	2 bpp, 4 colors
0	0	0	1 bpp, 2 gray levels	Reserved	1 bpp, 2 colors

(Default: 0b000)

(1) 1-bpp (2-gray-level/2-color) mode

One pixel is represented by 1 bit, displayed in two gray levels (STN) or two colors (TFT).

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.

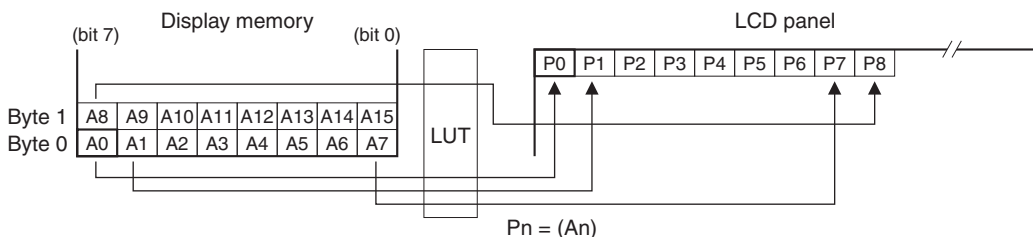


Figure VIII.1.5.4.1 Data Format in 1-bpp Mode

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 (STN) or four colors (TFT).

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.

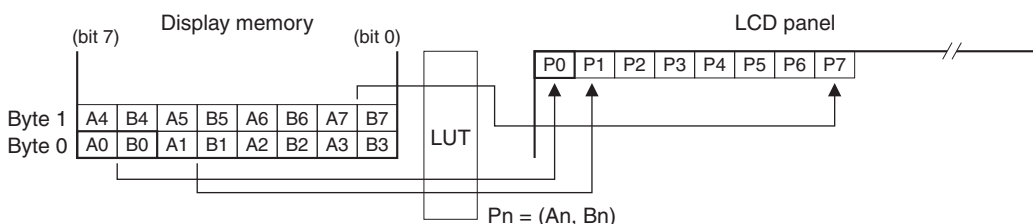


Figure VIII.1.5.4.2 Data Format in 2-bpp Mode

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 (STN) or 16 colors (TFT).

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.

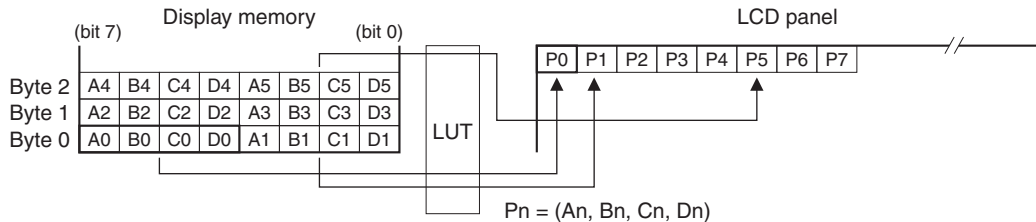


Figure VIII.1.5.4.3 Data Format in 4-bpp Mode

The look-up table can be bypassed in this mode to reduce power consumption.

(4) 8-bpp (256-color) mode

One pixel is represented by 8 bits, displayed in 256 colors.

In this mode, 256 discrete combinations are configured using 8 entries in the red and green look-up tables, and 4 entries in the blue look-up table.

Data for one pixel is stored as one byte in the display memory.

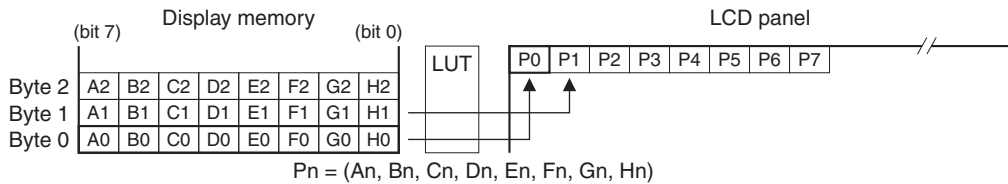


Figure VIII.1.5.4.4 Color Data Format in 8-bpp Mode

The look-up tables can be bypassed in this mode. In this case, the display data stored in the display memory directly specifies a color. The following figure shows the correspondence between the memory data and the pixel data to be sent to the LCD panel.

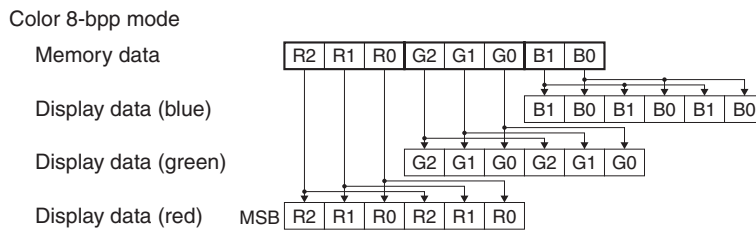


Figure VIII.1.5.4.5 Pixel Data when LUT is Bypassed

(5) 12-bpp (4K-color) mode

One pixel is represented by 12 bits, displayed in 4K colors.

In this mode, 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.

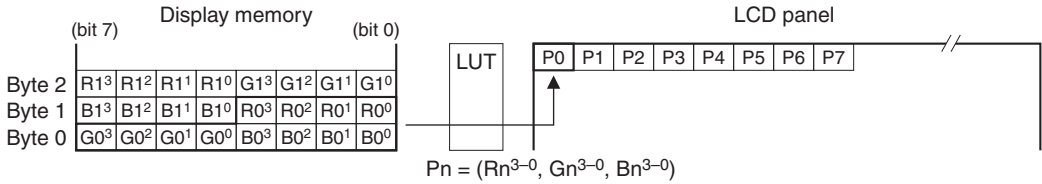


Figure VIII.1.5.4.6 Color Display Data Format in 12-bpp Mode

The look-up tables can be bypassed in this mode. In this case, the display data stored in the display memory directly specifies a color. The following figure shows the correspondence between the memory data and the pixel data to be sent to the LCD panel.

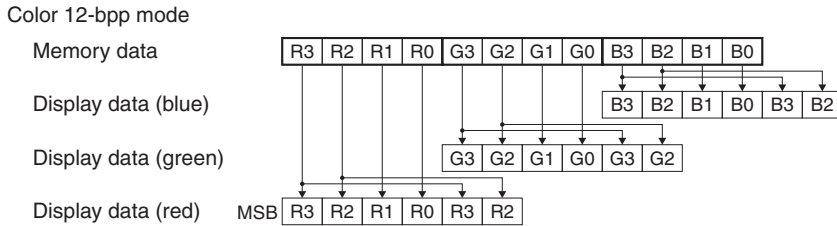


Figure VIII.1.5.4.7 Pixel Data when LUT is Bypassed

(6) 16-bpp (64K-color) mode

One pixel is represented by 16 bits, displayed in 64K colors (STN).

Data for one pixel is stored as two bytes in the display memory.

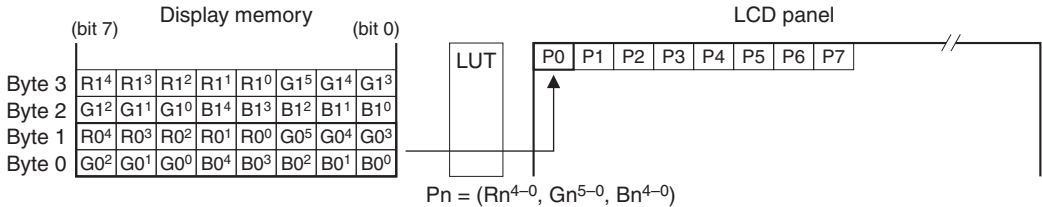


Figure VIII.1.5.4.8 Color Display Data Format in 16-bpp Mode

The look-up tables are bypassed in this mode, and the display data stored in the display memory directly specifies a color. The following figure shows the correspondence between the memory data and the pixel data to be sent to the LCD panel.

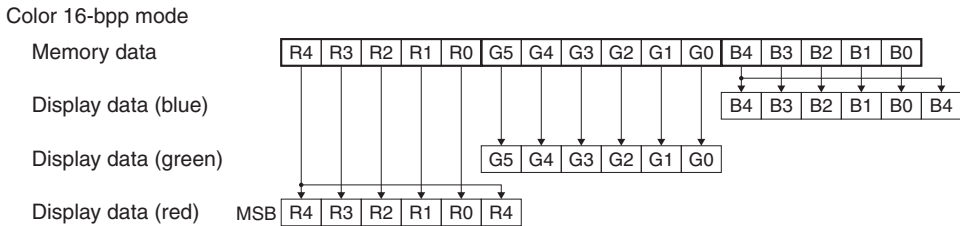


Figure VIII.1.5.4.9 Pixel Data when LUT is Bypassed

In addition to the bpp mode selection, set DITHEN (D6/0x301A60) to 1 to display in 64K colors. When DITHEN (D6/0x301A60) is set to 0, the number of colors that can be displayed is limited to 4K colors.

* **DITHEN**: Dither Mode Enable Bit in the LCDC Display Mode Register (D6/0x301A60)

Note: 16-bpp mode cannot be set when a HR-TFT panel is used.

VIII.1.5.5 Look-up Tables

The LCD controller contains a look-up table consisting of 16 6-bit entries, one for each of the RGB color elements (red, green, and blue).

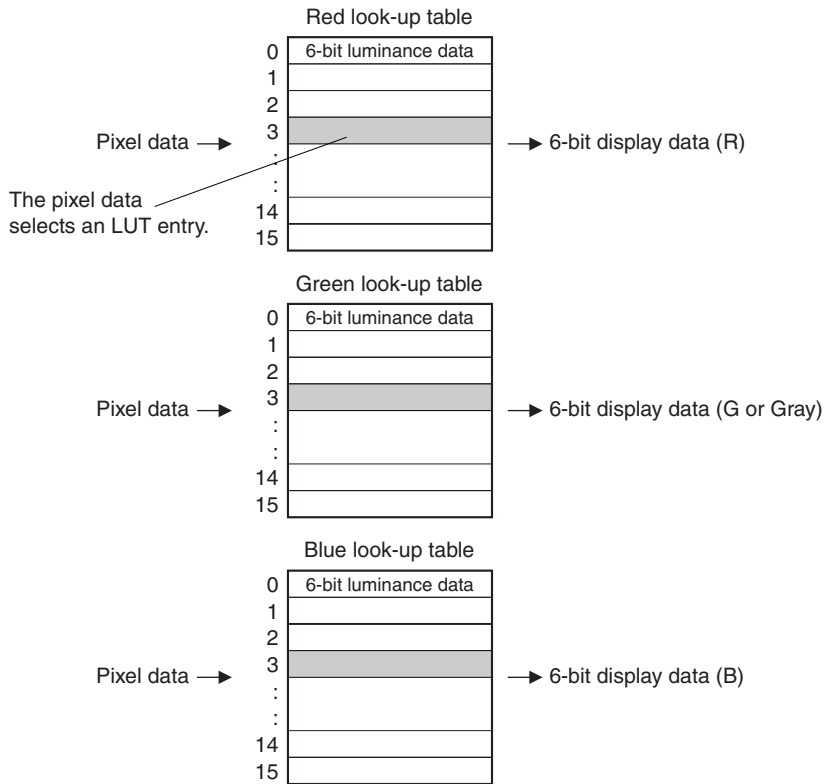


Figure VIII.1.5.5.1 Configuration of the Look-up Tables

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, and 12-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 (D4/0x301A60) to 1.

* **LUTPASS**: LUT Bypass Mode Select Bit in the LCDC Display Mode Register (D4/0x301A60)

The look-up table cannot be used in 1-, 2-, 4-, and 16-bpp color mode.

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.

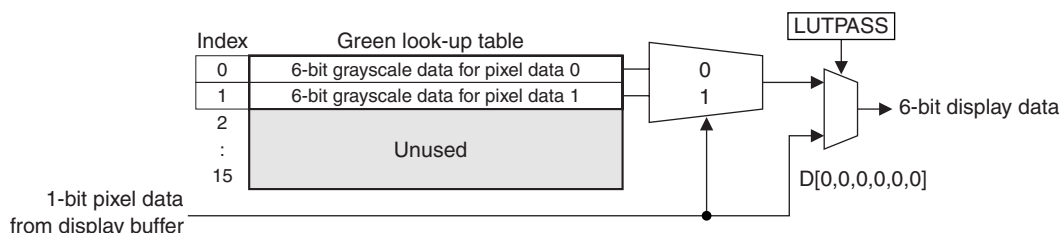


Figure VIII.1.5.5.2 Look-up Table in 1-bpp (2-Gray-Level) Mode

Table VIII.1.5.5.1 shows an example of the basic data setting.

Table VIII.1.5.5.1 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
2-15	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.

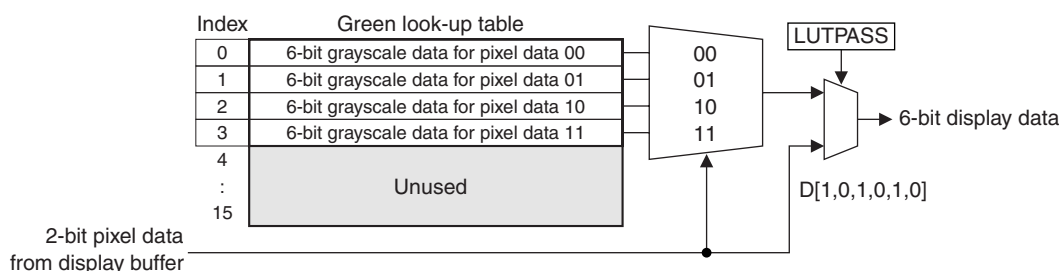


Figure VIII.1.5.5.3 Look-up Table in 2-bpp (4-Gray-Level) Mode

Table VIII.1.5.5.2 shows an example of the basic data setting.

Table VIII.1.5.5.2 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
4-15	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 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.

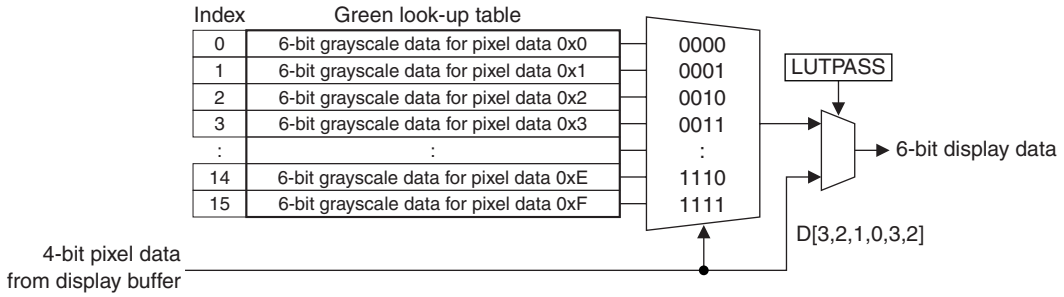


Figure VIII.1.5.5.4 Look-up Table in 4-bpp (16-Gray-Level) Mode

Table VIII.1.5.5.3 shows an example of the basic data setting.

Table VIII.1.5.5.3 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

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•3F•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 (256 or 4K 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) 8-bpp (256-color) mode

One pixel is represented by 8 bits, displayed in 256 colors. In this mode, 256 discrete combinations are configured using 8 entries in each of the red and green look-up tables, and 4 entries in the blue look-up table. Bits 7–5 in 8 bits of pixel data are used as an index to the red look-up table, while bits 4–2 and bits 1–0 are used as indices to the green and blue look-up tables, respectively.

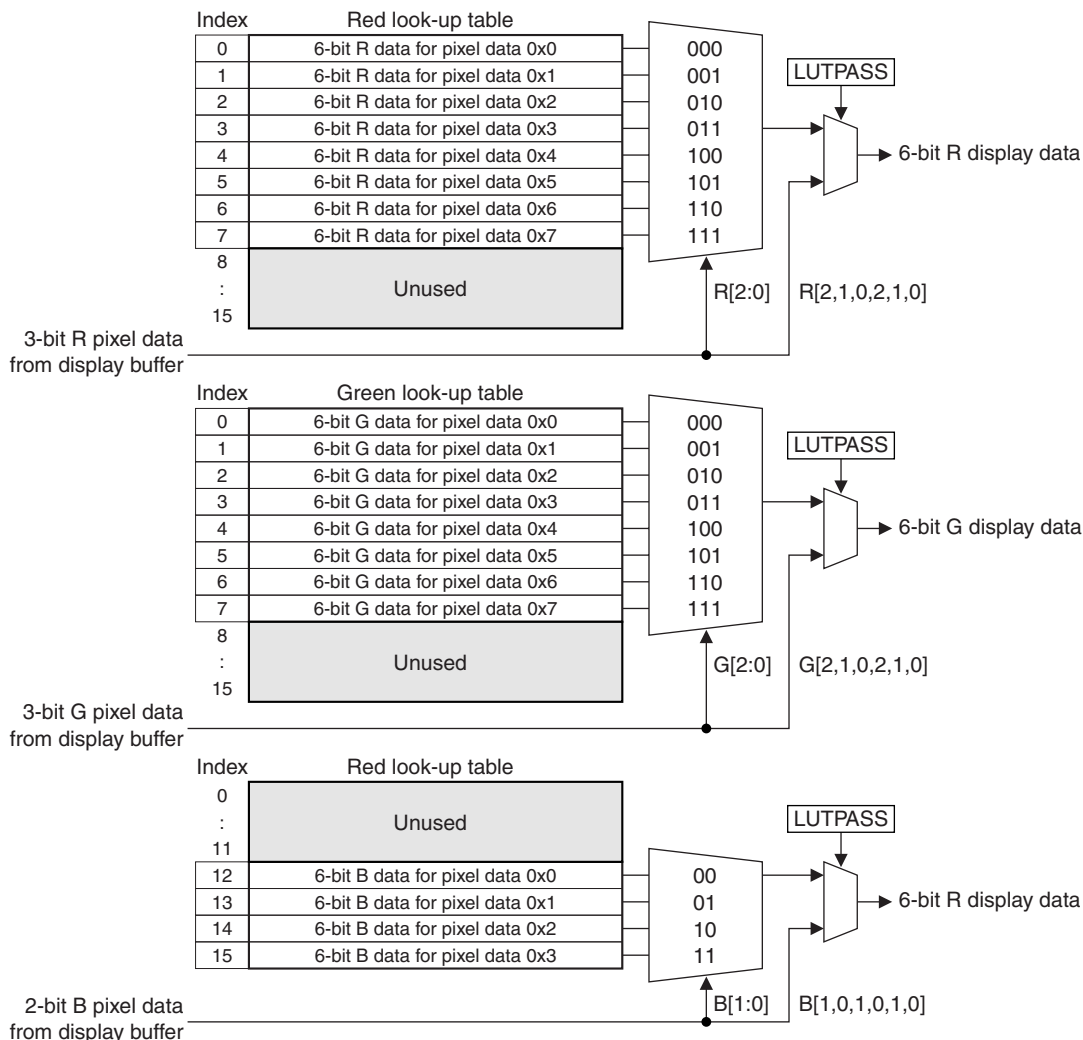


Figure VIII.1.5.5 Look-up Table in 8-bpp (256-Color) Mode

(2) 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.

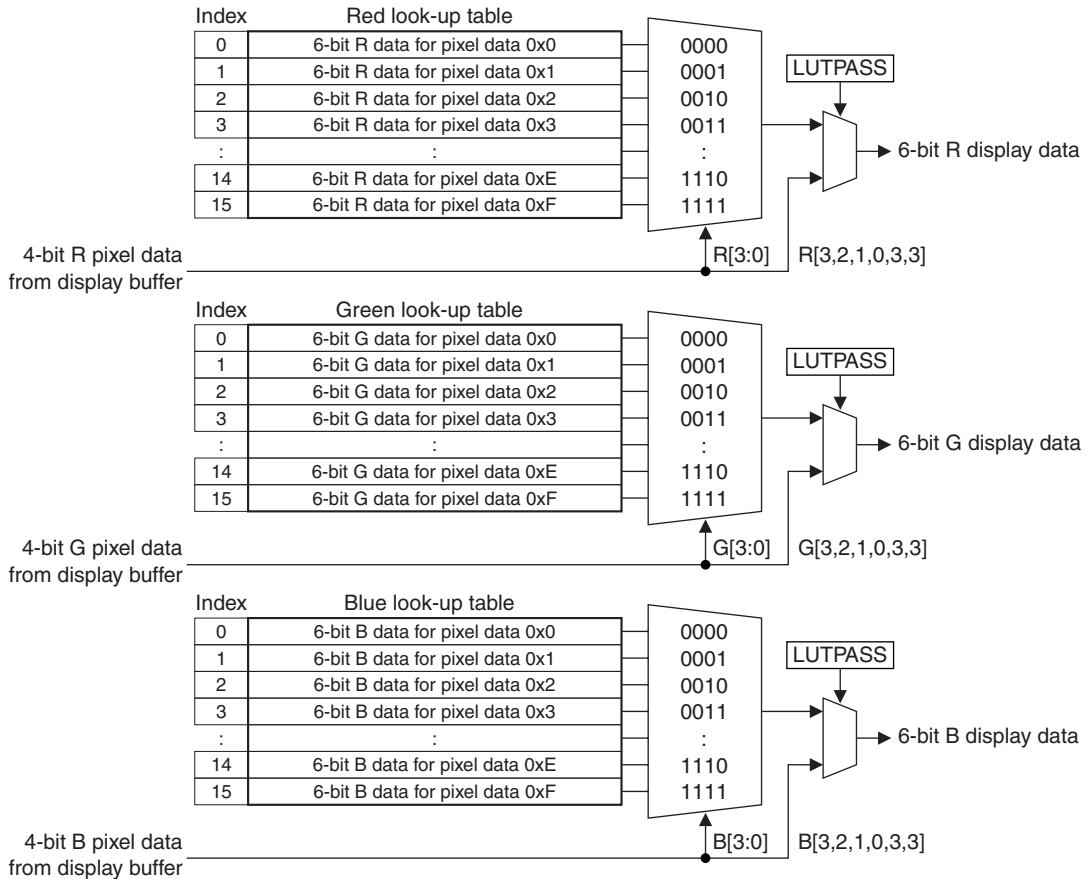


Figure VIII.1.5.5.6 Look-up Table in 12-bpp (4K-Color) Mode

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 0x301AA0 to 0x301AAC.

Table VIII.1.5.5.4 Look-up Table Register Address

Address	Data bit	LUT entry
0x301AA0	D[7:0] (D[1:0] are ineffective)	0
	D[15:8] (D[9:8] are ineffective)	1
	D[23:16] (D[17:16] are ineffective)	2
	D[31:24] (D[25:24] are ineffective)	3
0x301AA4	D[7:0] (D[1:0] are ineffective)	4
	D[15:8] (D[9:8] are ineffective)	5
	D[23:16] (D[17:16] are ineffective)	6
	D[31:24] (D[25:24] are ineffective)	7
0x301AA8	D[7:0] (D[1:0] are ineffective)	8
	D[15:8] (D[9:8] are ineffective)	9
	D[23:16] (D[17:16] are ineffective)	10
	D[31:24] (D[25:24] are ineffective)	11
0x301AAC	D[7:0] (D[1:0] are ineffective)	12
	D[15:8] (D[9:8] are ineffective)	13
	D[23:16] (D[17:16] are ineffective)	14
	D[31:24] (D[25:24] are ineffective)	15

Write data to all the entry to be used. Although each entry is 6-bit width, the registers above allow writing with 8-bit data corresponding to the LCD output (6 high-order bits are effective). Write each data with the low-order 2 bits set to 0. The data written to the register is set to the R, G, and B look-up tables simultaneously. No writing sequence is defined. Writing can be started from any entry.

VIII.1.5.6 Frame Rates

The frame rate is calculated from the LCD panel's horizontal and vertical total periods, and pixel clock frequency, as shown below.

$$\text{Frame rate} = \frac{f_{\text{LCDC_CLK}}}{\text{HT} \times \text{VT}}$$

$f_{\text{LCDC_CLK}}$: Pixel clock frequency

$f_{\text{LCDC_CLK}} = \text{OSC3/1 to OSC3/16 (Hz)}$ see Section III.1.9.3, "Clock Supply to the LCDC."

HT: Horizontal total period

$\text{HT} = (\text{HTCNT}[6:0] + 1) \times 8 (T_s)$ where $T_s = \text{pixel clock cycle}$

* **HTCNT[6:0]**: Horizontal Total Period (HT) Setup Bits in the Horizontal Display Register (D[22:16]/0x301A10)

VT: Vertical total period

$\text{VT} = \text{VTCNT}[9:0] + 1 (\text{lines})$

* **VTCNT[9:0]**: Vertical Total Period (VT) Setup Bits in the Vertical Display Register (D[25:16]/0x301A14)

VIII.1.5.7 Other Settings

MOD rate

The period during which the MOD signal is switched can be set using the MOD[5:0] (D[5:0]/0x301A18).

MOD = 0x0: MOD signal switched at a period of the FPFAME signal (default)

MOD = other than 0x0: Switched at a period of MOD + 1 FPLINE pulses

* **MOD[5:0]**: LCD MOD Rate Setup Bits in the MOD Rate Register (D[5:0]/0x301A18)

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 (D7/0x301A60).

FRMRPT = 1: FRM pattern repeated (for EL panel)

FRMRPT = 0: FRM pattern not repeated (default)

* **FRMRPT**: Frame Repeat for EL Panel Bit in the LCDC Display Mode Register (D7/0x301A60)

VIII.1.6 Display Control

VIII.1.6.1 Controlling LCD Power Up/Down

The LCD controller is activated when the LCDC clocks are supplied from the CMU. Following initial reset, the LCD controller is set in power-save mode. Supplying the clocks 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 PSAVE[1:0] (D[1:0]/0x301A04) 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 PSAVE[1:0] (D[1:0]/0x301A04) 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.

* **PSAVE[1:0]**: Power Save Mode Enable Bits in the Status and Power Save Configuration Register (D[1:0]/0x301A04)

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.

The procedure for initializing the LCD at power-on is summarized below.

1. Configure the clocks, pins, and display memory area (refer to “VIII.1.4 System Settings”).
2. Set the LCD-panel parameters, display mode, and look-up tables (refer to “VIII.1.5 Setting the LCD Panel”).
3. Enable the LCDC interrupt.
4. Write display data to the display memory.
5. Set the display start address (refer to “VIII.1.6.2 Setting the Display Start Address and Line Address Offset”).
6. Place the LCD controller in normal mode (PSAVE = 0b11).
7. The LCD controller starts outputting the LCD signals.
8. Wait time should be inserted depending on the LCD panel power source.
9. 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 (PSAVE = 0b00).
4. The LCD controller pulls LCD signals down to low.

VIII.1.6.2 Setting the Display Start Address and Line Address Offset

Display start address

The display memory address from which to start display for the main window can be changed as desired using the Main Window Display Start Address Register (0x301A70). The start address set in the Main Window Display Start Address Register (0x301A70) corresponds to the upper left edge of the LCD panel. Note that a word boundary address (A[1:0] = 0b00) in IVRAM or the external SDRAM must be specified to this register.

Line address offset

The S1C33E08 LCDC manipulates display data in units of words. Therefore, the image width (number of pixels) must be a multiple of (32 bits ÷ bpp). The line address offset is the number of words corresponding to the image width and it should be specified in the following cases:

1. When the Picture-in-Picture Plus function is used (when a sub-window is displayed)
2. When the LCD panel horizontal resolution is not a multiple of (32 bits ÷ bpp)

The line address offset is calculated as follows:

$$\text{Main window line address offset} = \text{Main window width in pixels} \times \text{bpp} / 32$$

Set this value to MWLADR[9:0] (D[9:0]/0x301A74).

* **MWLADR[9:0]**: Main Window Line Address Offset Bits in the Main Window Line Address Offset Register (D[9:0]/0x301A74)

(1) When the LCD panel horizontal resolution is a multiple of (32 bits ÷ bpp)

Example 1) LCD panel width = 320 pixels, 1-bpp mode

$$\text{Main window line address offset} = 320 \times 1 / 32 = 10 \text{ [words]}$$

Example 2) LCD panel width = 160 pixels, 8-bpp mode

$$\text{Main window line address offset} = 160 \times 8 / 32 = 40 \text{ [words]}$$

When using the Picture-in-Picture Plus function, set the calculated value to MWLADR[9:0] (D[9:0]/0x301A74).

When the Picture-in-Picture Plus function is not used, it is not necessary to set MWLADR[9:0] (D[9:0]/0x301A74). Leave it unaltered from 0 or set the calculated value to MWLADR[9:0] (D[9:0]/0x301A74) to display normally. Be sure to avoid setting another value.

(2) When the LCD panel horizontal resolution is not a multiple of (32 bits ÷ bpp)

Example 3) LCD panel width = 240 pixels, 1-bpp mode

$$\text{Main window line address offset} = 240 \times 1 / 32 = 7.5 \text{ [words]} \rightarrow 8 \text{ [words]}$$

In this case, the calculated results have a decimal fraction. It must be rounded up.

MWLADR[9:0] (D[9:0]/0x301A74) must be set to 8.

Furthermore, the image area with 256 (8 × 32) pixels wide must be prepared in the display memory.

Image data should be left justified in the area. The 0.5 word data (16 pixels) at the end of each line will not be displayed.

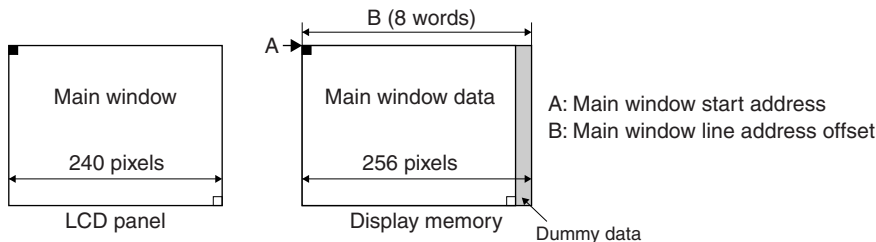


Figure VIII.1.6.2.1 Data Layout in Image Area with Offset

When the LCD panel horizontal resolution is not a multiple of (32 bits ÷ bpp), a main window line address offset must be set to MWLADR[9:0] (D[9:0]/0x301A74) regardless of whether the Picture-in-Picture Plus function is used or not.

The main window line address offset corresponds to one display line in the display memory. By adding/subtracting this value or a multiple to/from the Main Window Display Start Address Register (0x301A70), the image can be scrolled vertically.

VIII.1.6.3 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, VNDPF (D7/0x301A04) is provided and is set to 1 if the display is in a vertical non-display period.

* **VNDPF**: Vertical Display Status Flag in the Status and Power Save Configuration Register (D7/0x301A04)

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 VNDPF (D7/0x301A04).

For more information on the LCDC interrupt, see Section VIII.1.7, "LCDC Interrupt and DMA."

VIII.1.6.4 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 BLANK (D24/0x301A60) 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 SWINV (D25/0x301A60) described below.

* **BLANK**: Display Blank Enable Bit in the LCDC Display Mode Register (D24/0x301A60)

* **SWINV**: Software Video Invert Bit in the LCDC Display Mode Register (D25/0x301A60)

Furthermore, the display can be inverted simply by manipulating a control bit. Setting SWINV (D25/0x301A60) to 1 inverts the display, and setting it to 0 returns the display to normal. (This function is effective only for STN panels and it does not support HR-TFT panels.) 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 (D7/0x301A04) = 1).

VIII.1.6.5 Picture-in-Picture Plus

Picture-in-Picture Plus enables a sub-window to be displayed within the main window. The sub-window may be positioned anywhere within the main window and is controlled through the sub-window control registers. The sub-window retains the same color depth as the main window.

The following diagram shows the sub-window configuration parameters.

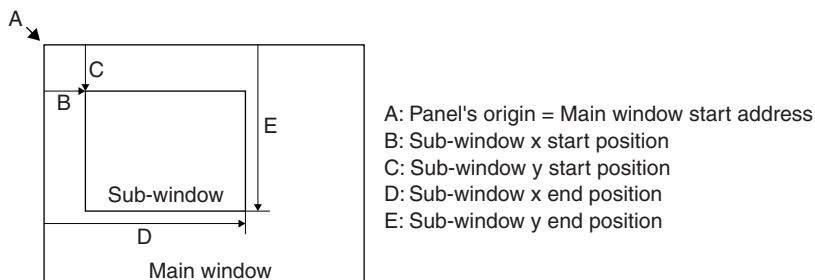


Figure VIII.1.6.5.1 Sub-window Configuration Parameters

Note: When using the Picture-in-Picture Plus function, the main window line address offset must be set to MWLADR[9:0] (D[9:0]/0x301A74). See Section VIII.1.6.2.

Display memory for the sub-window

The display data for the sub-window can be stored in IVRAM or the external SDRAM. If the same memory as the main window is used, make sure that the display data areas for the main window and sub-window do not overlap.

The sub-window start address is specified by the Sub-window Display Start Address Register (0x301A80) in the same manner as the main window. The start address set in the Sub-window Display Start Address Register (0x301A80) corresponds to the upper left corner of the sub-window.

Note that a word boundary address (A[1:0] = 0b00) in IVRAM or the external SDRAM must be specified to this register.

The sub-window width must be a multiple of (32 bits ÷ bpp).

Sub-window coordinates

The display position and size of the sub-window are configured with the X and Y coordinates of the start position (upper left corner) and end position (lower right corner).

Specify the sub-window start position using PIPXST[9:0] (D[9:0]/0x301A88) for the X coordinate and PIPYST[9:0] (D[25:16]/0x301A88) for the Y coordinate. Use PIPXEND[9:0] (D[9:0]/0x301A8C) for specifying the X coordinate of the end position and PIPYEND[9:0] (D[25:16]/0x301A8C) for the Y coordinate.

- * **PIPXST[9:0]**: Sub-window Horizontal (X) Start Position Bits in the Sub-window Start Position Register (D[9:0]/0x301A88)
- * **PIPYST[9:0]**: Sub-window Vertical (Y) Start Position Bits in the Sub-window Start Position Register (D[25:16]/0x301A88)
- * **PIPXEND[9:0]**: Sub-window Horizontal (X) End Position Bits in the Sub-window End Position Register (D[9:0]/0x301A8C)
- * **PIPYEND[9:0]**: Sub-window Vertical (Y) End Position Bits in the Sub-window End Position Register (D[25:16]/0x301A8C)

The X coordinate should be specified with the number of data words converted from the number of pixels from the LCD panel origin point according to the bpp mode. Therefore, it can be specified in (32 bits ÷ bpp) pixel increments.

- 1-bpp mode: 1-word = 32-pixel units
- 2-bpp mode: 1-word = 16-pixel units
- 4-bpp mode: 1-word = 8-pixel units
- 8-bpp mode: 1-word = 4-pixel units
- 12-bpp mode: 3-word = 8-pixel units (because the value must be an integer)
- 16-bpp mode: 1-word = 2-pixel units

For example, to specify the sub-window horizontal start position as 80 pixels in 8-bpp mode, set PIPXST[9:0] (D[9:0]/0x301A88) to 20.

The Y coordinate is specified with the number of lines from the LCD panel origin point in line units. For example, to specify the sub-window vertical start position as 60 lines, set PIPYST[9:0] (D[25:16]/0x301A88) to 60.

Sub-window display control

The Picture-in-Picture Plus function is enabled and the sub-window is displayed by setting PIPEN (D31/0x301A88) to 1. This bit must be set after the sub-window configuration parameters are set up.

- * **PIPEN**: PIP Sub-window Enable Bit in the Sub-window Start Position Register (D31/0x301A88)

At initial reset, PIPEN (D31/0x301A88) is set to 0 and sub-window is disabled for display.

Sub-window configuration example

The following shows an example to configure main and sub-windows.

[Conditions]

- LCD panel resolution: 320 × 240 pixels
- bpp mode: 4 bpp (16 shades of gray/16 colors)
- Memory used: External SDRAM (area 19, 0x10000000–)
- Virtual screen size: 320 × 240 pixels
- Sub-window size: 160 × 120 pixels
- Sub-window start position: X = 80 pixels, Y = 60 pixels

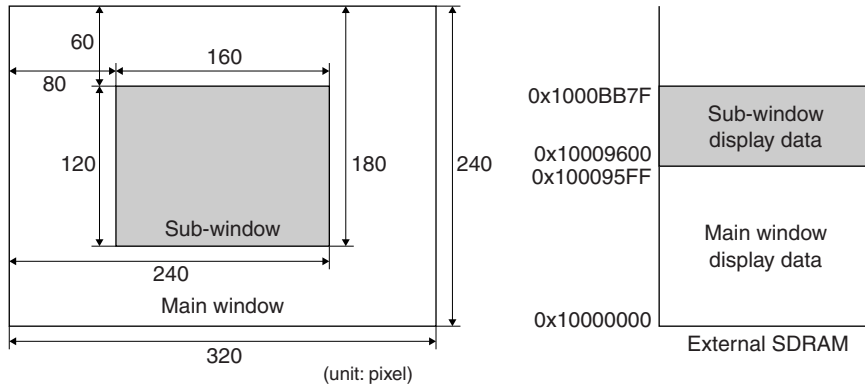


Figure VIII.1.6.5.2 Sub-Window Configuration Example

1. Main window start address

This example assumes that the display data for the main window is located at the beginning of the external SDRAM (area 19).

Main Window Display Start Address Register (0x301A70) = 0x10000000

2. Main window line address offset

Convert the screen width (320 pixels) into the number of words in 4-bpp mode and set it to MWLADR[9:0] (D[9:0]/0x301A74).

MWLADR[9:0] (D[9:0]/0x301A74) = 320 pixels × 4 bpp ÷ 32 bits = 40 words

Main Window Line Address Offset Register (0x301A74) = 40 (= 0x28)

3. Sub-window start address

This example assumes that the sub-window data is located immediately following the main window data area.

Main window data size = 320 pixels × 240 pixels × 4 bpp ÷ 8 bits = 0x9600 bytes

Sub-window Display Start Address Register (0x301A80) = 0x10000000 + 0x9600 = 0x10009600

4. Sub-window start and end positions

The X coordinates should be converted into the number of words in 4-bpp mode to set them to the registers. The Y coordinates can be set to the registers in the line numbers.

Sub-window start position

PIPXST[9:0] (D[9:0]/0x301A88) = 80 pixels × 4 bpp ÷ 32 bits = 10 words (= 0x0A)

PIPYST[9:0] (D[25:16]/0x301A88) = 60 lines (= 0x3C)

Sub-window Start Position Register (0x301A88) = 0x003C000A

Sub-window end position

PIPXEND[9:0] (D[9:0]/0x301A8C) = (80 + 160) pixels × 4 bpp ÷ 32 bits - 1 = 29 words (= 0x1D)

PIPYEND[9:0] (D[25:16]/0x301A8C) = 60 + 120 lines - 1 = 180 lines (= 0xB3)

Sub-window End Position Register (0x301A8C) = 0x00B3001D

After the settings above are completed, write 0x803C000A to the Sub-window Start Position Register (0x301A88) to enable display of the sub-window.

VIII.1.7 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 INTF (D31/0x301A04) 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 INTEN (D0/0x301A00) 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.

- * **INTF**: Frame Interrupt Flag in the Status and Power Save Configuration Register (D31/0x301A04)
- * **INTEN**: Frame Interrupt Enable Bit in the Frame Interrupt Register (D0/0x301A00)

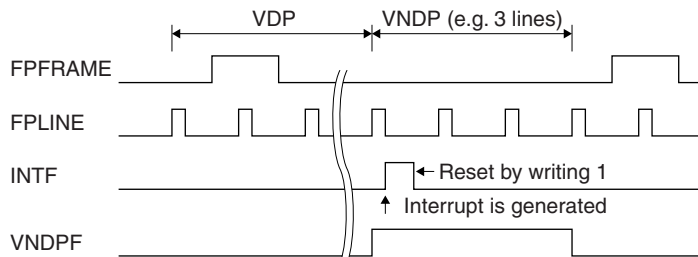


Figure VIII.1.7.1 Frame Interrupt Timing

Once the INTF 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 INTF before INTEN is set to 1 in order to avoid an unnecessary interrupt. When not using the LCDC interrupt, set INTEN to 0.

Note: The LCDC does not support the frame interrupt when the TFT interface is selected (TFTSEL (D31/0x301A60) = 1).

- * **TFTSEL**: HR-TFT Panel Select Bit in the LCDC Display Mode Register (D31/0x301A60)

Control registers of the interrupt controller

Table VIII.1.7.1 shows the ITC's control registers for the LCDC interrupts.

Table VIII.1.7.1 Control Registers of Interrupt Controller

Cause-of-interrupt flag	Interrupt enable register	Interrupt priority register
FLCDC (D1/0x300288)	ELCDC (D1/0x300278)	PLCDC[2:0] (D[2:0]/0x300269)

When the cause of interrupt occurs, the cause-of-interrupt flag is set to 1. If the interrupt enable register bit for that cause of interrupt has been set to 1, an interrupt request is generated.

Interrupts due to a cause of interrupt can be disabled by leaving the interrupt enable register bit for that cause of interrupt set to 0. The cause-of-interrupt 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 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 Section III.2, "Interrupt Controller (ITC)."

Intelligent DMA

The cause of LCDC interrupt 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 DMA transfer.

The IDMA channel number set for the LCDC interrupt is 33.

The IDMA request and enable bits shown in Table VIII.1.7.2 must be set to 1 for IDMA to be invoked. Transfer conditions, etc. on the IDMA side must also be set in advance.

Table VIII.1.7.2 Control Bits for IDMA Transfer

IDMA request bit	IDMA enable bit
RLCDC (D1/0x30029B)	DELCDC (D1/0x30029C)

If a cause of 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 DMA transfer performed.

For details on DMA transfer and how to control interrupts upon completion of a DMA transfer, refer to Section II.2, "Intelligent DMA (IDMA)."

Trap vectors

The trap-vector address for the LCDC interrupt is set to 0x0C00124 by default.

The base address of the trap table can be changed using the TTBR register.

VIII.1.8 Power Save

The LCD controller has two types of power-save modes. Use PSAVE[1:0] (D[1:0]/0x301A04) to set power-save modes.

* **PSAVE[1:0]**: Power Save Mode Select Bits in the Status and Power Save Configuration Register (D[1:0]/0x301A04)

Table VIII.1.8.1 Settings of Power-Save Modes

PSAVE1	PSAVE0	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 PSAVE (D[1:0]/0x301A04) to 0b00.

The LCD controller is taken out of power-save mode by setting PSAVE (D[1:0]/0x301A04) to 0b11.

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 not 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 VIII.1.8.2.

Table VIII.1.8.2 Differences between Power-Save Modes

Item	LCDC disabled	Power-save mode	Doze mode	Normal
Accessing I/O registers	Disabled	Enabled	Enabled	Enabled
Accessing look-up table	Disabled	Enabled	Enabled	Enabled
Accessing VRAM	Enabled	Enabled	Enabled	Enabled
Display (STN panels)	Inactive	Inactive	Active	Active
Display (HR-TFT panels)	Inactive	Inactive	Inactive	Active
LCDC display-data-fetch operation	Inactive	Inactive	Inactive	Active
FPDAT[11:0] signals (STN, HR-TFT panels)	Low	Low	Low	Active
FPSHIFT signal (STN panels)	Low	Low	Low	Active
FPLINE, FPFRAME, FPDRDY signals (STN panels)	Low	Low	Active	Active
FPSHIFT signal (HR-TFT panels) when FPSPOL (D1/0x301A40) = 0	Low	High	High	Active
FPSHIFT signal (HR-TFT panels) when FPSPOL (D1/0x301A40) = 1	Low	Low	Low	Active
FPFRAME signal (HR-TFT panels) when FPFPOL (D7/0x301A2C) = 0	Low	High	Active	Active
FPFRAME signal (HR-TFT panels) when FPFPOL (D7/0x301A2C) = 1	Low	Low	Active	Active
FPLINE signal (HR-TFT panels) when FPLPOL (D7/0x301A28) = 0	Low	High	Active	Active
FPLINE signal (HR-TFT panels) when FPLPOL (D7/0x301A28) = 1	Low	Low	Active	Active
TFT_CTL1 signal* (HR-TFT panels) when CTL1ST[9:0] (D[9:0]/0x301A44) = 0	Low	High	Active	Active
TFT_CTL1 signal* (HR-TFT panels) when CTL1ST[9:0] (D[9:0]/0x301A44) ≠ 0	Low	Low	Active	Active
TFT_CTL0, TFT_CTL2, TFT_CTL3 signals (HR-TFT panels)	Low	Low	Active	Active

* The TFT_CTL1 signal is configured with CTL1CTL (D3/0x301A40) = 1, PRESET (D2/0x301A40) = 1, and CTL5WAP (D0/0x301A40) = 0.

VIII.1.9 Details of Control Registers

Table VIII.1.9.1 List of LCDC Registers

Address	Register name	Size	Function
0x00301A00	Frame Interrupt Register (pLCDC_INT)	32	Enables LCDC interrupts.
0x00301A04	Status and Power Save Configuration Register (pLCDC_PS)	32	Controls power-save mode and indicates interrupt status.
0x00301A10	Horizontal Display Register (pLCDC_HD)	32	Sets horizontal total and display periods.
0x00301A14	Vertical Display Register (pLCDC_VD)	32	Sets vertical total and display periods.
0x00301A18	MOD Rate Register (pLCDC_MR)	32	Sets MOD rate.
0x00301A20	Horizontal Display Start Position Register (pLCDC_HDPS)	32	Sets horizontal display period start position for HR-TFT.
0x00301A24	Vertical Display Start Position Register (pLCDC_VDPS)	32	Sets vertical display period start position for HR-TFT.
0x00301A28	FPLINE Pulse Setup Register (pLCDC_L)	32	Sets FPLINE pulse for HR-TFT.
0x00301A2C	FPFRAME Pulse Setup Register (pLCDC_F)	32	Sets FPFRAME pulse for HR-TFT.
0x00301A30	FPFRAME Pulse Offset Register (pLCDC_FO)	32	Sets FPFRAME pulse offset for HR-TFT.
0x00301A40	HR-TFT Special Output Register (pLCDC_TSO)	32	Controls HR-TFT signals.
0x00301A44	TFT_CTL1 Pulse Register (pLCDC_TC1)	32	Sets TFT_CTL1 pulse.
0x00301A48	TFT_CTL0 Pulse Register (pLCDC_TC0)	32	Sets TFT_CTL0 pulse.
0x00301A4C	TFT_CTL2 Register (pLCDC_TC2)	32	Sets TFT_CTL2 signal.
0x00301A60	LCDC Display Mode Register (pLCDC_DMD)	32	Sets display mode and controls display.
0x00301A64	IRAM Select Register (pLCDC_IRAM)	32	Selects IRAM allocation.
0x00301A70	Main Window Display Start Address Register (pLCDC_MADD)	32	Sets main window display start address.
0x00301A74	Main Window Line Address Offset Register (pLCDC_MLADD)	32	Sets main window line address offset.
0x00301A80	Sub-window Display Start Address Register (pLCDC_SADD)	32	Sets sub-window display start address.
0x00301A88	Sub-window Start Position Register (pLCDC_SSP)	32	Sets sub-window start position.
0x00301A8C	Sub-window End Position Register (pLCDC_SEP)	32	Sets sub-window end position.
0x00301AA0	Look-up Table Data Register 0 (pLCDC_LUT_03)	32	Look-up table data (entries 0 to 3)
0x00301AA4	Look-up Table Data Register 1 (pLCDC_LUT_47)	32	Look-up table data (entries 4 to 7)
0x00301AA8	Look-up Table Data Register 2 (pLCDC_LUT_8B)	32	Look-up table data (entries 8 to 11)
0x00301AAC	Look-up Table Data Register 3 (pLCDC_LUT_CF)	32	Look-up table data (entries 12 to 15)

The following describes each LCDC control register.

The LCDC control registers are mapped in the 32-bit device area from 0x301A00 to 0x301AAC, and can be accessed only in units of words.

Note: When setting the LCDC control registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x301A00: Frame Interrupt Register (pLCDC_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Frame interrupt register (pLCDC_INT)	00301A00 (W)	D31-1	–	reserved	–		–	–	0 when being read.
		D0	INTEN	Frame interrupt enable	1 Enabled	0 Disabled	0	R/W	

D[31:1] Reserved**D0 INTEN: Frame Interrupt Enable Bit**

Enables the LCDC frame interrupt request.

1 (R/W): Enable

0 (R/W): Disable (default)

When using the frame interrupt, set INTEN 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.

Note: The LCDC does not support the frame interrupt when the TFT interface is selected (TFTSEL (D31/0x301A60) = 1).

0x301A04: Status and Power Save Configuration Register (pLCDC_PS)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Status and power save configuration register (pLCDC_PS)	00301A04 (W)	D31	INTF	Frame interrupt flag	1	Generated	0	Not generated	0	R/W	Reset by writing 1.
		D30–8	–	reserved	–		–	–	–	–	0 when being read.
		D7	VNDPF	Vertical display status	1	VNDP	0	VDP	1	R	
		D6–2	–	reserved	–		–	–	–	–	0 when being read.
		D1	PSAVE1	Power save mode	PSAVE[1:0]		Mode		0	R/W	
		D0	PSAVE0		1	1	Normal operation		0		
		1	0		Doze mode						
		0	1		reserved						
				0	0	Power save mode					

D31 INTF: Frame Interrupt Flag

Indicates the frame interrupt status.

- 1 (R): Interrupt is generated
- 0 (R): Interrupt is not generated (default)
- 1 (W): Flag is reset
- 0 (W): Invalid

INTF is set to 1 when a vertical non-display period begins. If INTEN (D0/0x301A00) is set to 1 to enable the frame interrupt, the interrupt signal is asserted and the LCDC interrupt cause flag in the ITC is set to 1. This flag can only be reset by writing 1 to it.

Note: The LCDC does not support the frame interrupt when the TFT interface is selected (TFTSEL (D31/0x301A60) = 1).

D[30:8] Reserved**D7 VNDPF: Vertical Display Status Flag**

Indicates whether the LCD panel is in a vertical non-display period or not.

- 1 (R): Vertical non-display period (default)
- 0 (R): Vertical display period

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.

Note: When a TFT LCD panel is used with the S1C33E08 TFT LCD interface selected (TFTSEL (D31/0x301A60) = 1), this flag may be fixed at 1 in some rare cases depending on the timing. Refer to Section VIII.1.10, “Precautions,” for how to solve this problem.

D[6:2] Reserved**D[1:0] PSAVE[1:0]: Power Save Mode Select Bits**

Selects power-save mode.

Table VIII.1.9.2 Settings of Power-Save Modes

PSAVE1	PSAVE0	Mode
1	1	Normal operation
1	0	Doze mode
0	1	Reserved
0	0	Power-save mode

(Default: 0b00 = Power-save mode)

The LCD controller is placed in power-save mode by setting PSAVE to 0b00. 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 PSAVE to 0b11.

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.

0x301A10: Horizontal Display Register (pLCDC_HD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Horizontal display register (pLCDC_HD)	00301A10 (W)	D31–23	–	reserved		–	–	0 when being read.
		D22	HTCNT6	Horizontal total period (HT) setup	$HT = (HTCNT + 1) \times 8 [Ts]$	0	R/W	
		D21	HTCNT5	$HT = HDP + HNDP$		0		
		D20	HTCNT4	$HT > HDPS + HDP$ (for HR-TFT)	$HNDP = (HTCNT - HDPCNT) \times 8 [Ts]$	0		
		D19	HTCNT3			0		
		D18	HTCNT2			0		
		D17	HTCNT1			0		
		D16	HTCNT0			0		
		D15–7	–	reserved		–	–	0 when being read.
		D6	HDPCNT6	Horizontal display period (HDP) setup	$HDP = (HDPCNT + 1) \times 8 [Ts]$	0	R/W	
		D5	HDPCNT5			0		
		D4	HDPCNT4			0		
		D3	HDPCNT3			0		
		D2	HDPCNT2			0		
D1	HDPCNT1			0				
D0	HDPCNT0			0				

D[31:23] Reserved**D[22:16] HTCNT[6:0]: Horizontal Total Period (HT) Setup Bits**

Sets the horizontal total period (HT) in 8-pixel increments. (Default: 0x0)

$$HT = (HTCNT[6:0] + 1) \times 8 [Ts] \quad (Ts: \text{pixel clock period})$$

The horizontal total period contains horizontal display period and horizontal non-display period and the maximum value that can be set is 1,024-pixel period.

The following conditions must be satisfied when setting HTCNT[6:0]:

$$HTCNT[6:0] \geq HDPCNT[6:0] + 3$$

$$HT > HDP + HDPS$$

D[15:7] Reserved**D[6:0] HDPCNT[6:0]: Horizontal Display Period (HDP) Setup Bits**

Sets the horizontal display period (HDP, panel horizontal resolution) in 8-pixel increments. (Default: 0x0)

$$HDP = (HDPCNT[6:0] + 1) \times 8 [Ts]$$

The following condition must be satisfied when setting HDPCNT[6:0]:

$$HDP \geq 16 \quad (HDPCNT[6:0] \geq 1)$$

0x301A14: Vertical Display Register (pLCDC_VD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Vertical display register (pLCDC_VD)	00301A14 (W)	D31–26	–	reserved	–	–	–	0 when being read.	
		D25	VTCNT9	Vertical total period (VT) setup	$VT = VTCNT + 1$ [lines]	0	R/W		
		D24	VTCNT8	$VT = VDP + VNDP$		0			
		D23	VTCNT7	$VT > VDPS + VDP$ (for HR-TFT)	$VNDP = HTCNT - HDCNT$	0			
		D22	VTCNT6		[lines]	0			
		D21	VTCNT5			0			
		D20	VTCNT4			0			
		D19	VTCNT3			0			
		D18	VTCNT2			0			
		D17	VTCNT1			0			
		D16	VTCNT0			0			
		D15–10	–	reserved	–	–	–	–	0 when being read.
		D9	VDP CNT9	Vertical display period (VDP) setup	$VDP = VDP CNT + 1$ [lines]	0	R/W		
		D8	VDP CNT8			0			
		D7	VDP CNT7			0			
		D6	VDP CNT6			0			
		D5	VDP CNT5			0			
		D4	VDP CNT4			0			
		D3	VDP CNT3			0			
		D2	VDP CNT2			0			
D1	VDP CNT1			0					
D0	VDP CNT0			0					

D[31:26] Reserved**D[25:16] VTCNT[9:0]: Vertical Total Period (VT) Setup Bits**

Sets the vertical total period (VT) in line units. (Default: 0x0)

$$VT = VTCNT[9:0] + 1 \text{ [lines]}$$

The vertical total period contains vertical display period and vertical non-display period and the maximum value that can be set is 1,024 lines.

The following condition must be satisfied when setting VTCNT[9:0]:

$$VT > VDP + VDPS$$

D[15:10] Reserved**D[9:0] VDP CNT[9:0]: Vertical Display Period (VDP) Setup Bits**

Sets the vertical display period (VDP, panel vertical resolution) in line units. (Default: 0x0)

$$VDP = VDP CNT[9:0] + 1 \text{ [lines]}$$

The following condition must be satisfied when setting VDP CNT[9:0]:

$$VT \geq VDP + 1$$

0x301A18: MOD Rate Register (pLCDC_MR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
MOD rate register (pLCDC_MR)	00301A18 (W)	D31-6	–	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		

D[31:6] Reserved**D[5:0] MOD: LCD MOD Rate Setup Bits**

Sets the cycle time at which to switch the MOD signal. (Default: 0x0)

When this register is 0x0, the MOD signal switches at the cycle time of the FPFAME signal. If another period is desired, set the FPLINE pulse-count value.

0x301A20: Horizontal Display Start Position Register (pLCDC_HDPS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Horizontal display start position register (pLCDC_HDPS)	00301A20 (W)	D31-10	–	reserved		–	–	0 when being read.
		D9	HDPSCNT9	Horizontal display period start position for HR-TFT	$HDPS = HDPSCNT + 1$ [pixels]	0	R/W	0x0 must be set for STN panels.
		D8	HDPSCNT8			0		
		D7	HDPSCNT7	HT > HDPS + HDP		0		
		D6	HDPSCNT6			0		
		D5	HDPSCNT5			0		
		D4	HDPSCNT4			0		
		D3	HDPSCNT3			0		
		D2	HDPSCNT2			0		
		D1	HDPSCNT1			0		
		D0	HDPSCNT0			0		

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:10] Reserved**D[9:0] HDPSCNT[9:0]: Horizontal Display Period Start Position Setup Bits**

Sets the horizontal display period start position (HDPS) for HR-TFT panels in pixel clock units. (Default: 0x0)

$$HDPS = HDPSCNT[9:0] + 1 [Ts] \quad (Ts: \text{pixel clock period})$$

The following condition must be satisfied when setting HDPSCNT[9:0]:
 $HT > HDP + HDPS$

0x301A24: Vertical Display Start Position Register (pLCDC_VDPS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vertical display start position register (pLCDC_VDPS)	00301A24 (W)	D31-10	–	reserved		–	–	0 when being read.
		D9	VDPSCNT9	Vertical display period start position for HR-TFT	VDPS = VDPSCNT [lines] VT > VDPS + VDP	0	R/W	0x0 must be set for STN panels.
		D8	VDPSCNT8			0		
		D7	VDPSCNT7			0		
		D6	VDPSCNT6			0		
		D5	VDPSCNT5			0		
		D4	VDPSCNT4			0		
		D3	VDPSCNT3			0		
		D2	VDPSCNT2			0		
		D1	VDPSCNT1			0		
		D0	VDPSCNT0			0		

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:10] Reserved**D[9:0] VDPSCNT[9:0]: Vertical Display Period Start Position Setup Bits**

Sets the vertical display period start position (VDPS) for HR-TFT panels in line units. (Default: 0x0)

$VDPS = VDPSCNT[9:0] [line]$

The following condition must be satisfied when setting VDPSCNT[9:0]:

$VT > VDP + VDPS$

0x301A28: FPLINE Pulse Setup Register (pLCDC_L)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPLINE pulse setup register (pLCDC_L)	00301A28 (W)	D31-26	–	reserved	–	–	–	0 when being read.
		D25	FPLST9	FPLINE pulse start position	Start position = FPLST + 1 [pixels]	0	R/W	*1: For HR-TFT 0x0 must be set for STN panels.
		D24	FPLST8			0		
		D23	FPLST7			0		
		D22	FPLST6			0		
		D21	FPLST5			0		
		D20	FPLST4			0		
		D19	FPLST3			0		
		D18	FPLST2			0		
		D17	FPLST1			0		
		D16	FPLST0	0				
		D15-8	–	reserved	–	–	–	0 when being read.
		D7	FPLPOL	FPLINE pulse polarity	1 Active high 0 Active low	0	R/W	(*1)
		D6	FPLWD6	FPLINE pulse width	Pulse width = FPLWD + 1 [pixels]	0	R/W	
		D5	FPLWD5			0		
D4	FPLWD4	0						
D3	FPLWD3	0						
D2	FPLWD2	0						
D1	FPLWD1	0						
D0	FPLWD0	0						

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:26] Reserved**D[25:16] FPLST[9:0]: FPLINE Pulse Start Position Setup Bits**

Sets the horizontal sync pulse (FPLINE or LP) start position (HPS) for HR-TFT panels in pixel clock units. (Default: 0x0)

$$\text{HPS} = \text{FPLST}[9:0] + 1 \text{ [Ts]} \quad (\text{Ts: pixel clock period})$$

D[15:8] Reserved**D7 FPLPOL: FPLINE Pulse Polarity Setup Bit**

Sets the horizontal sync pulse polarity for HR-TFT panels.

1 (R/W): Active high

0 (R/W): Active low (default)

D[6:0] FPLWD[6:0]: FPLINE Pulse Width Setup Bits

Sets the horizontal sync pulse width (HPW) for HR-TFT panels in pixel clock units. (Default: 0x0)

$$\text{HPW} = \text{FPLWD}[6:0] + 1 \text{ [Ts]} \quad (\text{Ts: pixel clock period})$$

0x301A2C: FPFFRAME Pulse Setup Register (pLCDC_F)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPFRAME pulse setup register (pLCDC_F)	00301A2C (W)	D31–26	–	reserved	–	–	–	0 when being read.
		D25	FPFST9	FPFRAME pulse start position	Start position = FPFST × HT [pixels]	0	R/W	*1: For HR-TFT 0x0 must be set for STN panels.
		D24	FPFST8			0		
		D23	FPFST7			0		
		D22	FPFST6			0		
		D21	FPFST5			0		
		D20	FPFST4			0		
		D19	FPFST3			0		
		D18	FPFST2			0		
		D17	FPFST1			0		
		D16	FPFST0	0				
		D15–8	–	reserved	–	–	–	0 when being read.
		D7	FPFPOL	FPFRAME pulse polarity	1 Active high 0 Active low	0	R/W	(≠1)
		D6–3	–	reserved	–	–	–	0 when being read.
D2	FPFWD2	FPFRAME pulse width	Pulse width = (FPFWD+1) × HT [pixels]	0	R/W	(*1)		
D1	FPFWD1			0				
D0	FPFWD0			0				

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:26] Reserved

D[25:16] FPFST[9:0]: FPFFRAME Pulse Start Position Setup Bits

Sets the vertical sync pulse (FPFRAME or SPS) start position (VPS) for HR-TFT panels. (Default: 0x0)

$$VPS = FPFST[9:0] \text{ [lines]} = FPFST[9:0] \times HT \text{ [Ts]} \quad (\text{Ts: pixel clock period})$$

D[15:8] Reserved

D7 FFPOL: FPFFRAME Pulse Polarity Setup Bit

Sets the vertical sync pulse polarity for HR-TFT panels.

1 (R/W): Active high

0 (R/W): Active low (default)

D[6:3] Reserved

D[2:0] FPFWD[2:0]: FPFFRAME Pulse Width Setup Bits

Sets the vertical sync pulse width (VPW) for HR-TFT panels. (Default: 0x0)

$$VPW = FPFWD[2:0] + 1 \text{ [lines]} = (FPFWD[2:0] + 1) \times HT \text{ [Ts]} \quad (\text{Ts: pixel clock period})$$

0x301A30: FPFRAmE Pulse Offset Register (pLCDC_FO)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPFRAME pulse offset register (pLCDC_FO)	00301A30 (W)	D31-26	–	reserved	–	–	–	0 when being read.
		D25	FPFSTPO9	FPFRAME pulse stop offset	Stop offset = FPFSTPO [pixels]	0	R/W	*1: For HR-TFT 0x0 must be set for STN panels.
		D24	FPFSTPO8			0		
		D23	FPFSTPO7			0		
		D22	FPFSTPO6			0		
		D21	FPFSTPO5			0		
		D20	FPFSTPO4			0		
		D19	FPFSTPO3			0		
		D18	FPFSTPO2			0		
		D17	FPFSTPO1			0		
		D16	FPFSTPO0			0		
		D15-10	–	reserved	–	–	–	0 when being read.
		D9	FPFSTO9	FPFRAME pulse start offset	Start offset = FPFSTO [pixels]	0	R/W	(*1)
		D8	FPFSTO8			0		
		D7	FPFSTO7			0		
		D6	FPFSTO6			0		
		D5	FPFSTO5			0		
		D4	FPFSTO4			0		
		D3	FPFSTO3			0		
		D2	FPFSTO2			0		
D1	FPFSTO1	0						
D0	FPFSTO0	0						

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:26] Reserved**D[25:16] FPFSTPO[9:0]: FPFRAmE Pulse Stop Offset Bits**

Adjusts the vertical sync pulse end position (pulse width), which has been set in line units, in pixel clock units. (Default: 0x0)

$$VPW' = (FPFWD[2:0] + 1) \times HT - FPFSTO[9:0] + FPFSTPO[9:0] [Ts] \quad (Ts: \text{pixel clock period})$$

D[15:10] Reserved**D[9:0] FPFSTO[9:0]: FPFRAmE Pulse Start Offset Bits**

Adjusts the vertical sync pulse start position, which has been set in line units, in pixel clock units. (Default: 0x0)

$$VPS' = FPFST[9:0] \times HT + FPFSTO[9:0] [Ts] \quad (Ts: \text{pixel clock period})$$

0x301A40: HR-TFT Special Output Register (pLCDC_TSO)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
HR-TFT special output register (pLCDC_TSO)	00301A40 (W)	D31-4	–	reserved	–			–	–	0 when being read.	
		D3	CTL1CTL	TFT_CTL1 control	1	Program	0	Toggle/line	0	R/W	For HR-TFT
		D2	PRESET	TFT_CTL0-2 preset enable	1	Program	0	Preset	0	R/W	0x0 must be set for
		D1	FPSPOL	FPSHIFT polarity	1	Falling	0	Rising	0	R/W	STN panels.
		D0	CTLSWAP	TFT_CTL0/TFT_CTL1 swap	1	Swap	0	Not swap	0	R/W	

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:4] Reserved**D3 CTL1CTL: TFT_CTL1 Control Bit**

Selects the behavior of the TFT_CTL1 (CLS) signal.

1 (R/W): Toggle at the programmed timing

0 (R/W): Toggle every line (default)

Set CTL1CTL to 1 when using the TFT_CTL1 (CLS) signal that has been programmed using the TFT_CTL1 Pulse Register (0x301A44) or preset with standard conditions. CTL1CTL is set to 0 by default, in this case the TFT_CTL1 (CLS) signal toggles between high and low every time the FPLINE (LP) pulse is output.

D2 PRESET: TFT_CTL0-2 Preset Enable Bit

Enables use of the programmed TFT_CTL0 (PS), TFT_CTL1 (CLS), and TFT_CTL2 (REV) signals.

1 (R/W): Programmed signals

0 (R/W): Preset signal (default)

By setting PRESET to 1, the signal timing conditions may be programmed using the registers shown below.

TFT_CTL0 (PS): TFT_CTL0 Pulse Register (0x301A48)

TFT_CTL1 (CLS): TFT_CTL1 Pulse Register (0x301A44)

TFT_CTL2 (REV): TFT_CTL2 Register (0x301A4C)

When PRESET = 0, the TFT_CTL0, TFT_CTL1, and TFT_CTL2 signals are fixed at low.

D1 FPSPOL: FPSHIFT Polarity Select Bit

Selects the polarity of the FPSHIFT (CLK) signal for HR-TFT panels.

1 (R/W): Falling edge

0 (R/W): Rising edge (default)

When FPSPOL is set to 1, the FPDAT[11:0] output signal toggles at the rising edge (sampled at the falling edge) of the FPSHIFT (CLK) signal. When FPSPOL is set to 0, the FPDAT[11:0] output signal toggles at the falling edge (sampled at the rising edge) of the FPSHIFT (CLK) signal.

D0 CTLSWAP: TFT_CTL0/TFT_CTL1 Swap Bit

Swaps the signal between TFT_CTL1 and TFT_CTL0.

1 (R/W): Swapped (TFT_CTL0 = CLS, TFT_CTL1 = PS)

0 (R/W): Not swapped (TFT_CTL0 = PS, TFT_CTL1 = CLS) (default)

0x301A44: TFT_CTL1 Pulse Register (pLCDC_TC1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TFT_CTL1 pulse register (pLCDC_TC1)	00301A44 (W)	D31-26	–	reserved		–	–	0 when being read.
		D25	CTL1STP9	TFT_CTL1 pulse stop offset	Stop offset = CTL1STP + 1 [pixels]	0	R/W	*2: For HR-TFT This register is enabled when PRESET = 1.
		D24	CTL1STP8			0		
		D23	CTL1STP7	TFT_CTL1 pulse width		0		
		D22	CTL1STP6	= (CTL1STP - CTL1ST + 1) Ts		0		
		D21	CTL1STP5			0		
		D20	CTL1STP4			0		
		D19	CTL1STP3			0		
		D18	CTL1STP2			0		
		D17	CTL1STP1			0		
		D16	CTL1STP0			0		
		D15-10	–	reserved		–	–	0 when being read.
		D9	CTL1ST9	TFT_CTL1 pulse start offset	Start offset = CTL1ST [pixels]	0	R/W	(*2)
		D8	CTL1ST8			0		
		D7	CTL1ST7			0		
		D6	CTL1ST6			0		
		D5	CTL1ST5			0		
		D4	CTL1ST4			0		
		D3	CTL1ST3			0		
		D2	CTL1ST2			0		
D1	CTL1ST1			0				
D0	CTL1ST0			0				

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:26] Reserved

D[25:16] CTL1STP[9:0]: TFT_CTL1 Pulse Stop Offset Setup Bits

Specifies the TFT_CTL1 (CLS) pulse end position with an offset value (in pixel clock units) from the FPLINE pulse start position. (Default: 0x0)

D[15:10] Reserved

D[9:0] CTL1ST[9:0]: TFT_CTL1 Pulse Start Offset Setup Bits

Specifies the TFT_CTL1 (CLS) pulse start position with an offset value (in pixel clock units) from the FPLINE pulse start position. (Default: 0x0)

Setting this register configures the TFT_CTL1 pulse width to “CTL1STP[9:0] - CTL1ST[9:0] + 1 [Ts].” To enable this register, set CTL1CTL (D3/0x301A40) and PRESET (D2/0x301A40) to 1.

0x301A48: TFT_CTL0 Pulse Register (pLCDC_TC0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TFT_CTL0 pulse register (pLCDC_TC0)	00301A48 (W)	D31–26	–	reserved		–	–	0 when being read.
		D25	CTL0STP9	TFT_CTL0 pulse stop offset	Stop offset = CTL0STP + 1 [pixels]	0	R/W	*2: For HR-TFT This register is enabled when PRESET = 1.
		D24	CTL0STP8			0		
		D23	CTL0STP7	TFT_CTL0 pulse width		0		
		D22	CTL0STP6	= (CTL0STP - CTL0ST + 1) Ts		0		
		D21	CTL0STP5			0		
		D20	CTL0STP4			0		
		D19	CTL0STP3			0		
		D18	CTL0STP2			0		
		D17	CTL0STP1			0		
		D16	CTL0STP0			0		
		D15–10	–	reserved		–	–	0 when being read.
		D9	CTL0ST9	TFT_CTL0 pulse start offset	Start offset = CTL0ST [pixels]	0	R/W	(*2)
		D8	CTL0ST8			0		
		D7	CTL0ST7			0		
		D6	CTL0ST6			0		
		D5	CTL0ST5			0		
		D4	CTL0ST4			0		
		D3	CTL0ST3			0		
		D2	CTL0ST2			0		
D1	CTL0ST1			0				
D0	CTL0ST0			0				

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:26] Reserved

D[25:16] CTL0STP[9:0]: TFT_CTL0 Pulse Stop Offset Setup Bits

Specifies the TFT_CTL0 (PS) pulse end position with an offset value (in pixel clock units) from the FPLINE pulse start position. (Default: 0x0)

D[15:10] Reserved

D[9:0] CTL0ST[9:0]: TFT_CTL0 Pulse Start Offset Setup Bits

Specifies the TFT_CTL0 (PS) pulse start position with an offset value (in pixel clock units) from the FPLINE pulse start position. (Default: 0x0)

Setting this register configures the TFT_CTL0 pulse width to “CTL0STP[9:0] - CTL0ST[9:0] + 1 [Ts].”
To enable this register, set PRESET (D2/0x301A40) to 1.

0x301A4C: TFT_CTL2 Register (pLCDC_TC2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TFT_CTL2 register (pLCDC_TC2)	00301A4C (W)	D31-10	–	reserved		–	–	0 when being read.
		D9	CTL2DLY9	TFT_CTL2 delay	Delay = CTL2DLY [pixels]	0	R/W	For HR-TFT This register is enabled when PRESET = 1.
		D8	CTL2DLY8			0		
		D7	CTL2DLY7			0		
		D6	CTL2DLY6			0		
		D5	CTL2DLY5			0		
		D4	CTL2DLY4			0		
		D3	CTL2DLY3			0		
		D2	CTL2DLY2			0		
		D1	CTL2DLY1			0		
		D0	CTL2DLY0			0		

Note: This register is used only for setting HR-TFT panel parameters. When using an STN panel, leave this register unaltered as 0x0.

D[31:10] Reserved**D[9:0] CTL2DLY[9:0]: TFT_CTL2 Delay Setup Bits**

Sets the delay time (in pixel clock units) from the FPLINE pulse start position until the TFT_CTL2 signal toggles. (Default: 0x0)

To enable this register, set PRESET (D2/0x301A40) to 1.

0x301A60: LCDC Display Mode Register (pLCDC_DMD)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks		
LCDC display mode register (pLCDC_DMD)	00301A60 (W)	D31	TFTSEL	HR-TFT panel selection	1	HR-TFT	0	STN	0	R/W		
		D30	COLOR	Color/mono selection	1	Color	0	Mono	0	R/W		
		D29	FPSMASK	FPSHIFT mask enable	1	Enabled	0	Disabled	0	R/W		
		D28	–	reserved					–	–		0 when being read.
		D27	DWD1 DWD0	LCD panel data width		1	DWD[1:0]	Data format		0		R/W
		11					8-bit (format2)		0			
		10					reserved					
		01					8-bit (format1)					
		D25	SWINV	Software video invert	1	Inverted	0	Normal	0	R/W		
		D24	BLANK	Display blank enable	1	Blank	0	Normal	0	R/W		
		D23–8	–	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.
D2–0	BPP2 BPP1 BPP0	Bit-per-pixel select		1	BPP[2:0]	bpp (color/gray)		0	R/W			
101					16 bpp (64Kc)		0					
100					12 bpp (4Kc)		0					
011					8 bpp (256c)							
010					4 bpp (16c/16gr)							
001					2 bpp (4c/4gr)							
000					1 bpp (2c/2gr)							
Other	reserved											

D31 TFTSEL: HR-TFT Panel Select Bit

Selects the type of connected LCD panel (STN or HR-TFT).

1 (R/W): Generic HR-TFT panel

0 (R/W): STN panel (default)

When Generic HR-TFT panel is selected, COLOR (D30) and DWD[1:0] (D[27:26]) settings are disabled.

D30 COLOR: Color/Mono Panel Select Bit

Selects the type of connected LCD panel (color or monochrome).

1 (R/W): Color panel

0 (R/W): Monochrome panel (default)

D29 FPSMASK: FPSHIFT Mask Enable Bit

Enables the FPSHIFT mask (effective only for STN monochrome LCD panels and HR-TFT panels).

1 (R/W): Enable

0 (R/W): Disable (default)

When FPSMASK is set to 1, the FPSHIFT signal is masked and is not output during the non-display period. When FPSMASK is set to 0, the FPSHIFT signal is output even during the non-display period. This setting is effective only for STN monochrome LCD panels (COLOR = 0) and HR-TFT panels. When an STN color LCD panel is used, the FPSHIFT signal is always masked regardless of the setting of this bit.

D28 Reserved

D[27:26] DWD[1:0]: LCD Panel Data Width Select Bits

Selects the LCD panel's data width and format.

Table VIII.1.9.3 Selection of LCD Panels

TFTSEL	COLOR	DWD1	DWD0	LCD panel
1	–	–	–	12-bit Generic HR-TFT LCD
0	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

D25 SWINV: Software Video Invert Bit

Inverts the display.

1 (R/W): Invert

0 (R/W): Normal display (default)

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. Inverse operation is applied to output of the look-up tables, and does not affect the display memory.

The software video invert function is effective only for STN panels and it does not support HR-TFT panels.

D24 BLANK: Display Blank Enable Bit

Clears the display (entire screen turned blank).

1 (R/W): Blank

0 (R/W): Normal display (default)

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 (when SWINV = 0) or high (when SWINV = 1) to clear the display. This setting does not affect the display memory.

This function is effective for both STN and HR-TFT panels.

D[23:8] Reserved**D7 FRMRPT: Frame Repeat for EL Panel Bit**

Selects whether to repeat the frame-rate modulation pattern (effective only for EL panels).

1 (R/W): Repeated

0 (R/W): Not repeated (default)

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.

D6 DITHEN: Dither Mode Enable Bit

Enables or disables dither mode.

1 (R/W): Dither mode

0 (R/W): Normal mode (default)

When DITHEN is set to 1, a maximum of 64K colors in 16 bpp mode will be generated. Setting DITHEN to 0 allows use of a maximum of 4K colors.

D5 Reserved

D4 LUTPASS: LUT Bypass Mode Select Bit

Selects whether the look-up table is bypassed.

1 (R/W): Bypassed

0 (R/W): Used (default)

When LUTPASS is set to 1, 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.

In 1-, 2-, 4-, and 16-bpp color mode, this bit must be set to 1 as the look-up table is not used.

D3 Reserved**D[2:0] BPP[2:0]: Bit-Per-Pixel Select Bits**

Selects display mode (bpp mode). The contents of selection, including that of COLOR (D30), are listed in Table VIII.1.9.4.

Table VIII.1.9.4 Specification of Display Modes

BPP2	BPP1	BPP0	Display mode		
			Monochrome (COLOR = 0)	Color (COLOR = 1), STN	Color (COLOR = 1), TFT
1	1	*	Reserved	Reserved	Reserved
1	0	1	Reserved	16 bpp, 64K colors	Reserved
1	0	0	Reserved	12 bpp, 4K colors	12 bpp, 4K colors
0	1	1	Reserved	8 bpp, 256 colors	8 bpp, 256 colors
0	1	0	4 bpp, 16 gray levels	Reserved	4 bpp, 16 colors
0	0	1	2 bpp, 4 gray levels	Reserved	2 bpp, 4 colors
0	0	0	1 bpp, 2 gray levels	Reserved	1 bpp, 2 colors

(Default: 0b000)

0x301A64: IRAM Select Register (pLCDC_IRAM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
IRAM select register (pLCDC_IRAM)	00301A64 (W)	D31-1	-	reserved	-		-	-	0 when being read.
		D0	IRAM	IRAM assignment	1 A0RAM	0 IVRAM	0	R/W	

D[31:1] Reserved**D0 IRAM: IRAM Assignment Bit**

Selects whether the 12K-byte IVRAM is used as a VRAM or a general-purpose RAM.

1 (R/W): A0RAM

0 (R/W): IVRAM (default)

The IVRAM is located at 0x80000 to 0x82FFF in area 3 by default and it can be used as a VRAM. When the IVRAM is not used as a VRAM, it can be located in area 0 as a general-purpose RAM by setting IRAM to 1.

0x301A70: Main Window Display Start Address Register (pLCDC_MADD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Main window display start address register (pLCDC_MADD)	00301A70 (W)	D31	MWADR31	Main window start address MWADR31 = MSB MWADR0 = LSB	0x0 to 0xFFFFFFFFC	0x0	R/W	
		D30	MWADR30					
		D1	MWADR1					
		D0	MWADR0					

D[31:0] MWADR[31:0]: Main Window Start Address

Sets the main window display start address. (Default: 0x0)

Note that a word boundary address ($A[1:0] = 0b00$) in the IVRAM or external SDRAM must be specified to this register.

0x301A74: Main Window Line Address Offset Register (pLCDC_MLADD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Main window line address offset register (pLCDC_MLADD)	00301A74 (W)	D31-10	--	reserved		--	--	0 when being read.
		D9	MWLADR9	Main window line address offset	Main window width (pixels) × BPP/32	0	R/W	
		D8	MWLADR8			0		
		D7	MWLADR7			0		
		D6	MWLADR6			0		
		D5	MWLADR5			0		
		D4	MWLADR4			0		
		D3	MWLADR3			0		
		D2	MWLADR2			0		
		D1	MWLADR1			0		
		D0	MWLADR0			0		

D[31:10] Reserved**D[9:0] MWLADR[9:0]: Main Window Line Address Offset Bits**

Sets the line address offset for displaying the main window. (Default: 0x0)

The S1C33E08 LCDC manipulates display data in units of words. Therefore, the image width (number of pixels) must be a multiple of (32 bits ÷ bpp). The line address offset is the number of words corresponding to the image width and it should be specified in the following cases:

1. When the Picture-in-Picture Plus function is used (when a sub-window is displayed)
2. When the LCD panel horizontal resolution is not a multiple of (32 bits ÷ bpp)

The line address offset is calculated as follows:

$$\text{Main Window Line Address Offset} = \text{Main window width in pixels} \times \text{bpp} / 32$$

For example, to realize the virtual screen with a 640-pixel width, set MWLADR[9:0] as follows:

Example 1) MWLADR[9:0] = $640 \times 8 / 32 = 160$ [words] (in 8-bpp mode)

Example 2) MWLADR[9:0] = $640 \times 1 / 32 = 20$ [words] (in 1-bpp mode)

If the calculated value has a decimal fraction it must be rounded up.

For example, if the LCD width and image width are 240 pixels in 1-bpp mode,

Example 3) MWLADR[9:0] = $240 \times 1 / 32 = 7.5$ [words]

In this case, MWLADR[9:0] must be set to 8. Furthermore, the image must be prepared in 256 (8 × 32) pixels wide.

For details, see Section VIII.1.6.2, "Setting the Display Start Address and Line Address Offset."

0x301A80: Sub-window Display Start Address Register (pLCDC_SADD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Sub-window display start address register (pLCDC_SADD)	00301A80 (W)	D31	SWADR31	Sub-window start address SWADR31 = MSB SWADR0 = LSB	0x0 to 0xFFFFFFFFC	0x0	R/W	
		D30	SWADR30					
		D1	SWADR1					
		D0	SWADR0					

D[31:0] SWADR[31:0]: Sub-window Start Address

Sets the sub-window display start address. (Default: 0x0)

Note that a word boundary address ($A[1:0] = 0b00$) in the IVRAM or external SDRAM must be specified to this register.

0x301A88: Sub-window Start Position Register (pLCDC_SSP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Sub-window start position register (pLCDC_SSP)	00301A88 (W)	D31	PIPEN	PIP sub-window enable	1 Enabled 0 Disabled	0	R/W	
		D30–26	–	reserved	–	–	–	0 when being read.
		D25	PIPYST9	Sub-window vertical (Y) start position	Y start position = PIPYST (lines) from the origin	0	R/W	*3: This register is enabled when PIPEN = 1.
		D24	PIPYST8			0		
		D23	PIPYST7			0		
		D22	PIPYST6			0		
		D21	PIPYST5			0		
		D20	PIPYST4			0		
		D19	PIPYST3			0		
		D18	PIPYST2			0		
		D17	PIPYST1			0		
		D16	PIPYST0	0				
		D15–10	–	reserved	–	–	–	0 when being read.
		D9	PIPXST9	Sub-window horizontal (X) start position	X start position = PIPXST (pixels) from the origin (word units)	0	R/W	(*3)
		D8	PIPXST8			0		
		D7	PIPXST7			0		
		D6	PIPXST6			0		
		D5	PIPXST5			0		
		D4	PIPXST4			0		
		D3	PIPXST3			0		
D2	PIPXST2	0						
D1	PIPXST1	0						
D0	PIPXST0	0						

D31 PIPEN: PIP Sub-window Enable Bit

Enables the Picture-in-Picture Plus function to display the sub-window in the main window.

1 (R/W): Enable

0 (R/W): Disable (default)

Configure the sub-window using the registers at 0x301A80 to 0x301A8C before setting PIPEN to 1.

D[30:26] Reserved**D[25:16] PIPYST[9:0]: Sub-window Vertical (Y) Start Position Bits**

Sets the sub-window vertical display start position. (Default: 0x0)

Specify the number of lines from the LCD panel origin point to the upper left corner of the sub-window in 1-line increments.

$PIPYST[9:0] = Y_{START}$ [lines]

For example, to specify the sub-window vertical start position as 60 lines, set PIPYST[9:0] to 60.

D[15:10] Reserved**D[9:0] PIPXST[9:0]: Sub-window Horizontal (X) Start Position Bits**

Sets the sub-window horizontal display start position. (Default: 0x0)

Convert the number of pixels from the LCD panel origin point to the upper left corner of the sub-window into the number of data words according to the bpp mode and set it to these bits.

$PIPXST[9:0] = X_{START} \text{ pixels} \times \text{bpp} \div 32$ [words]

It can be specified in $(32 \text{ bits} \div \text{bpp})$ pixel increments.

1-bpp mode: 1-word = 32-pixel units

2-bpp mode: 1-word = 16-pixel units

4-bpp mode: 1-word = 8-pixel units

8-bpp mode: 1-word = 4-pixel units

12-bpp mode: 3-word = 8-pixel units (because the value must be an integer)

16-bpp mode: 1-word = 2-pixel units

For example, to specify the sub-window horizontal start position as 80 pixels in 8-bpp mode, set PIPXST[9:0] to 20.

0x301A8C: Sub-window End Position Register (pLCDC_SEP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Sub-window end position register (pLCDC_SEP)	00301A8C (W)	D31-26	–	reserved	–	–	–	0 when being read.
		D25	PIPYEND9	Sub-window vertical (Y) end position	Y end position = PIPYEND (lines) from the origin	0	R/W	*3: This register is enabled when PIPEN = 1.
		D24	PIPYEND8			0		
		D23	PIPYEND7			0		
		D22	PIPYEND6			0		
		D21	PIPYEND5			0		
		D20	PIPYEND4			0		
		D19	PIPYEND3			0		
		D18	PIPYEND2			0		
		D17	PIPYEND1			0		
		D16	PIPYEND0			0		
		D15-10	–	reserved	–	–	–	0 when being read.
		D9	PIPXEND9	Sub-window horizontal (X) end position	X end position = PIPXEND (pixels) from the origin (word units)	0	R/W	(*3)
		D8	PIPXEND8			0		
		D7	PIPXEND7			0		
		D6	PIPXEND6			0		
		D5	PIPXEND5			0		
		D4	PIPXEND4			0		
		D3	PIPXEND3			0		
		D2	PIPXEND2			0		
D1	PIPXEND1	0						
D0	PIPXEND0	0						

D[31:26] Reserved**D[25:16] PIPYEND[9:0]: Sub-window Vertical (Y) End Position Bits**

Sets the sub-window vertical display end position. (Default: 0x0)

Specify the number of lines from the LCD panel origin point to the lower right corner of the sub-window in 1-line increments.

$$\text{PIPYEND}[9:0] = Y_{\text{END}} - 1 \text{ [lines]}$$

D[15:10] Reserved**D[9:0] PIPXEND[9:0]: Sub-window Horizontal (X) End Position Bits**

Sets the sub-window horizontal display end position. (Default: 0x0)

Convert the number of pixels from the LCD panel origin point to the lower right corner of the sub-window into the number of data words according to the bpp mode and set it to these bits.

$$\text{PIPXEND}[9:0] = X_{\text{END}} \text{ pixels} \times \text{bpp} \div 32 - 1 \text{ [words]}$$

0x301AA0: Look-up Table Data Register 0 (pLCDC_LUT_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Look-up table data register 0 (pLCDC_LUT_03)	00301AA0 (W)	D31	LUT35	Look-up table entry 3 data	0x0 to 0x3F	0	R/W	
		D30	LUT34					
		D29	LUT33					
		D28	LUT32					
		D27	LUT31					
		D26	LUT30					
		D25-24	--	reserved	--	--	0 when being read.	
		D23	LUT25	Look-up table entry 2 data	0x0 to 0x3F	0	R/W	
		D22	LUT24					
		D21	LUT23					
		D20	LUT22					
		D19	LUT21					
		D18	LUT20					
		D17-16	--	reserved	--	--	0 when being read.	
		D15	LUT15	Look-up table entry 1 data	0x0 to 0x3F	0	R/W	
		D14	LUT14					
		D13	LUT13					
		D12	LUT12					
		D11	LUT11					
		D10	LUT10					
		D9-8	--	reserved	--	--	0 when being read.	
		D7	LUT05	Look-up table entry 0 data	0x0 to 0x3F	0	R/W	
		D6	LUT04					
		D5	LUT03					
		D4	LUT02					
		D3	LUT01					
		D2	LUT00					
		D1-0	--	reserved	--	--	0 when being read.	

This register is used to set data to the look-up table entries 3 to 0.

Data written to this register is set to the entries 3 to 0 in the red, green, and blue look-up tables simultaneously. Also this register allows reading of the currently set look-up table data.

D[31:26] LUT3[5:0]: Look-up Table Entry 3 Data

Set the 6-bit data for the look-up table entry 3. (Default: 0x0)

D[25:24] Reserved**D[23:18] LUT2[5:0]: Look-up Table Entry 2 Data**

Set the 6-bit data for the look-up table entry 2. (Default: 0x0)

D[17:16] Reserved**D[15:10] LUT1[5:0]: Look-up Table Entry 1 Data**

Set the 6-bit data for the look-up table entry 1. (Default: 0x0)

D[9:8] Reserved**D[7:2] LUT0[5:0]: Look-up Table Entry 0 Data**

Set the 6-bit data for the look-up table entry 0. (Default: 0x0)

D[1:0] Reserved

0x301AA4: Look-up Table Data Register 1 (pLCDC_LUT_47)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Look-up table data register 1 (pLCDC_LUT_47)	00301AA4 (W)	D31	LUT75	Look-up table entry 7 data	0x0 to 0x3F	0	R/W	
		D30	LUT74			0		
		D29	LUT73			0		
		D28	LUT72			0		
		D27	LUT71			0		
		D26	LUT70			0		
		D25–24	–	reserved	–	–	0 when being read.	
		D23	LUT65	Look-up table entry 6 data	0x0 to 0x3F	0	R/W	
		D22	LUT64			0		
		D21	LUT63			0		
		D20	LUT62			0		
		D19	LUT61			0		
		D18	LUT60			0		
		D17–16	–	reserved	–	–	0 when being read.	
		D15	LUT55	Look-up table entry 5 data	0x0 to 0x3F	0	R/W	
		D14	LUT54			0		
		D13	LUT53			0		
		D12	LUT52			0		
		D11	LUT51			0		
		D10	LUT50			0		
		D9–8	–	reserved	–	–	0 when being read.	
		D7	LUT45	Look-up table entry 4 data	0x0 to 0x3F	0	R/W	
		D6	LUT44			0		
		D5	LUT43			0		
		D4	LUT42			0		
		D3	LUT41			0		
		D2	LUT40			0		
		D1–0	–	reserved	–	–	0 when being read.	

This register is used to set data to the look-up table entries 7 to 4.

Data written to this register is set to the entries 7 to 4 in the red, green, and blue look-up tables simultaneously. Also this register allows reading of the currently set look-up table data.

D[31:26] LUT7[5:0]: Look-up Table Entry 7 Data

Set the 6-bit data for the look-up table entry 7. (Default: 0x0)

D[25:24] Reserved**D[23:18] LUT6[5:0]: Look-up Table Entry 6 Data**

Set the 6-bit data for the look-up table entry 6. (Default: 0x0)

D[17:16] Reserved**D[15:10] LUT5[5:0]: Look-up Table Entry 5 Data**

Set the 6-bit data for the look-up table entry 5. (Default: 0x0)

D[9:8] Reserved**D[7:2] LUT4[5:0]: Look-up Table Entry 4 Data**

Set the 6-bit data for the look-up table entry 4. (Default: 0x0)

D[1:0] Reserved

0x301AA8: Look-up Table Data Register 2 (pLCDC_LUT_8B)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Look-up table data register 2 (pLCDC_LUT_8B)	00301AA8 (W)	D31	LUTB5	Look-up table entry 11 data	0x0 to 0x3F	0	R/W	
		D30	LUTB4					
		D29	LUTB3					
		D28	LUTB2					
		D27	LUTB1					
		D26	LUTB0					
		D25-24	--	reserved	--	--	0 when being read.	
		D23	LUTA5	Look-up table entry 10 data	0x0 to 0x3F	0	R/W	
		D22	LUTA4					
		D21	LUTA3					
		D20	LUTA2					
		D19	LUTA1					
		D18	LUTA0					
		D17-16	--	reserved	--	--	0 when being read.	
		D15	LUT95	Look-up table entry 9 data	0x0 to 0x3F	0	R/W	
		D14	LUT94					
		D13	LUT93					
		D12	LUT92					
		D11	LUT91					
		D10	LUT90					
		D9-8	--	reserved	--	--	0 when being read.	
		D7	LUT85	Look-up table entry 8 data	0x0 to 0x3F	0	R/W	
		D6	LUT84					
		D5	LUT83					
		D4	LUT82					
		D3	LUT81					
		D2	LUT80					
		D1-0	--	reserved	--	--	0 when being read.	

This register is used to set data to the look-up table entries 11 to 8.

Data written to this register is set to the entries 11 to 8 in the red, green, and blue look-up tables simultaneously. Also this register allows reading of the currently set look-up table data.

D[31:26] LUTB[5:0]: Look-up Table Entry 11 Data

Set the 6-bit data for the look-up table entry 11. (Default: 0x0)

D[25:24] Reserved**D[23:18] LUTA[5:0]: Look-up Table Entry 10 Data**

Set the 6-bit data for the look-up table entry 10. (Default: 0x0)

D[17:16] Reserved**D[15:10] LUT9[5:0]: Look-up Table Entry 9 Data**

Set the 6-bit data for the look-up table entry 9. (Default: 0x0)

D[9:8] Reserved**D[7:2] LUT8[5:0]: Look-up Table Entry 8 Data**

Set the 6-bit data for the look-up table entry 8. (Default: 0x0)

D[1:0] Reserved

0x301AAC: Look-up Table Data Register 3 (pLCDC_LUT_CF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Look-up table data register 3 (pLCDC_LUT_CF)	00301AAC (W)	D31	LUTF5	Look-up table entry 15 data	0x0 to 0x3F	0	R/W	
		D30	LUTF4					
		D29	LUTF3					
		D28	LUTF2					
		D27	LUTF1					
		D26	LUTF0					
		D25–24	–	reserved	–	–	–	0 when being read.
		D23	LUTE5	Look-up table entry 14 data	0x0 to 0x3F	0	R/W	
		D22	LUTE4					
		D21	LUTE3					
		D20	LUTE2					
		D19	LUTE1					
		D18	LUTE0					
		D17–16	–	reserved	–	–	–	0 when being read.
		D15	LUTD5	Look-up table entry 13 data	0x0 to 0x3F	0	R/W	
		D14	LUTD4					
		D13	LUTD3					
		D12	LUTD2					
		D11	LUTD1					
		D10	LUTD0					
		D9–8	–	reserved	–	–	–	0 when being read.
		D7	LUTC5	Look-up table entry 12 data	0x0 to 0x3F	0	R/W	
		D6	LUTC4					
		D5	LUTC3					
		D4	LUTC2					
		D3	LUTC1					
		D2	LUTC0					
		D1–0	–	reserved	–	–	–	0 when being read.

This register is used to set data to the look-up table entries 15 to 12.

Data written to this register is set to the entries 15 to 12 in the red, green, and blue look-up tables simultaneously. Also this register allows reading of the currently set look-up table data.

D[31:26] LUTF[5:0]: Look-up Table Entry 15 Data

Set the 6-bit data for the look-up table entry 15. (Default: 0x0)

D[25:24] Reserved**D[23:18] LUTE[5:0]: Look-up Table Entry 14 Data**

Set the 6-bit data for the look-up table entry 14. (Default: 0x0)

D[17:16] Reserved**D[15:10] LUTD[5:0]: Look-up Table Entry 13 Data**

Set the 6-bit data for the look-up table entry 13. (Default: 0x0)

D[9:8] Reserved**D[7:2] LUTC[5:0]: Look-up Table Entry 12 Data**

Set the 6-bit data for the look-up table entry 12. (Default: 0x0)

D[1:0] Reserved

VIII.1.10 Precautions

- The LCDC clock supply cannot be stopped while the LCD displays a screen. Before the LCDC clock supply can be stopped, the LCDC must enter power save mode.
- When using an STN panel, the registers for setting the HR-TFT timing parameters must be set to 0x0.
- Display addresses and positions are specified with a word boundary address or in word units, therefore the Main Window Line Address Offset Register (D[9:0]/0x301A74) must be set to a multiple of (32 bits ÷ bpp). Depending on the LCD horizontal resolution and the bpp mode selected, it may be necessary to reserve a larger image area than the LCD panel resolution and set the appropriate line address offset even if the application does not need a larger image than the LCD panel to be displayed.
For example, if the LCD width and image width are 240 pixels in 1-bpp mode,

Line address offset = $240 \times 1 / 32 = 7.5$ [words]

In this case, MWLADR[9:0] (D[9:0]/0x301A74) must be set to 8. Furthermore, the image must be prepared in 256 (8 × 32) pixels wide.

* **MWLADR[9:0]**: Main Window Line Address Offset Bits in the Main Window Line Address Offset Register (D[9:0]/0x301A74)

- When a TFT LCD panel is used with the S1C33E08 TFT LCD interface selected (TFTSEL (D31/0x301A60) = 1), the VNDPF flag (D7/0x301A04) may be fixed at 1 (vertical non-display period) in some rare cases depending on the timing. This means that the flag cannot normally indicate the vertical display and vertical non-display periods in the TFT interface.

* **TFTSEL**: HR-TFT Panel Select Bit in the LCDC Display Mode Register (D31/0x301A60)

* **VNDPF**: Vertical Display Status Flag in the Status and Power Save Configuration Register (D7/0x301A04)

How to solve this problem

For example, when the application needs to know a vertical non-display period to perform the following processing (switching the system clock in a vertical non-display period):

- (1) The system clock is 60 MHz (using the PLL) and the TFT LCD is displaying.
- (2) The program detects a vertical non-display period and switches the system clock to 12 MHz while keeping the TFT LCD displayed.
- (3) The program sets the CPU to enter HALT mode.

To realize the above processing, use the following procedure to detect a vertical non-display period:

- (1) The system clock is 60 MHz (using the PLL) and the TFT LCD is displaying.
- (2) Set the Clock Control Register of a 16-bit timer.
- (3) Set the Comparison Data B Setup Register of the 16-bit timer.
- (4) Clear the 16-bit timer compare-match B interrupt flag.
- (5) Detect the FPFFRAME signal by reading the data register of the I/O port that has been configured for the FPFFRAME output and run the 16-bit timer when the FPFFRAME signal is asserted (high level is detected when FPFFRAME is configured to high active, or low level is detected when it is configured to low active).
- (6) Detect that the 16-bit timer compare-match B interrupt flag is set (it means that a vertical non-display period is detected).
- (7) Switch the system clock to 12 MHz.
- (8) Enter HALT mode.

Use the following equation to calculate the comparison data B to be set to the 16-bit timer:

$$\text{Comparison Data B} = \frac{1}{f_{\text{pshift}}} \times ((\text{VDPS} + \text{VDP}) \times \text{HT} - \text{VPS}') \times f_{\text{t16}}$$

f_{pshift} : LCDC FPSHIFT clock frequency [Hz]

VDPS: Vertical display period start position (VDPS = VDPCNT[9:0] [lines])

VDP: Vertical display period (VDP = VDPCNT[9:0] + 1 [lines])

HT: Horizontal total period (HT = (HTCNT[6:0] + 1) × 8 [Ts])

VPS': Vertical sync pulse start position (VPS' = FPFST[9:0] × HT + FPFSTO[9:0] [Ts])

f_{t16} : 16-bit timer x clock frequency [Hz]

VDPCNT[9:0]: D[9:0]/0x301A24

VDPCNT[9:0]: D[9:0]/0x301A14

HTCNT[6:0]: D[22:16]/0x301A10

FPFST[9:0]: D[25:16]/0x301A2C

FPFSTO[9:0]: D[9:0]/0x301A30

VIII PERIPHERAL MODULES 6 (LCD): LCD CONTROLLER (LCDC)

- The LCDC does not support the frame interrupt (INTEN (D0/0x301A00), INTF (D31/0x301A04)) when TFT interface is selected (TFTSEL (D31/0x301A60) = 1).
 - * **INTF**: Frame Interrupt Flag in the Status and Power Save Configuration Register (D31/0x301A04)
 - * **INTEN**: Frame Interrupt Enable Bit in the Frame Interrupt Register (D0/0x301A00)

VIII.2 IVRAM and IVRAM Arbiter

VIII.2.1 IVRAM

The S1C33E08 has a built-in 12K-byte RAM (IVRAM) to be used as a video RAM for the LCDC, which is located at 0x80000 to 0x82FFF in area 3 by default. When the LCDC is not used or when an external SDRAM is used as a video RAM, IVRAM can be moved to area 0 to use as a general-purpose high-speed RAM. The LCDC provides the control bit IRAM (D0/0x301A64) for this switch over. Setting IRAM (D0/0x301A64) to 1 configures the RAM as a general-purpose RAM in area 0; setting to 0 configures it as a VRAM in area 3.

* **IRAM**: IRAM Assignment Bit in the IRAM Select Register (D0/0x301A64)

Table VIII.2.1.1 lists the differences in the IVRAM function depending on the allocated areas.

Table VIII.2.1.1 IVRAM Function in Different Areas

Item	IRAM = 0 (IVRAM)	IRAM = 1 (general-purpose RAM)
Location	Area 3, 0x80000–0x82FFF	Area 0, 0x2000–0x4FFF*
Accessing from the LCDC	Enabled (via IVRAM arbiter) 32-bit read only	Disabled
Accessing from the CPU	Enabled (via IVRAM arbiter) 8, 16, 32-bit read/write	Enabled (direct) 8, 16, 32-bit read/write
Accessing by DMA	Enabled (via IVRAM arbiter)	Disabled
Use for VRAM	Enabled	Disabled
Program execution	Disabled	Enabled
Storing general-purpose data	Enabled	Enabled
Wait cycle	2 cycles min.	0 cycles min.

* Can be used as a 20KB RAM with A0RAM

When using IVRAM as a display memory, the memory size required for a screen depends on the screen size and bpp mode (shades of gray/colors). It can be expressed by the following equation:

$$\text{Screen data size} = H_PIXEL \times V_PIXEL \times \text{bpp} / 32 \text{ [words]}$$

(The fractional portion of the number must be rounded up.)

H_PIXEL: Number of horizontal pixels

V_PIXEL: Number of vertical pixels

bpp: Number of bits per pixel (1, 2, 4, 8, 12, 16)

Table VIII.2.1.2 IVRAM Usage Samples

Panel size	Color depth	Data size	Used IVRAM range*1
320 × 240	1 bpp	2400 words (9600 bytes)	0x80000–0x8257F
240 × 180	1 bpp	1440 words (5760 bytes)*2	0x80000–0x8167F
	2 bpp	2700 words (10800 bytes)	0x80000–0x82A2F
160 × 160	1 bpp	800 words (3200 bytes)	0x80000–0x80C7F
	2 bpp	1600 words (6400 bytes)	0x80000–0x818FF

*1 These examples assume that IVRAM is used from the beginning. The display start address can be changed using an LCDC register, note, however, that a word boundary address must be specified.

*2 The image width must be set in word units (= multiple of 32/bpp pixels). A 240-pixel of image width in 1-bpp mode cannot be specified in word units. In this case, the image area must be allocated to a 256-pixel width (the LCD can display images with a 240-pixel width).

VIII.2.2 IVRAM Arbiter

The IVRAM arbiter circuit arbitrates between the VRAM data read request from the LCDC and the access request from the CPU.

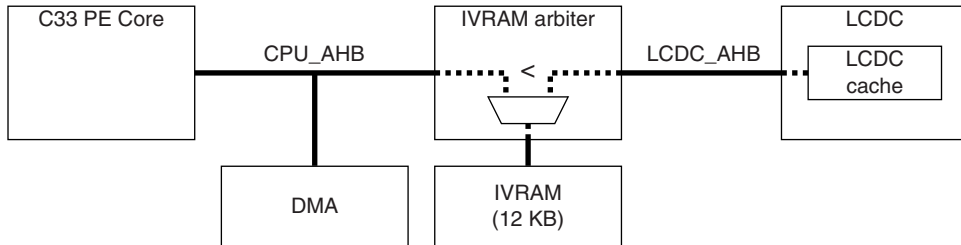


Figure VIII.2.2.1 IVRAM Arbiter

When the CPU (or DMA) and LCDC request to access IVRAM at the same time or the CPU (or DMA) requests to access IVRAM 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.

VIII.2.3 IVRAM Arbiter Operating Clock

The IVRAM arbiter use the IVRAM_ARB_CLK clock (= MCLK) generated by the CMU as the operating clock.

Controlling the supply of the operating clock

IVRAM_ARB_CLK is supplied to the IVRAM arbiter with default settings. It can be turned off using IVRAMARB_CKE (D19/0x301B04) to reduce the amount of power consumed on the chip if the IVRAM arbiter is not used.

* **IVRAMARB_CKE**: IVRAM Arbiter Clock Control Bit in the Gated Clock Control Register 1 (D19/0x301B04)

Setting IVRAMARB_CKE (D19/0x301B04) to 0 (1 by default) turns off the clock supply to the IVRAM arbiter. For details on how to set and control the clock, refer to Section III.1, “Clock Management Unit (CMU).”

Note: The Gated Clock Control Register 1 (0x301B04) is write-protected. Write protection of this and other CMU control registers at addresses 0x301B00 to 0x301B14 to be rewritten must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 could cause the system to operate erratically, make sure the data set in the Clock Control Protect Register (0x301B24) is other than 0x96, unless re-writing said registers.

Clock state in standby mode

The clock supply to the IVRAM arbiter stops depending on type of standby mode.

HALT mode: The operating clock is supplied the same way as in normal mode.

SLEEP mode: The operating clock supply stops.

Therefore, the IVRAM arbiter also stops operating in SLEEP mode.

VIII.2.4 Details of the Control Register

The following describes the IRAM Select Register.

The IRAM Select Register is mapped in the 32-bit device area, and can be accessed only in units of words.

Note: When setting the IRAM Select Register, be sure to write a 0, and not a 1, for all “reserved bits.”

0x301A64: IRAM Select Register (pLCDC_IRAM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
IRAM select register (pLCDC_IRAM)	00301A64 (W)	D31-1	–	reserved	–		–	–	0 when being read.
		D0	IRAM	IRAM assignment	1 A0RAM	0 IVRAM	0	R/W	

D[31:1] Reserved

D0 IRAM: IRAM Assignment Bit

Selects whether the 12K-byte IVRAM is used as a VRAM or a general-purpose RAM.

1 (R/W): A0RAM

0 (R/W): IVRAM (default)

The IVRAM is located at 0x80000 to 0x82FFF in area 3 by default and it can be used as a VRAM. When the IVRAM is not used as a VRAM, it can be located in area 0 as a general-purpose RAM by setting IRAM to 1.

S1C33E08 Technical Manual

IX PERIPHERAL MODULES 7 (USB)

IX.1 USB Function Controller (USB)

IX.1.1 Outline of the USB Function Controller

The S1C33E08 has a built-in USB function controller that supports the Full-Speed mode defined in the USB2.0 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 IX.1.1.1 shows the block diagram of the USB function controller.

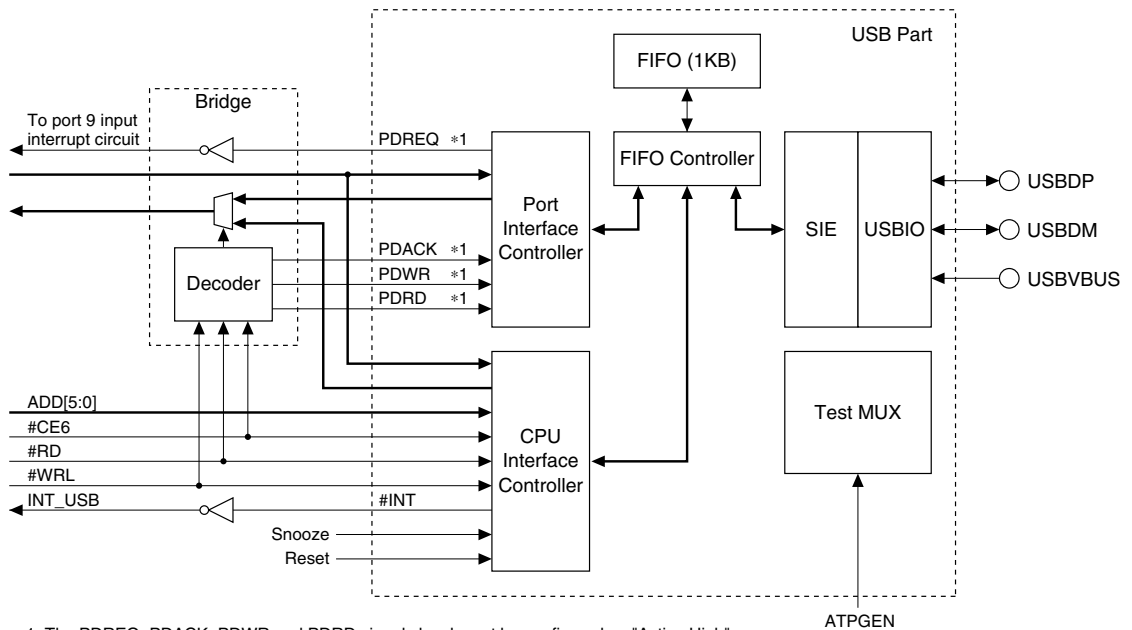


Figure IX.1.1.1 USB Function Controller Block Diagram

IX PERIPHERAL MODULES 7 (USB): USB FUNCTION CONTROLLER (USB)

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.

IX.1.2 Pins for the USB Interface

Table IX.1.2.1 list the pins used for the USB interface.

Table IX.1.2.1 USB Interface Pins

Pin name	I/O	Function
USBDP	I/O	USB D+ pin
USBDM	I/O	USB D- pin
USBVBUS	I	USB VBUS pin. Allows input of 5 V.

IX.1.3 USB Operating Clocks and DMA Registers

IX.1.3.1 Controlling the USB Clocks

The CMU provides the clock paths with a control bit shown below for the USB module. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) USB clock (USB_CLK)

This clock (OSC3 = 48 MHz) is used for the USB interface module. USB_CKE (D8/0x301B00) is used for clock supply control (default: off).

(2) Control register clock (USBSAPB_CLK)

This clock (MCLK) is used to control the USB registers located in area 6. This clock is required for accessing the USB registers and the DMA area, and it can be stopped otherwise. USBSAPB_CKE (D9/0x301B00) is used for clock supply control (default: off).

Note: The Gated Clock Control Register 0 (0x301B00) is write-protected. To rewrite this register and other CMU control registers at addresses 0x301B00 to 0x301B14, write protection must be removed by writing 0x96 to the Clock Control Protect Register (0x301B24). Since unnecessary rewrites to addresses 0x301B00 to 0x301B14 may cause the system to operate erratically, make sure that data set in the Clock Control Protect Register (0x301B24) is other than 0x96 unless rewriting said registers.

IX.1.3.2 Setting the Misc Register

1. USBWT[2:0] (D[2:0])/USB Wait Control Register (0x300012)
USBWT[2:0] (D[2:0]/0x300012) used to set the number of wait cycles to be inserted when accessing the USB registers.

Table IX.1.3.2.1 Number of Wait Cycles during USB Access

USBWT2	USBWT1	USBWT0	Number of wait cycles (in units of MCLK cycles)	MCLK clock frequency
1	1	1	7 cycles	Less than 60 MHz
1	1	0	6 cycles	Less than 56 MHz
1	0	1	5 cycles	Less than 45 MHz
1	0	0	4 cycles	Less than 36 MHz
0	1	1	3 cycles	Less than 24 MHz
0	1	0	2 cycles	Less than 16 MHz
0	0	1	1 cycle	Less than 8 MHz
0	0	0	0 cycles	Less than 8 MHz

(Default: 0b111 = 7 cycles)

The number of wait cycles should be set according to the MCLK clock frequency.

2. USBSNZ (D5)/USB Wait Control Register (0x300012) = 0
This disables the USB snooze control.

The OSC3 oscillator circuit must be turned on before the USB function controller can be used. Refer to “III.1 Clock Management Unit (CMU)” and “III.4 Misc Registers” for details of the clock control.

IX.1.3.3 Setting the ITC and HSDMA Controllers

The USB function controller supports data transfer using the HSDMA (High-Speed DMA) function. Therefore, the ITC and the HSDMA (Ch.1) must be set up before starting USB data transfer. This section describes how the ITC and the HSDMA control registers are set up. Refer to “III.2 Interrupt Controller (ITC)” and “II.1 High-Speed DMA (HSDMA)” for details of the ITC and the HSDMA, respectively.

1. Setting the signals used for DMA transfer

The USB controller has the following pads used for DMA data transfer and they are connected to the corresponding SRAMC/ITC/HSDMA.

PDREQ * (DMA request interrupt signal)
 PDACK (DMA acknowledge signal)
 PDRD (Data read signal)
 PDWR (Data write signal)

* The PDREQ signal is inverted before it is input to the ITC.

The active level of these signals can be selected using the USB register. Set the respective control bits as follows.

PDREQ: PDREQ_Level (D3/0x300994) = 0 (Active High)
 PDACK: PDACK_Level (D2/0x300994) = 0 (Active High)
 PDRD, PDWR: PDRDWR_Level (D1/0x300994) = 0 (Active High)

2. Setting the ITC

DMA request (PDREQ)

The PDREQ signal used for a DMA request is input to the port 9 input interrupt (FPT9) system. The port 9 input interrupt circuit is configured by selecting a port (signal) to be used for generating an interrupt from P91, PDREQ, P81, and P71. When using the DMA request/interrupt by PDREQ, set SPT9[1:0] (D[3:2]/0x3003C4) to 10 to select PDREQ. This setting enables the PDREQ signal to be sent to the ITC as the port 9 input interrupt signal and it will be used as a trigger for HSDMA Ch.1. Furthermore, SPPT9 (D1/0x3003C6) for selecting a polarity of the FPT9 input signal should be set to 0 (low level or falling edge) since the PDREQ signal is inverted before input to the ITC.

The interrupt control bits (cause-of-interrupt flag, interrupt enable register, IDMA request register, interrupt priority register) do not affect this invocation.

USB interrupt (INT_USB)

The USB interrupt signal (INT_USB) is input to the port 10 input interrupt (FPT10) system. The port 10 input interrupt circuit is configured by selecting a port (signal) to be used for generating an interrupt from P92, INT_USB, P82, and P72. When using the DMA request/interrupt by INT_USB, set SPTA[1:0] (D[5:4]/0x3003C4) to 10 to select INT_USB. This setting enables the INT_USB signal to be sent to the ITC as the port 10 input interrupt signal for requesting an interrupt. Set SEPTA (D2/0x3003C7) = 0 and SPPTA (D2/0x3003C6) = 1 to select the detection mode and polarity of the FPT10 input signal to high level.

3. Setting the HSDMA

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.1 in this mode, set DUALM1 (D15/0x301132) to 1.

Note: Do not set the HSDMA Ch.1 to single-address mode when using it for USB data transfer.

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 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 HSDMA 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 D1MOD[1:0] (D[15:14]/0x30113A) 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 DATSIZE1. If data transfer need to be performed a number of times as set by the transfer counter, an equal number of triggers are required.

Setting the transfer data size

DATSIZE1 (D14/0x301136) is used to set the unit size of data to be transferred. Set this bit to 0 (8 bits).

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.

BLKLEN1[7:0]: Ch.1 transfer counter [7:0] (D[7:0])/HSDMA Ch.1 Transfer Counter Register (0x301130)

TC1_L[7:0]: Ch.1 transfer counter [15:8] (D[15:8])/HSDMA Ch.1 Transfer Counter Register (0x301130)

TC1_H[7:0]: Ch.1 transfer counter [23:16] (D[7:0])/HSDMA Ch.1 Control Register (0x301132)

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.

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.

S1ADRL[15:0]: Ch.1 source address [15:0] (D[15:0])/HSDMA Ch.1 Low-order Source Address Set-up Register (0x301134)

S1ADRH[11:0]: Ch.1 source address [27:16] (D[11:0])/HSDMA Ch.1 High-order Source Address Set-up Register (0x301136)

D1ADRL[15:0]: Ch.1 destination address [15:0] (D[15:0])/HSDMA Ch.1 Low-order Destination Address Set-up Register (0x301138)

D1ADRH[11:0]: Ch.1 destination address [27:16] (D[11:0])/HSDMA Ch.1 High-order Destination Address Set-up Register (0x30113A)

Note: The DMA transfer address for the USB function controller must be located in Area 6 (0x300A00 to 0x300AFF, 256 bytes). Make sure that the transfer address does not exceed the address range from 0x300A00 to 0x300AFF by the address increment/decrement operation during DMA transfer.

Setting the address increment/decrement conditions

The source and/or destination addresses can be incremented or decremented when one data transfer is completed. S1IN[1:0] (D[13:12]0x301136) for source address and D1IN[1:0] (D[13:12]/0x30113A) for destination address are used to set this condition.

S1IN/D1IN = 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.

S1IN/D1IN = 01: address decremented

The address is decremented by an amount equal to the data size set by DATSIZE1 when one data transfer is completed. The address that has been decremented during transfer does not return to the initial value.

S1IN/D1IN = 10 or 11: address incremented

The address is incremented by an amount equal to the data size set by DATSIZE1 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.

Selecting the DMA trigger factor

The HSDMA trigger factor for the USB function controller is port 9 input (FPT9). HSD1S[3:0] (D[7:4]/0x300289) must be set to 1101.

By selecting a cause of interrupt with the HSDMA trigger set-up register, the HSDMA channel is invoked when the selected cause of interrupt occurs. The interrupt control bits (cause-of-interrupt flag, interrupt enable register, IDMA request register, interrupt priority register) do not affect this invocation. The cause of interrupt that invokes HSDMA sets the cause-of-interrupt 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 cause of interrupt 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 HS3_TF (D0/0x30113E). 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.

Enabling/Disabling DMA transfer

The HSDMA transfer is enabled by writing 1 to HS1_EN (D0/0x30113C). However, the control information must always be set correctly before enabling a DMA transfer. Note that the control information cannot be set when HS1_EN = 1. When HS1_EN is set to 0, HSDMA requests are no longer accepted. When a DMA transfer is completed (transfer counter = 0), HS1_EN is reset to 0 to disable the following trigger inputs.

IX.1.4 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.RcvEPOSETUP bit”, or “ForceNAK bit of the EPOControlOUT register”
- * When a register present for a specific end-point is referred to:
 - EP x { $x=0,a,b,c,d$ }register name + register, EP x { $x=a,b,c,d$ }register name + register, and so forth.
 - Example: “EP x { $x=0,a,b,c,d$ }IntStat register”, “EP x { $x=a,b,c,d$ }Control register”

IX.1.4.1 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 terms 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 IX.1.4.1.1 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 IX.1.4.1.1 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 IX.1.4.1.2 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 four discrete endpoints to be used. Set up and/or enable these endpoints as appropriate according to the definitions for the USB-defined interface.

Table IX.1.4.1.2 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 IX.1.4.1.3 lists control items and statuses related to transaction control on the EP0 endpoint.

Table IX.1.4.1.3 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	EP0Control.IN.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	EP0Control.IN.ToggleStat, EP0Control.OUT.ToggleStat	Indicates the state of the toggle sequence bit. This setting is automatically initialized by the SETUP stage.
Set toggle	EP0Control.IN.ToggleSet, EP0Control.OUT.ToggleSet	Sets the toggle sequence bit.
Clear toggle	EP0Control.IN.ToggleClr, EP0Control.OUT.ToggleClr	Clears the toggle sequence bit.
Forced NAK response	EP0Control.IN.ForceNAK, EP0Control.OUT.ForceNAK	Returns a NAK response to IN or OUT transactions regardless of the number of data or vacancies in the FIFO region.
STALL response	EP0Control.IN.ForceSTALL, EP0Control.OUT.ForceSTALL	Returns a STALL response to IN or OUT transactions.
Set automatic ForceNAK	EP0Control.OUT.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.

IX PERIPHERAL MODULES 7 (USB): USB FUNCTION CONTROLLER (USB)

Table IX.1.4.1.4 lists control items and statuses related to transaction processing on the EPa, EPb, EPc, and EPd endpoints.

Table IX.1.4.1.4 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_TrانACK, EPx{x=a,b,c,d}IntStat.OUT_TrانACK, EPx{x=a,b,c,d}IntStat.IN_TrانNAK, EPx{x=a,b,c,d}IntStat.OUT_TrانNAK, EPx{x=a,b,c,d}IntStat.IN_TrانErr, EPx{x=a,b,c,d}IntStat.OUT_TrانErr	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 IX.1.4.1.1 illustrates how the SETUP transaction is executed.

- The host issues a SETUP token addressed to the EP0 endpoint of this node.
- Next, the host sends an 8-byte long data packet. The macro writes these data in the EP0Setup_0 through EP0Setup_7 registers.
- The macro automatically returns an ACK response. In addition, it sets registers to be automatically set up and issues a status to the firmware.

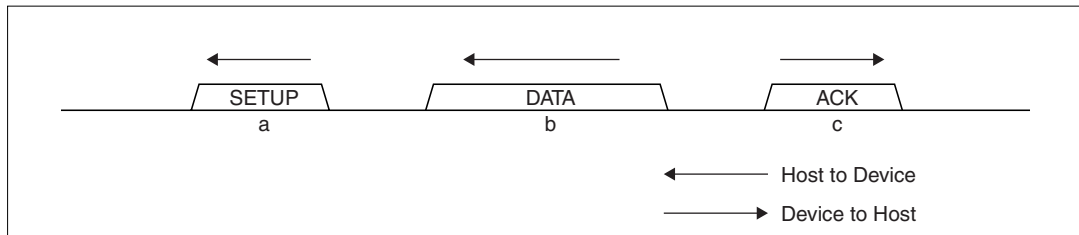


Figure IX.1.4.1.1 SETUP Transaction

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_TransACK status of the relevant endpoint (EP x { $x=0,a,b,c,d$ }IntStat.OUT_TransACK 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 (EP x { $x=a,b,c,d$ }IntStat.OUT_ShortACK bit) to be issued, in addition to executing the above closing process. If the EP x { $x=a,b,c,d$ }Control.DisAF_NAK_Short bit is cleared, the relevant endpoint's EP x { $x=a,b,c,d$ }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_TransErr status (EP x { $x=0,a,b,c,d$ }IntStat.OUT_TransErr 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_TransNAK status (EP x { $x=a,b,c,d$ }IntStat.OUT_TransNAK bit) is issued. Accordingly, the FIFO is not updated.

Figure IX.1.4.1.2 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.

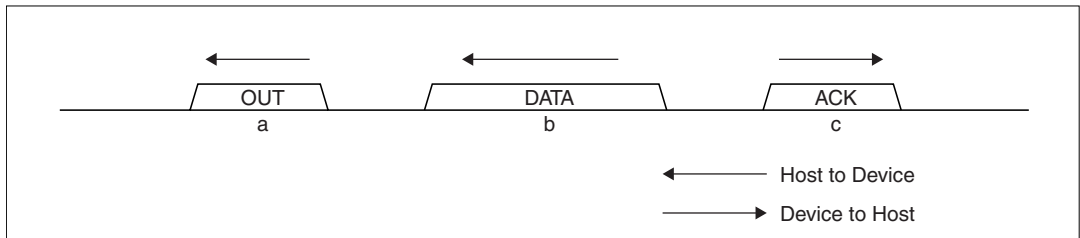


Figure IX.1.4.1.2 OUT Transaction

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_TrانACK status (EPx{x=0,a,b,c,d}IntStat.IN_TrانACK 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_TrانErr status (EPx{x=0,a,b,c,d}IntStat.IN_TrانErr 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_TrانNAK status (EPx{x=0,a,b,c,d}IntStat.IN_TrانNAK bit) is issued to the firmware. Accordingly, the FIFO is not updated, or no space is freed.

Figure IX.1.4.1.3 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.

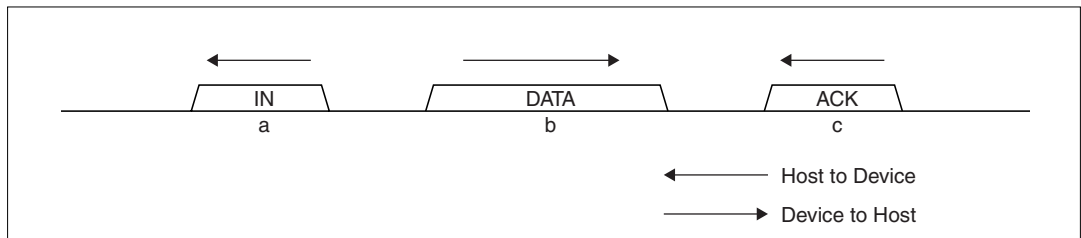


Figure IX.1.4.1.3 IN Transaction

Control transfer

Control transfer on the EP0 endpoint is controlled as a combination of a number of discrete transactions. Figure IX.1.4.1.4 illustrates how control transfer is executed for an OUT data stage.

- 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.
- The host issues an OUT transaction and executes a data stage, and the device receives data.
- 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_TrانNAK status (EP0IntStat.IN_TrانNAK bit) as a trigger to transit to a status stage from a data stage.

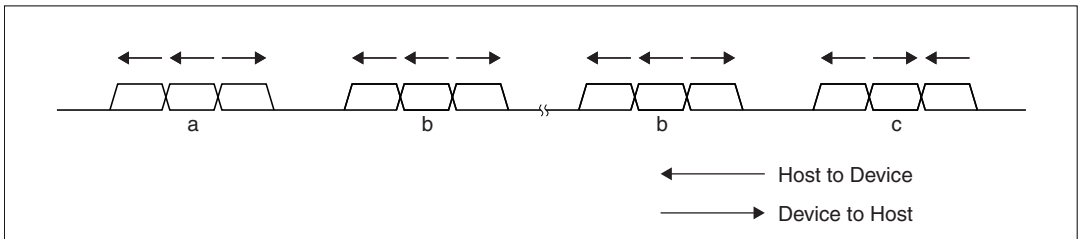


Figure IX.1.4.1.4 Control Transfer Having an OUT Data Stage

Figure IX.1.4.1.5 illustrates how control transfer is executed for an IN data stage.

- 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.
- The host issues an IN transaction and executes a data stage, and the device transmits data.
- 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_TrانNAK status (EP0IntStat.OUT_TrانNAK bit) as a trigger to transit to a status stage from a data stage.

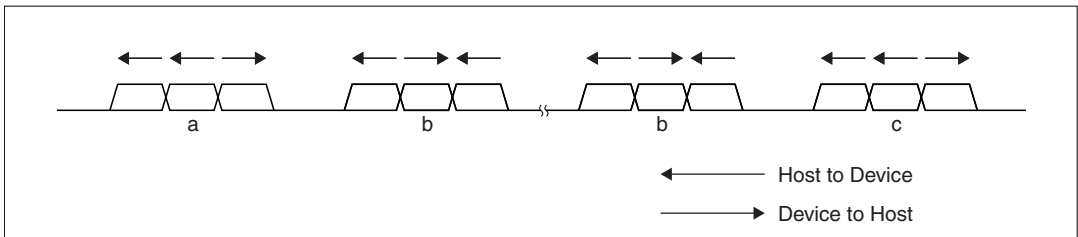


Figure IX.1.4.1.5 Control Transfer Having an IN Data Stage

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 (SIE_IntStat.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 a 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 (FIFO_IntStat.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 a 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 and DMA_Remain_L registers 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 IX.1.4.1.6 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.

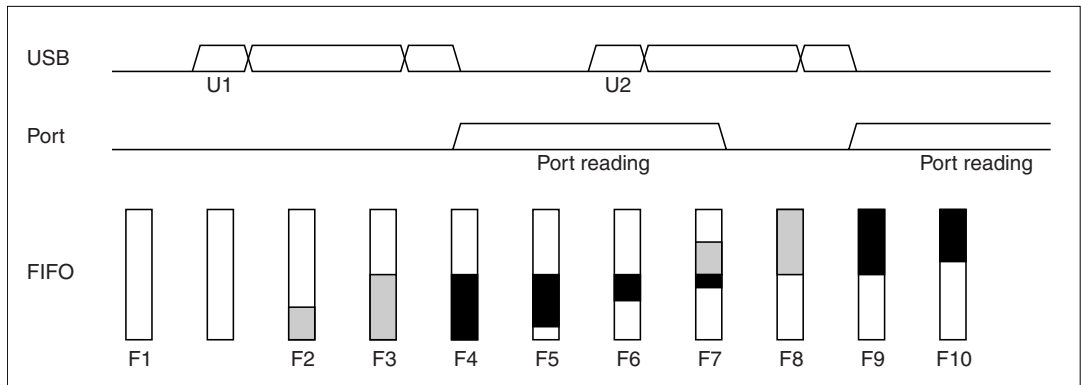


Figure IX.1.4.1.6 Example of Data Flow in OUT Transfer

- (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 PDREQ 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 PDREQ 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 through 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 and EPnWrRemain_L registers 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_FIFO_Control.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 IX.1.4.1.7 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.

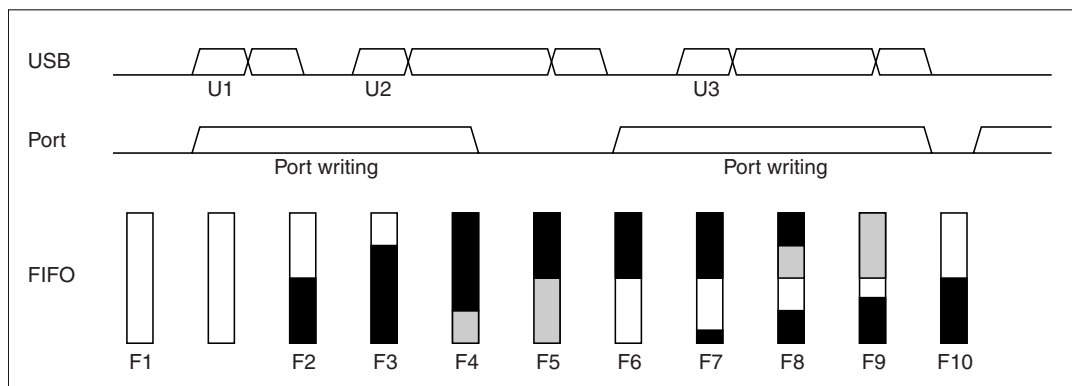


Figure IX.1.4.1.7 Example of Data Flow in IN Transfer

- (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 PDREQ 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 PDREQ 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 PDREQ 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 and Resume detection, checking the state of the USB bus for each operation. You can check each interruption (DetectReset and DetectSuspend) to confirm what has been actually performed.

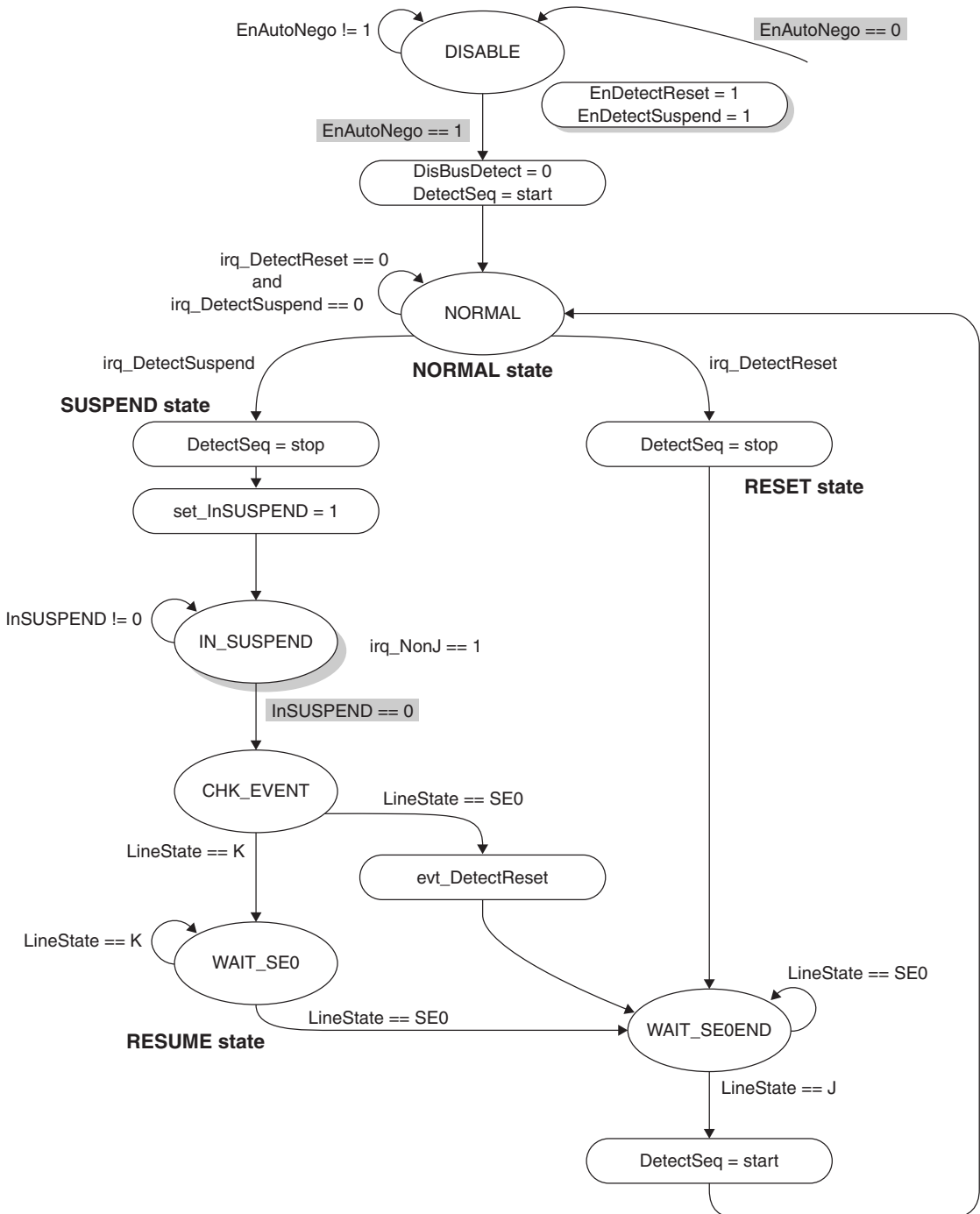


Figure IX.1.4.1.8 Auto-negotiator

(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 μ s 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 suspended, 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 starting 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 (T₁).
- (2) At T₂, 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 USBSNZ (D5/0x300012) to 1 to start processing Snooze before reaching T₄. As for self-powered products, however, the firmware does not have to perform Snooze. (Figure IX.1.4.1.9 shows the operation when Snooze is performed.)

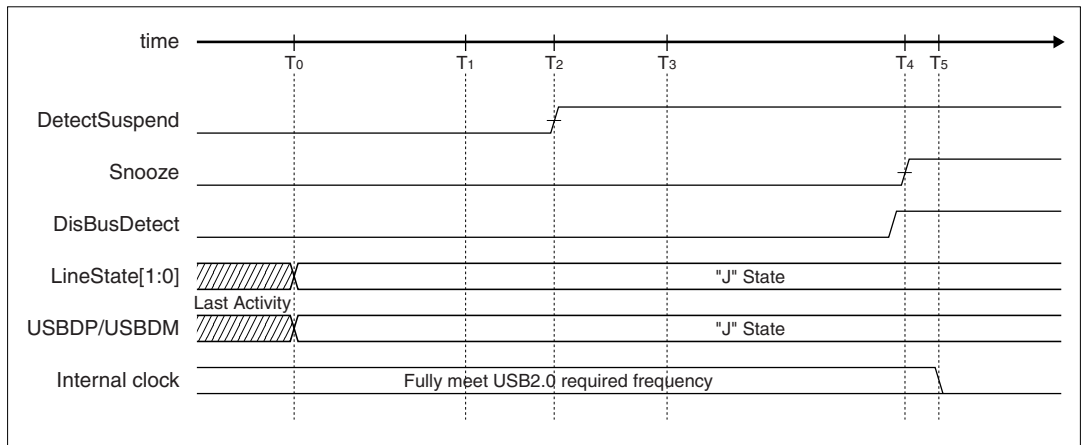


Figure IX.1.4.1.9 Suspend Timing (FS mode)

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 μ s 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.

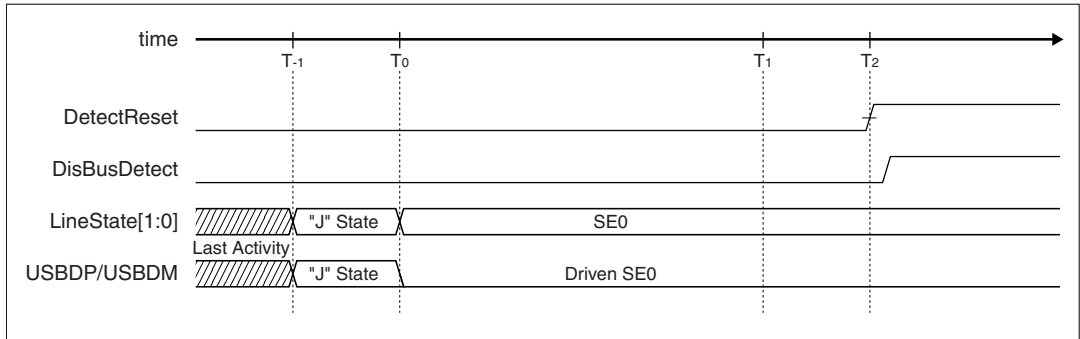


Figure IX.1.4.1.10 Reset Timing (FS mode)

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 S1C33E08 supports Snooze state. This section describes the operation for issuing Resume when the oscillation circuit is in operation (USB_CKE (D8/0x301B00) = 1, not in Sleep). Steps (3), (4), (8) and (9) below are handled by the macro hardware automatically. Perform steps (1), (2), (6), (6a) and (10) on the firmware that controls this macro.

- (1) Clear the SIE_IntEnb.EnNonJ and USBSNZ (D5/0x300012) 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 XcvtControl.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, clear the USB_Control.InSUSPEND 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.

To detect the Resume signal sent from downstream port, the following procedure is needed after step (6) is performed.

- (6a) Set the USB_Control.StartDetectJ 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 SIE_IntStat.DetectJ bit is set.
- (9) If the SIE_IntEnb.EnDectectJ bit is set, the macro asserts the #INT signal.
- (10) Clear the USB_Control.StartDetectJ bit.

However, steps (6a)–(10) is not necessary when the auto-negotiation function is used, so just wait another event.

This section describes the operation of the oscillation circuit by assuming that it is in operation (USB_CKE (D8/0x301B00) = 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 USBSNZ (D5/0x300012) 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), (5), (5a) and (9). 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 USBSNZ (D5/0x300012).
- (5) Clear the USB_Control.SendWakeup bit and suspend Resume signal send-out. After that, clear the USB_Control.InSUSPEND bit.
- (6) The downstream port suspends “K” send-out.

To detect the Resume signal sent from downstream port, the following procedure is needed after step (5) is performed.

- (5a) Set the USB_Control.StartDetectJ bit.
- (6) The downstream port suspends “K” send-out.
- (7) The SIE_IntStat.DetectJ bit is set.
- (8) If the SIE_IntEnb.EnDectectJ bit is set, the macro asserts the #INT signal.
- (9) Clear the USB_Control.StartDetectJ bit.

However, steps (5a)–(9) is not necessary when the auto-negotiation function is used, so just wait another event.

This section describes the operation of the oscillation circuit by assuming that it is in operation (USB_CKE (D8/0x301B00) = 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 USBSNZ (D5/0x300012) 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 bits (T₀).
- (2) If the SIE_IntEnb.EnVBUS_Changed and MainIntEnb.EnSIE_IntStat bits are set, the macro asserts the #INT signal.
- (3) Set USB_CKE (D8/0x301B00) to start supplying the USB clock (T₁).
- (4) Clear USBSNZ (D5/0x300012) (T₂).
- (5) The downstream port sends out Reset (T₄).

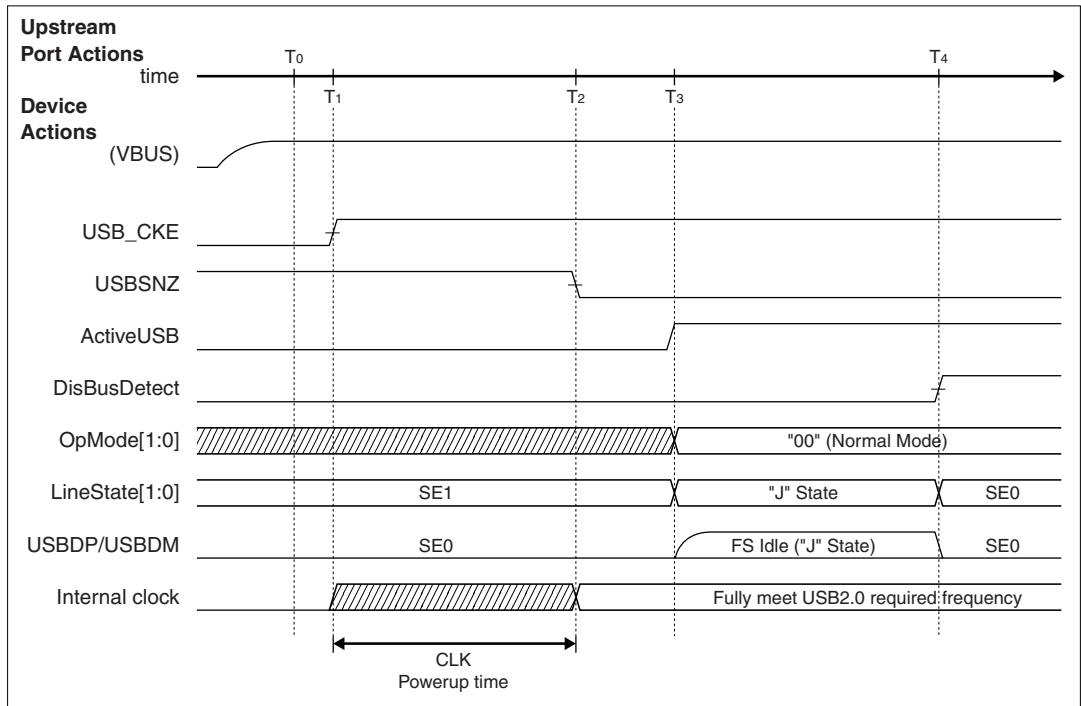


Figure IX.1.4.1.11 Device Attach Timing

Table IX.1.4.1.5 Device Attach Timing Values

Timing parameter	Description	Value
T ₀	VBUS is enabled.	0 (Reference)
T ₁	Set USB_CKE 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

IX.1.4.2 FIFO Management

FIFO memory map

This section describes the memory map for the FIFO region.

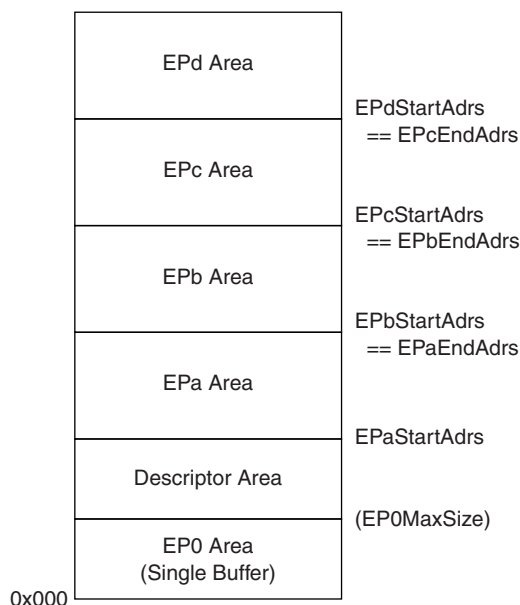


Figure IX.1.4.2.1 FIFO Memory Map

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 as 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 EPaStartAdrs. (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 registers, 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 FIFO_IntStat.DescriptorCmp is set. At this stage, the FIFO_IntEnb.EnDescriptorCmp bit is set and the MainIntEnb.EnEPrIntStat 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_TransNAK status and the FIFO_IntStat.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_TransNAK, MainIntEnb.EnEP0IntStat and MainIntEnb.EnEPrIntStat bits, or
- (2) The FIFO_IntEnb.EnDescriptorCmp and MainIntEnb.EnEPrIntStat 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 registers 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.

IX.1.4.3 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 Read/Write-strobe signal.

Basic operations

This section describes the basic operations of the Port interface.

Register setting

Table IX.1.4.3.1 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 IX.1.4.3.1 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.PDRDWR_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 PDREQ.
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 DMA_IntStat.DMA_Cmp bit 1. At this time, if the DMA_IntEnb.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, Data Transfer Permit signal PDACK and Read-strobe PDRD/Write-strobe PDWR. 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 to the HSDMA (master) if any available space is found at the connected endpoint. The DMA loads the data and writes them to the endpoint when PDWR is rising (when the DMA_Config_0.PDRDWR_Level bit is set to 1). When available space is entirely consumed at the endpoint, the interface negates PDREQ to the HSDMA (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 once after completing transfer of 4-byte data, and does not assert PDREQ as long as $130 \text{ ns} \times N$ ($N = \text{DMA_Latency.DMA_Latency}[3:0]$).

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 HSDMA (master) and then terminate the macro's DMA transfer.

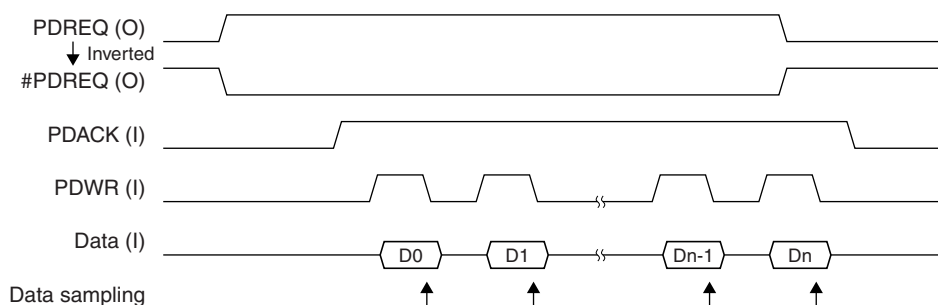


Figure IX.1.4.3.1 Transfer Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Writing

Providing 1 to the DMA_Config_1.DMA_RcvLimitMode bit enables the RcvLimit mode. The RcvLimit mode is not available in Countdown mode.

When the DMA is writing asynchronously in RcvLimit mode, up to 16 bytes of data can be received even after this macro negates PDREQ.

In this mode, PDREQ 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 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 is negated.

In this mode, PDREQ is negated before the endpoint becomes completely full. If the region set with the EP{a,b,c,d}StartAdrs 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}StartAdrs register.

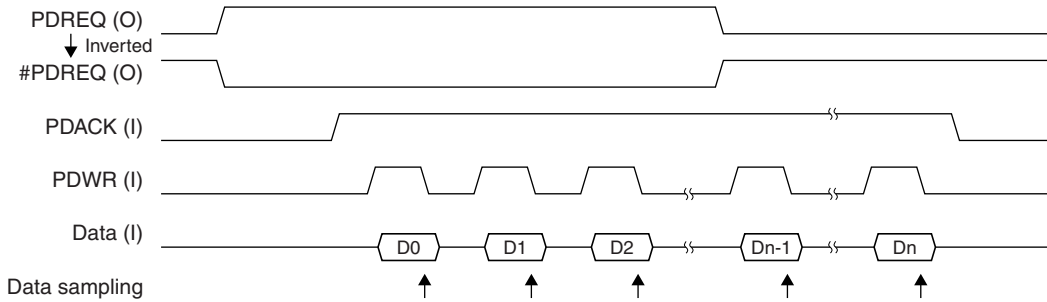


Figure IX.1.4.3.2 Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Writing (RcvLimit mode)

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 to the HSDMA (master) if any data exist at the connected endpoint. Turning PDACK to active starts outputting transferred data to the data bus. Have the HSDMA (master) load the data while PDRD is rising (when the DMA_Config_0.PDRDWR_Level bit is set to 1). When no data remains at the endpoint, the interface negates PDREQ to the HSDMA (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 once after completing transfer of 4-byte data, and does not assert PDREQ as long as $130 \text{ ns} \times N$ ($N = \text{DMA_Latency.DMA_Latency}[3:0]$).

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 HSDMA (master) and then terminate the macro's DMA transfer.

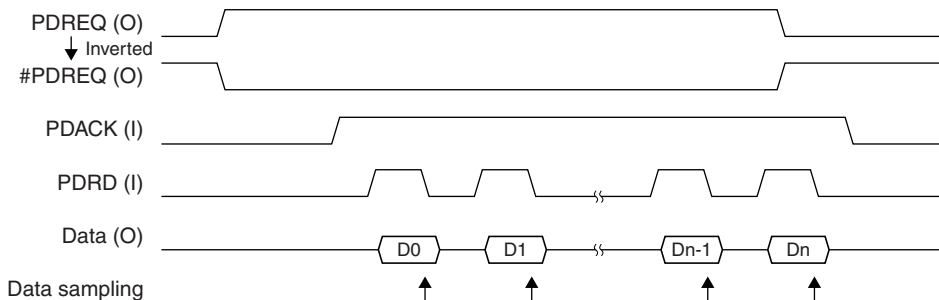


Figure IX.1.4.3.3 Transfer Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Reading

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 to the HSDMA (master) if any available space is found at the connected endpoint. The DMA loads the data and writes them to the endpoint when PDWR is rising (when the DMA_Config_0.PDRDWR_Level bit is set to 1). This mode negates PDREQ after transferring 1-byte data (PDWR becomes active).

At this point, if any space is still available at the endpoint, it requests data transfer by asserting PDREQ to the HSDMA (master). If there is no available space left at the endpoint, PDREQ 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 once after completing transfer of 4-byte data, and does not assert PDREQ as long as $130 \text{ ns} \times N$ ($N = \text{DMA_Latency.DMA_Latency}[3:0]$).

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 HSDMA (master) and then terminate the macro's DMA transfer.

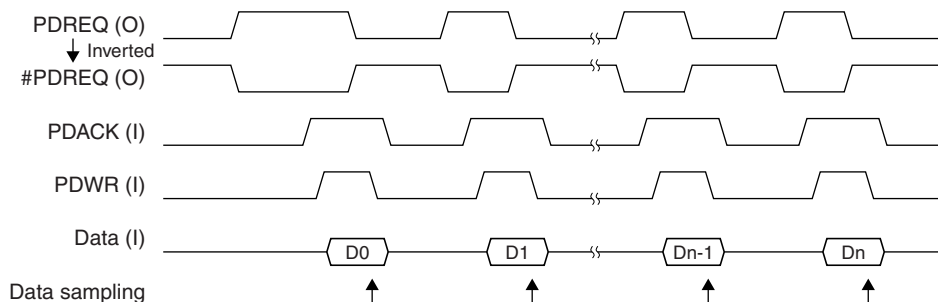


Figure IX.1.4.3.4 Transfer Waveforms in Asynchronous Single-Word DMA Transfer Mode - Writing

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 to the HSDMA (master) if any data exist at the connected endpoint. Turning PDACK to active starts outputting transferred data to the data bus. Have the HSDMA (master) load the data while PDRD is rising (when the DMA_Config_0.PDRDWR_Level bit is set to 1). This mode negates PDREQ after transferring 1-byte data (PDRD becomes active). At this point, if any data still remain at the endpoint, it requests data transfer by asserting PDREQ to the HSDMA (master). If there are no data left at the endpoint, PDREQ 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 once after completing transfer of 4-byte data, and does not assert PDREQ as long as $130 \text{ ns} \times N$ ($N = \text{DMA_Latency.DMA_Latency}[3:0]$).

IX PERIPHERAL MODULES 7 (USB): USB FUNCTION CONTROLLER (USB)

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 HSDMA (master) and then terminate the macro's DMA transfer.

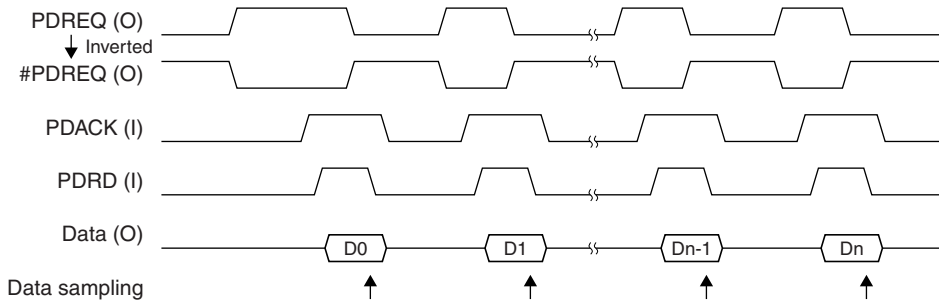


Figure IX.1.4.3.5 Transfer Waveforms in Asynchronous Single-Word DMA Transfer Mode - Reading

IX.1.4.4 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 (D5/0x300012), 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 bits of the SIE_IntStat register. (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.

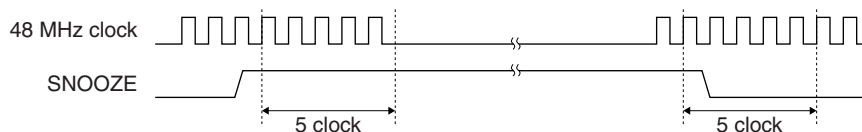


Figure IX.1.4.4.1 Snooze Sequence

Snooze mode should be set or canceled by the following procedure:

Setting snooze mode

- (1) Write 0x96 to the Misc Protect Register (0x300020) to disable write protection for the Misc registers.
- (2) Set USBSNZ (D5/0x300012) in the USB Wait Control Register to 1 to enable the snooze control.
- (3) Write a value other than 0x96 to the Misc Protect Register (0x300020) to enable write protection for the Misc registers.
- (4) Write 0x96 to the Clock Control Protect Register (0x301B24) to disable write protection for the CMU registers.
- (5) Set USB_CKE (D8/0x301B00) in the Gated Clock Control Register 0 to 0 to stop supplying the USB clock.
- (6) Write a value other than 0x96 to the Clock Control Protect Register (0x301B24) to enable write protection for the CMU registers.

Canceling snooze mode

- (1) Write 0x96 to the Clock Control Protect Register (0x301B24) to disable write protection for the CMU registers.
- (2) Set USB_CKE (D8/0x301B00) in the Gated Clock Control Register 0 to 1 to start supplying the USB clock.
- (3) Write a value other than 0x96 to the Clock Control Protect Register (0x301B24) to enable write protection for the CMU registers.
- (4) Write 0x96 to the Misc Protect Register (0x300020) to disable write protection for the Misc registers.
- (5) Set USBSNZ (D5/0x300012) in the USB Wait Control Register to 0 to disable the snooze control.
- (6) Write a value other than 0x96 to the Misc Protect Register (0x300020) to enable write protection for the Misc registers.

IX.1.5 Registers

IX.1.5.1 List of Registers

• *Italic & bold* represents readable/writable registers in SNOOZE/SLEEP mode.

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
300900	MainIntStat	R/(W)	0x00	<i>SIE_IntStat</i>	EPrintStat	DMA_IntStat	FIFO_IntStat			EP0IntStat	RcvEP0SETUP
300901	SIE_IntStat	R/(W)	0x00	<i>VBUS_Changed</i>	<i>NonJ</i>	DetectReset	DetectSuspend	RcvSOF	DetectJ		SetAddressCmp
300902	EPrintStat	R	0x00					EPdIntStat	EPcIntStat	EPbIntStat	EPaIntStat
300903	DMA_IntStat	R/(W)	0x00							DMA_CountUp	DMA_Cmp
300904	FIFO_IntStat	R/(W)	0x00	DescriptorCmp						FIFO_IN_Cmp	FIFO_OUT_Cmp
300905											
300906											
300907	EP0IntStat	R/(W)	0x00			IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
300908	EPaIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
300909	EPbIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
30090A	EPcIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
30090B	EPdIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
30090C											
30090D											
30090E											
30090F											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
300910	MainIntEnb	R/W	0x00	<i>EnSIE_IntStat</i>	EnEPrintStat	EnDMA_IntStat	EnFIFO_IntStat			EnEP0IntStat	EnRcvEP0SETUP
300911	SIE_IntEnb	R/W	0x00	<i>EnVBUS_Changed</i>	<i>EnNonJ</i>	EnDetectReset	EnDetectSuspend	EnRcvSOF	EnDetectJ		EnSetAddressCmp
300912	EPrintEnb	R/W	0x00					EnEPdIntStat	EnEPcIntStat	EnEPbIntStat	EnEPaIntStat
300913	DMA_IntEnb	R/W	0x00							EnDMA_CountUp	EnDMA_Cmp
300914	FIFO_IntEnb	R/W	0x00	EnDescriptorCmp						EnFIFO_IN_Cmp	EnFIFO_OUT_Cmp
300915											
300916											
300917	EP0IntEnb	R/W	0x00			EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
300918	EPaIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
300919	EPbIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
30091A	EPcIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
30091B	EPdIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
30091C											
30091D											
30091E											
30091F											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
300920	RevisionNum	R	0x12	<i>Revision Number[7:0]</i>									
300921	USB_Control	R/W	0x00	DisBusDetect	EnAutoNego	InSUSPEND	StartDetectJ	SendWakeUp			ActiveUSB		
300922	USB_Status	R	0xXX	<i>VBUS</i>	<i>1(FS)</i>					<i>LineState[1:0]</i>			
300923	XcrrControl	R/W	0x01	RpuEnb						<i>OpMode[1:0]</i>			
300924	USB_Test	R/W	0x00	EnUSB_Test				Test_SE0_NAK	Test_J	Test_K	Test_Packet		
300925	EPnControl	W	0x00	AllForceNAK	EPrForceSTALL	AllFIFO_Clr					EP0FIFO_Clr		
300926	EPrFIFO_Clr	W	0x00					EPdFIFO_Clr	EPcFIFO_Clr	EPbFIFO_Clr	EPaFIFO_Clr		
300927													
300928													
300929													
30092A													
30092B													
30092C													
30092D													
30092E	FrameNumber_H	R	0x80	FnlInvalid						FrameNumber[10:8]			
30092F	FrameNumber_L	R	0x00				FrameNumber[7:0]						

IX PERIPHERAL MODULES 7 (USB): USB FUNCTION CONTROLLER (USB)

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
300930	EPOSetup_0	R	0x00										
300931	EPOSetup_1	R	0x00										
300932	EPOSetup_2	R	0x00										
300933	EPOSetup_3	R	0x00										
300934	EPOSetup_4	R	0x00										
300935	EPOSetup_5	R	0x00										
300936	EPOSetup_6	R	0x00										
300937	EPOSetup_7	R	0x00										
300938	USB_Address	R/W	0x00	AutoSetAddress	USB_Address[6:0]								
300939	EP0Control	R/W	0x00	InxOUT							ReplyDescriptor		
30093A	EP0ControlIN	R/W	0x00		EnShortPkt		ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL		
30093B	EP0ControlOUT	R/W	0x00	AutoForceNAK			ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL		
30093C													
30093D													
30093E													
30093F	EP0MaxSize	R/W	0x08	EP0MaxSize[6:3]									

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
300940	EPaControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
300941	EPbControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
300942	EPcControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
300943	EPdControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
300944											
300945											
300946											
300947											
300948											
300949											
30094A											
30094B											
30094C											
30094D											
30094E											
30094F											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
300950	EPaMaxSize_H	R/W	0x00							EPaMaxSize[9:8]			
300951	EPaMaxSize_L	R/W	0x00	EPaMaxSize[7:0]									
300952	EPaConfig_0	R/W	0x00	InxOUT	ToggleMode	EnEndPoint			EndPointNumber[3:0]				
300953	EPaConfig_1	R/W	0x00	ISO	ISO_CRCmode								
300954	EPbMaxSize_H	R/W	0x00							EPbMaxSize[9:8]			
300955	EPbMaxSize_L	R/W	0x00	EPbMaxSize[7:0]									
300956	EPbConfig_0	R/W	0x00	InxOUT	ToggleMode	EnEndPoint			EndPointNumber[3:0]				
300957	EPbConfig_1	R/W	0x00	ISO	ISO_CRCmode								
300958	EPcMaxSize_H	R/W	0x00							EPcMaxSize[9:8]			
300959	EPcMaxSize_L	R/W	0x00	EPcMaxSize[7:0]									
30095A	EPcConfig_0	R/W	0x00	InxOUT	ToggleMode	EnEndPoint			EndPointNumber[3:0]				
30095B	EPcConfig_1	R/W	0x00	ISO	ISO_CRCmode								
30095C	EPdMaxSize_H	R/W	0x00							EPdMaxSize[9:8]			
30095D	EPdMaxSize_L	R/W	0x00	EPdMaxSize[7:0]									
30095E	EPdConfig_0	R/W	0x00	InxOUT	ToggleMode	EnEndPoint			EndPointNumber[3:0]				
30095F	EPdConfig_1	R/W	0x00	ISO	ISO_CRCmode								

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
300970	EPaStartAdrs_H	R/W	0x00						EPaStartAdrs[11:8]				
300971	EPaStartAdrs_L	R/W	0x00	EPaStartAdrs[7:2]									
300972	EPbStartAdrs_H	R/W	0x00						EPbStartAdrs[11:8]				
300973	EPbStartAdrs_L	R/W	0x00	EPbStartAdrs[7:2]									
300974	EPcStartAdrs_H	R/W	0x00						EPcStartAdrs[11:8]				
300975	EPcStartAdrs_L	R/W	0x00	EPcStartAdrs[7:2]									
300976	EPdStartAdrs_H	R/W	0x00						EPdStartAdrs[11:8]				
300977	EPdStartAdrs_L	R/W	0x00	EPdStartAdrs[7:2]									
300978													
300979													
30097A													
30097B													
30097C													
30097D													
30097E													
30097F													

IX PERIPHERAL MODULES 7 (USB): USB FUNCTION CONTROLLER (USB)

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
300980	CPU_JoinRd	R/W	0x00					JoinEPdRd	JoinEPcRd	JoinEPbRd	JoinEPaRd
300981	CPU_JoinWr	R/W	0x00					JoinEPdWr	JoinEPcWr	JoinEPbWr	JoinEPaWr
300982	EnEPnFIFO_Access	R/W	0x00							EnEPnFIFO_Wr	EnEPnFIFO_Rd
300983	EPnFIFOforCPU	R/W	0xXX	EPnFIFOData[7:0]							
300984	EPnRdRemain_H	R	0x00						EPnRdRemain[11:8]		
300985	EPnRdRemain_L	R	0x00	EPnRdRemain[7:0]							
300986	EPnWrRemain_H	R	0x00						EPnWrRemain[11:8]		
300987	EPnWrRemain_L	R	0x00	EPnWrRemain[7:0]							
300988	DescAdrs_H	R/W	0x00						DescAdrs[11:8]		
300989	DescAdrs_L	R/W	0x00	DescAdrs[7:0]							
30098A	DescSize_H	R/W	0x00							DescSize[9:8]	
30098B	DescSize_L	R/W	0x00	DescSize[7:0]							
30098C											
30098D											
30098E											
30098F	DescDoor	R/W	0x00	DescMode[7:0]							

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
300990	DMA_FIFO_Control	R/W	0x00	FIFO_Running	AutoEnShort						
300991	DMA_Join	R/W	0x00					JoinEPdDMA	JoinEPcDMA	JoinEPbDMA	JoinEPaDMA
300992	DMA_Control	R/W	0xX0	DMA_Running	PDREQ	PDACK		CounterClr		DMA_Stop	DMA_Go
300993											
300994	DMA_Config_0	R/W	0x00	ActivePort				PDREQ_Level	PDACK_Level	PDRDWR_Level	
300995	DMA_Config_1	R/W	0x00	RcvLimitMode				SingleWord			CountMode
300996											
300997	DMA_Latency	R/W	0x00						DMA_Latency[3:0]		
300998	DMA_Remain_H	R	0x00						DMA_Remain[11:8]		
300999	DMA_Remain_L	R	0x00	DMA_Remain[7:0]							
30099A											
30099B											
30099C	DMA_Count_HH	R/W	0x00	DMA_Count[31:24]							
30099D	DMA_Count_HL	R/W	0x00	DMA_Count[23:16]							
30099E	DMA_Count_LH	R/W	0x00	DMA_Count[15:8]							
30099F	DMA_Count_LL	R/W	0x00	DMA_Count[7:0]							

IX.1.5.2 Detailed Description of Registers

0x300900: MainIntStat (Main Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
MainIntStat (Main interrupt status)	00300900 (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 causes of interrupt having occurred in the USB function controller. This register has the bit indirectly showing causes of interrupt and the bit directly showing causes of interrupt.

The bit indirectly showing causes of interrupt can be traced to the bit directly showing causes of interrupt by reading the relevant status registers. The bit showing causes of interrupt is read-only, and is automatically cleared by clearing the bit directly showing causes of interrupt at the main source. The bits showing causes of interrupt are writable, and the causes of interrupt can be cleared by setting the relevant bits to 1. When the corresponding bits are enabled by the MainIntEnb register, setting the cause of interrupt to 1 asserts the #INT signal, and causes an interruption of the CPU. Clearing all relevant causes of interrupt negates the #INT signal.

D7 SIE_IntStat

Shows a cause of interrupt indirectly.

When the SIE_IntStat register has a cause of interrupt and the SIE_IntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1. Reading this bit is valid during snooze as well.

D6 EPrIntStat

Shows a cause of interrupt indirectly.

When the EPrIntStat register has a cause of interrupt and the EPrIntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1.

D5 DMA_IntStat

Shows a cause of interrupt indirectly.

When the DMA_IntStat register has a cause of interrupt and the DMA_IntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1.

D4 FIFO_IntStat

Shows a cause of interrupt indirectly.

When the FIFO_IntStat register has a cause of interrupt and the FIFO_IntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1.

D[3:2] Reserved

D1 EP0IntStat

Shows a cause of interrupt indirectly.

When the EP0IntStat register has a cause of interrupt and the EP0IntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1.

D0 RcvEP0SETUP

Shows a cause of interrupt 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.

0x300901: SIE_IntStat (SIE Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
SIE_IntStat (SIE interrupt status)	00300901 (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 a cause of interrupt 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 a cause of interrupt 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 a cause of interrupt 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 a cause of interrupt directly.

Set to 1 when the suspend state of the USB is detected.

After detecting the USB suspend state, setting the USBSNZ bit of the USB Wait Control Register (0x300012) to 1 enables the IC to enter the snooze mode (to stop the built-in PLL oscillation).

D3 RcvSOF

Shows a cause of interrupt directly.

Set to 1 when the SOF token is received.

D2 DetectJ

Shows a cause of interrupt directly.

Set to 1 when the J-state is detected.

D1 Reserved**D0 SetAddressCmp**

Shows a cause of interrupt 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.

0x300902: EPrintStat (EPr Interrupt Status)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks	
EPrintStat (EPr interrupt status)	00300902 (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	

D[7:4] Reserved

D3 EPdIntStat

Shows a cause of interrupt indirectly.

When the EPdIntStat register has a cause of interrupt and the EPdIntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1.

D2 EPcIntStat

Shows a cause of interrupt indirectly.

When the EPcIntStat register has a cause of interrupt and the EPcIntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1.

D1 EPbIntStat

Shows a cause of interrupt indirectly.

When the EPbIntStat register has a cause of interrupt and the EPbIntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1.

D0 EPaIntStat

Shows a cause of interrupt indirectly.

When the EPaIntStat register has a cause of interrupt and the EPaIntEnb register bit corresponding to the cause of interrupt is enabled, this bit is set to 1.

0x300903: DMA_IntStat (DMA Interrupt Status)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_IntStat (DMA interrupt status)	00300903 (B)	D7-2	–		–	–	0 when being read.
		D1	DMA_CountUp	1 DMA counter overflow	0 None	0	R(W)
		D0	DMA_Cmp	1 DMA complete	0 None	0	R(W)

This register displays the interrupt status of the DMA.

D[7:2] Reserved**D1 DMA_CountUp**

Shows a cause of interrupt 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 a cause of interrupt directly.

Set to 1 when the DMA is stopped or completes the specified number of transfer operations and the end processing.

0x300904: FIFO_IntStat (FIFO Interrupt Status)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks
FIFO_IntStat (FIFO interrupt status)	00300904 (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 a cause of interrupt directly.

Set to 1 when as many data as specified in the DescSize 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 DescSize register.

D[6:2] Reserved**D1 FIFO_IN_Cmp**

Shows a cause of interrupt 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 a cause of interrupt 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.

0x300907: EP0IntStat (EP0 Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EP0IntStat (EP0 interrupt status)	00300907 (B)	D7-6	–			–	–	0 when being read.	
		D5	IN_TrانACK	1	In transaction ACK	0	None	0	R(W)
		D4	OUT_TrانACK	1	Out transaction ACK	0	None	0	R(W)
		D3	IN_TrانNAK	1	In transaction NAK	0	None	0	R(W)
		D2	OUT_TrانNAK	1	Out transaction NAK	0	None	0	R(W)
		D1	IN_TrانErr	1	In transaction error	0	None	0	R(W)
		D0	OUT_TrانErr	1	Out transaction error	0	None	0	R(W)

This register displays the interrupt status of the endpoint EP0.

D[7:6] Reserved**D5 IN_TrانACK**

Shows a cause of interrupt directly.
Set to 1 when ACK is received in the IN transaction.

D4 OUT_TrانACK

Shows a cause of interrupt directly.
Set to 1 when ACK is replied in the OUT transaction.

D3 IN_TrانNAK

Shows a cause of interrupt directly.
Set to 1 when NAK is replied in the IN transaction.

D2 OUT_TrانNAK

Shows a cause of interrupt directly.
Set to 1 when NAK is replied in the OUT transaction.

D1 IN_TrانErr

Shows a cause of interrupt 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_TrانErr

Shows a cause of interrupt directly.
Set to 1 when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x300908: EPaintStat (EPa Interrupt Status)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks	
EPaintStat (EPa interrupt status)	00300908 (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 a cause of interrupt 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 a cause of interrupt directly.

Set to 1 when ACK is received in the IN transaction.

D4 OUT_TranACK

Shows a cause of interrupt directly.

Set to 1 when ACK is replied in the OUT transaction.

D3 IN_TranNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the IN transaction.

D2 OUT_TranNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the OUT transaction.

D1 IN_TranErr

Shows a cause of interrupt 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 a cause of interrupt directly.

Set to 1 when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x300909: EPbIntStat (EPb Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPbIntStat (EPb interrupt status)	00300909 (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 a cause of interrupt 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 a cause of interrupt directly.

Set to 1 when ACK is received in the IN transaction.

D4 OUT_TranACK

Shows a cause of interrupt directly.

Set to 1 when ACK is replied in the OUT transaction.

D3 IN_TranNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the IN transaction.

D2 OUT_TranNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the OUT transaction.

D1 IN_TranErr

Shows a cause of interrupt 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 a cause of interrupt directly.

Set to 1 when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x30090A: EPcIntStat (EPc Interrupt Status)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks	
EPcIntStat (EPc interrupt status)	0030090A (B)	D7	–				–	–	0 when being read.	
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)	
		D5	IN_TrانACK	1	In transaction ACK	0	None	0	R(W)	
		D4	OUT_TrانACK	1	Out transaction ACK	0	None	0	R(W)	
		D3	IN_TrانNAK	1	In transaction NAK	0	None	0	R(W)	
		D2	OUT_TrانNAK	1	Out transaction NAK	0	None	0	R(W)	
		D1	IN_TrانErr	1	In transaction error	0	None	0	R(W)	
		D0	OUT_TrانErr	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 a cause of interrupt directly.

Set to 1 when a short packet is received and ACK is replied in the OUT transaction, OUT_TrانACK and this bits are set to 1 at the same time.

D5 IN_TrانACK

Shows a cause of interrupt directly.

Set to 1 when ACK is received in the IN transaction.

D4 OUT_TrانACK

Shows a cause of interrupt directly.

Set to 1 when ACK is replied in the OUT transaction.

D3 IN_TrانNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the IN transaction.

D2 OUT_TrانNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the OUT transaction.

D1 IN_TrانErr

Shows a cause of interrupt 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_TrانErr

Shows a cause of interrupt directly.

Set to 1 when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x30090B: EPdIntStat (EPd Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPdIntStat (EPd interrupt status)	0030090B (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 a cause of interrupt 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 a cause of interrupt directly.

Set to 1 when ACK is received in the IN transaction.

D4 OUT_TranACK

Shows a cause of interrupt directly.

Set to 1 when ACK is replied in the OUT transaction.

D3 IN_TranNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the IN transaction.

D2 OUT_TranNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the OUT transaction.

D1 IN_TranErr

Shows a cause of interrupt 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 a cause of interrupt directly.

Set to 1 when STALL is replied in the OUT transaction or when an error occurred in the packet.

0x300910: MainIntEnb (Main Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
MainIntEnb (Main interrupt enable)	00300910 (B)	D7	EnSIE_IntStat	1	Enabled	0	Disabled	0	R/W	
		D6	EnEPrintStat					0	R/W	
		D5	EnDMA_IntStat					0	R/W	
		D4	EnFIFO_IntStat					0	R/W	
		D3-2	—	—		—	—	0 when being read.		
		D1	EnEP0IntStat	1	Enabled	0	Disabled	0	R/W	
		D0	EnRcvEP0SETUP					0	R/W	

This register enables/disables assertion of the interrupt signal (#INT) with the cause of interrupt of the MainIntStat register. Setting the corresponding bit to 1 enables interrupt. EnSIE_IntStat bit is valid during snooze as well.

0x300911: SIE_IntEnb (SIE Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks				
SIE_IntEnb (SIE interrupt enable)	00300911 (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

This register enables/disables assertion of the SIE_IntStat bit of the MainIntStat register with the cause of interrupt of the SIE_IntStat register. EnVBUS_Changed and EnNonJ bits are valid during snooze as well.

0x300912: EPrintEnb (EPr Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPrintEnb (EPr interrupt enable)	00300912 (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 cause of interrupt of the EPrIntStat register.

0x300913: DMA_IntEnb (DMA Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks			
DMA_IntEnb (DMA interrupt enable)	00300913 (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 cause of interrupt of the DMA_IntStat register.

0x300914: FIFO_IntEnb (FIFO Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
FIFO_IntEnb (FIFO interrupt enable)	00300914 (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 cause of interrupt of the FIFO_IntStat register.

0x300917: EP0IntEnb (EP0 Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EP0IntEnb (EP0 interrupt enable)	00300917 (B)	D7-6	–	–		–	–	0 when being read.
		D5	EnIN_TrانACK	1 Enabled	0 Disabled	0	R/W	
		D4	EnOUT_TrانACK			0	R/W	
		D3	EnIN_TrانNAK			0	R/W	
		D2	EnOUT_TrانNAK			0	R/W	
		D1	EnIN_TrانErr			0	R/W	
		D0	EnOUT_TrانErr			0	R/W	

This register enables/disables assertion of the EP0IntStat bit of the MainIntStat register with the cause of interrupt of the EP0IntStat register.

0x300918: EPaIntEnb (EPa Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPaIntEnb (EPa interrupt enable)	00300918 (B)	D7	–	–		–	–	0 when being read.
		D6	EnOUT_ShortACK	1	Enabled	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 cause of interrupt of the EPaIntStat register.

0x300919: EPbIntEnb (EPb Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPbIntEnb (EPb interrupt enable)	00300919 (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 cause of interrupt of the EPbIntStat register.

0x30091A: EPcIntEnb (EPc Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPcIntEnb (EPc interrupt enable)	0030091A (B)	D7	–	–		–	–	0 when being read.
		D6	EnOUT_ShortACK	1	Enabled	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 cause of interrupt of the EPcIntStat register.

0x30091B: EPdIntEnb (EPd Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPdIntEnb (EPd interrupt enable)	0030091B (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 cause of interrupt of the EPdIntStat register.

0x300920: RevisionNum (Revision Number)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
RevisionNum (Revision number)	00300920 (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.

0x300921: USB_Control (USB Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
USB_Control (USB control register)	00300921 (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 cause of interrupt (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.

D[2:1] 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.

0x300922: USB_Status (USB Status)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks	
USB_Status (USB status register)	00300922 (B)	D7	VBUS	1	VBUS=High	0	VBUS=Low	X	R	
		D6	FS	1	FS mode (fixed)	0	–	1	R	
		D5–2	–	–	–	–	–	–	–	0 when being read.
		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).

D[5:2] Reserved**D[1:0] 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

0x300923: XcvrControl (Xcvr Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
XcvrControl (Xcvr control register)	00300923 (B)	D7	RpuEnb	1	Enable pull-up	0	Disable pull-up	0	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.

D[6:2] Reserved**D[1:0] 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 0x01.

0x300924: USB_Test (USB Test)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
USB_Test (USB test)	00300924 (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.

D[6:4] 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.

- 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.
- Do not overlap the above setting with the settings of the endpoints EPa and EPb.
Or clear the EPaConfig_0.EnEndPoint bit and EPbConfig_0.EnEndPoint bit.
- Clear the FIFO of the EPc and write data for the following test packet into this FIFO.
- 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.)

0x300925: EPnControl (Endpoint Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks
EPnControl (Endpoint control)	00300925 (B)	D7	AllForceNAK	1	Set all ForceNAK	0	Do nothing	0 when being read.
		D6	EPPrForceSTALL	1	Set EP's ForceSTALL	0	Do nothing	
		D5	AllFIFO_Clr	1	Clear all FIFO	0	Do nothing	
		D4-1	–			–	–	
		D0	EP0FIFO_Clr	1	Clear EP0 FIFO	0	Do nothing	

This register sets operations of entire endpoints, and display them.

D7 AllForceNAK

Set the ForceNAK bit of all endpoints to 1.

D6 EPPrForceSTALL

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.

D[4:1] 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.

0x300926: EPrFIFO_Clr (EPr FIFO Clear)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPrFIFO_Clr (EPr FIFO clear)	00300926 (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 clears the FIFO of the endpoints.

D[7:4] 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 occur.

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 is 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 occur.

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 occur.

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 occur.

0x30092E: FrameNumber_H (Frame Number HIGH)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
FrameNumber_H (Frame number high)	0030092E (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.

D[6:3] Reserved**D[2:0] FrameNumber[10:8]**

The upper order 3 bits in the FrameNumber field of the received SOF packet are stored in these bits.

0x30092F: FrameNumber_L (Frame Number LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
FrameNumber_L (Frame number low)	0030092F (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		

D[7:0] FrameNumber[7:0]

The lower order 8 bits in the FrameNumber field of the received SOF packet are stored in these bits.

0x300930–0x300937: EP0Setup_0 (EP0 Setup 0)–EP0Setup_7 (EP0 Setup 7)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EP0Setup_0 (EP0 set-up 0)	00300930	D7	EP0Setup_n[7]	Endpoint 0 set-up data 0	0	R	
		D6	EP0Setup_n[6]		0		
	00300937	D5	EP0Setup_n[5]	Endpoint 0 set-up data 7	0		
EP0Setup_7 (EP0 set-up 7)	(B)	D4	EP0Setup_n[4]		0		
		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.

0x300930: EP0Setup_0

BmRequestType is set.

0x300931: EP0Setup_1

BRequest is set.

0x300932: EP0Setup_2

The lower order 8 bits in Wvalue are set.

0x300933: EP0Setup_3

The upper order 8 bits in Wvalue are set.

0x300934: EP0Setup_4

The lower order 8 bits in WIndex are set.

0x300935: EP0Setup_5

The upper order 8 bits in WIndex are set.

0x300936: EP0Setup_6

The lower order 8 bits in WLength are set.

0x300937: EP0Setup_7

The upper order 8 bits in WLength are set.

0x300938: USB_Address (USB Address)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
USB_Address (USB address)	00300938 (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 RcvEPOSETUP bit of the MainIntStat register is set to 1. Read the EPOSetup_0–7 registers and interpret the request.

- (2) Set the AutoSetAddress bit.
- (3) Set the INxOUT bit of the EPOControl register.
- (4) Clear the ForceNAK bit of the EPOControlIN register, and set the EnShortPkt bit.
- (5) Wait for the end of the status stage.

The SetAddressCmp bit of the SIE_IntStat register is set to 1.

D[6:0] USB_Address

These bits set up the USB address.

The USB address is written automatically by the AutoSetAddress function. Or it can be written.

0x300939: EP0Control (EP0 Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EP0Control (EP0 control)	00300939 (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.

D[6:1] 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_TransACK bit of the EPnIntStat register is set to 1.

Refer to the section describing operations, for details.

0x30093A: EP0ControlIN (EP0 Control IN)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EP0ControlIN (EP0 control IN)	0030093A (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 RcvEPOSETUP 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 RcvEPOSETUP 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 RcvEPOSETUP 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 RcvEPOSETUP 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.

0x30093B: EP0ControlOUT (EP0 Control OUT)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EP0ControlOUT (EP0 control OUT)	0030093B (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.

D[6:5] 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 RcvEPOSETUP 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 RcvEPOSETUP 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 RcvEPOSETUP 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 RcvEPOSETUP 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.

0x30093F: EP0MaxSize (EP0 Max Packet Size)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EP0MaxSize (EP0 max packet size)	0030093F (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.		

D7 **Reserved**

D[6:3] **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.

D[2:0] **Reserved**

0x300940: EPaControl (EPa Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks			
EPaControl (EPa control)	00300940 (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.

0x300941: EPbControl (EPb Control)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks		
EPbControl (EPb control)	00300941 (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.

0x300942: EPcControl (EPc Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks			
EPcControl (EPc control)	00300942 (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.

0x300943: EPdControl (EPd Control)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks		
EPdControl (EPd control)	00300943 (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.

0x300950: EPaMaxSize_H (EPa Max Packet Size HIGH)**0x300951: EPaMaxSize_L (EPa Max Packet Size LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPaMaxSize_H (EPa max packet size high)	00300950 (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)	00300951 (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.

0x300952: EPaConfig_0 (EPa Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPaConfig_0 (EPa configuration 0)	00300952 (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	–	–	–	–	–	
		D3	EndPointNumber[3]	Endpoint number		0	R/W	
		D2	EndPointNumber[2]	(0x1 to 0xF)		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**D[3:0] EndPointNumber**

Set an endpoint number between 0x1 and 0xF.

0x300953: EPaConfig_1 (EPa Configuration 1)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPaConfig_1 (EPa configuration 1)	00300953 (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	–			–		–	–

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 occurs 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.

D[5:0] Reserved

0x300954: EPbMaxSize_H (EPb Max Packet Size HIGH)**0x300955: EPbMaxSize_L (EPb Max Packet Size LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPbMaxSize_H (EPb max packet size high)	00300954 (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)	00300955 (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.

0x300956: EPbConfig_0 (EPb Configuration 0)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EPbConfig_0 (EPb configuration 0)	00300956 (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		0	R/W			
		D2	EndPointNumber[2]	(0x1 to 0xF)		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**D[3:0] EndPointNumber**

Set an endpoint number between 0x1 and 0xF.

0x300957: EPbConfig_1 (EPb Configuration 1)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks	
EPbConfig_1 (EPb configuration 1)	00300957 (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	--						--	--

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 occurs 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.

D[5:0] Reserved

0x300958: EPcMaxSize_H (EPc Max Packet Size HIGH)**0x300959: EPcMaxSize_L (EPc Max Packet Size LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPcMaxSize_H (EPc max packet size high)	00300958 (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)	00300959 (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.

0x30095A: EPcConfig_0 (EPc Configuration 0)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EPcConfig_0 (EPc configuration 0)	0030095A (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		0	R/W			
		D2	EndPointNumber[2]	(0x1 to 0xF)		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**D[3:0] EndPointNumber**

Set an endpoint number between 0x1 and 0xF.

0x30095B: EPcConfig_1 (EPc Configuration 1)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPcConfig_1 (EPc configuration 1)	0030095B (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	–			–		–	–

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 occurs 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.

D[5:0] Reserved

0x30095C: EPdMaxSize_H (EPd Max Packet Size HIGH)**0x30095D: EPdMaxSize_L (EPd Max Packet Size LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPdMaxSize_H (EPd max packet size high)	0030095C (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)	0030095D (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.

0x30095E: EPdConfig_0 (EPd Configuration 0)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EPdConfig_0 (EPd configuration 0)	0030095E (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		0	R/W			
		D2	EndPointNumber[2]	(0x1 to 0xF)		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**D[3:0] EndPointNumber**

Set an endpoint number between 0x1 and 0xF.

0x30095F: EPdConfig_1 (EPd Configuration 1)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks	
EPdConfig_1 (EPd configuration 1)	0030095F (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	--						--	--

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 occurs 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.

D[5:0] Reserved

0x300970: EPaStartAdrs_H (EPa FIFO Start Address HIGH)**0x300971: EPaStartAdrs_L (EPa FIFO Start Address LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPaStartAdrs_H (EPa FIFO start address high)	00300970 (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)	00300971 (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.

0x300972: EPbStartAdrs_H (EPb FIFO Start Address HIGH)**0x300973: EPbStartAdrs_L (EPb FIFO Start Address LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPbStartAdrs_H (EPb FIFO start address high)	00300972 (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)	00300973 (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.

0x300974: EPcStartAdrs_H (EPc FIFO Start Address HIGH)**0x300975: EPcStartAdrs_L (EPc FIFO Start Address LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPcStartAdrs_H (EPc FIFO start address high)	00300974 (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)	00300975 (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.

0x300976: EPdStartAdrs_H (EPd FIFO Start Address HIGH)**0x300977: EPdStartAdrs_L (EPd FIFO Start Address LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPdStartAdrs_H (EPd FIFO start address high)	00300976 (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)	00300977 (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.

0x300980: CPU_JoinRd (CPU Join FIFO Read)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
CPU_JoinRd (CPU join FIFO read)	00300980 (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.

D[7:4] 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.

0x300981: CPU_JoinWr (CPU Join FIFO Write)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks	
CPU_JoinWr (CPU join FIFO write)	00300981 (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 written. 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 writing 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.

D[7:4] 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.

0x300982: EnEPnFIFO_Access (Enable EPn FIFO Access)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks
EnEPnFIFO_Access (Enable EPn FIFO access)	00300982 (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.

D[7:2] 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.

0x300983: EPnFIFOforCPU (EPn FIFO for CPU)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnFIFOforCPU (EPn FIFO for CPU)	00300983 (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		

D[7:0] EPnFIFOData[7:0]

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.

0x300984: EPnRdRemain_H (EPn FIFO Read Remain HIGH)**0x300985: EPnRdRemain_L (EPn FIFO Read Remain LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnRdRemain_H (EPn FIFO read remain high)	00300984 (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)	00300985 (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		

EPnRdRemain[11:0]

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.

0x300986: EPnWrRemain_H (EPn FIFO Write Remain HIGH)**0x300987: EPnWrRemain_L (EPn FIFO Write Remain LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnWrRemain_H (EPn FIFO write remain high)	00300986 (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)	00300987 (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		

EPnWrRemain[11:0]

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.

0x300988: DescAdrs_H (Descriptor Address HIGH)**0x300989: DescAdrs_L (Descriptor Address LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescAdrs_H (Descriptor address high)	00300988 (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		
DescAdrs_L (Descriptor address low)	00300989 (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		

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 Descriptor 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 EP0Control 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.

0x30098A: DescSize_H (Descriptor Size HIGH)**0x30098B: DescSize_L (Descriptor Size LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescSize_H (Descriptor size high)	0030098A (B)	D7-2	–	–	–	–	0 when being read.
		D1	DescSize[9]	Descriptor size	0	R/W	
		D0	DescSize[8]		0		
DescSize_L (Descriptor size low)	0030098B (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		

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 EP0Control 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.

0x30098F: DescDoor (Descriptor Door)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescDoor (Descriptor door)	0030098F (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		

D[7:0] DescMode[7:0]

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.

0x300990: DMA_FIFO_Control (DMA FIFO Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
DMA_FIFO_Control (DMA FIFO control)	00300990 (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.

D[5:0] Reserved

0x300991: DMA_Join (DMA Join FIFO)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks
DMA_Join (DMA join FIFO)	00300991 (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.

D[7:4] Reserved

D[3:0] JoinEPdDMA, JoinEPcDMA, JoinEPbDMA, JoinEPaDMA

When this bit is set to 1, the DMA transfer is enabled through the endpoint EP x ($x=a,b,c,d$). 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 EP x ($x=a,b,c,d$) by the DMA_Remain H, L register, is enabled.

0x300992: DMA_Control (DMA Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
DMA_Control (DMA control)	(B)	D7	DMA_Running	1	DMA is running	0	DMA is not running	0	R	0 when being read.
		D6	PDREQ		PDREQ signal logic		0	R		
		D5	PDACK		PDACK signal logic		0	R		
		D4	–		–		–	–		
		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 signal for monitoring.

D5 PDACK

Shows the logic level of the PDACK 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 DMA_IntStat 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.

0x300994: DMA_Config_0 (DMA Configuration 0)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
DMA_Config_0 (DMA configuration 0)	00300994 (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	–	–	–	–	–	–	–

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.

D[6:4] Reserved**D3 PDREQ_Level**

Set the PDREQ logic level. Set to 0 (active-high).

D2 PDACK_Level

Set the PDACK logic level. Set to 0 (active-high).

D1 PDRDWR_Level

Set the logic levels of the PDRD and PDWR signals. Set to 0 (active-high).

D0 Reserved

0x300995: DMA_Config_1 (DMA Configuration 1)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
DMA_Config_1 (DMA configuration 1)	00300995 (B)	D7	RcvLimitMode	1 Receive 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 signal.

In this mode, the PDREQ signal is negated when the space of the endpoint becomes less than 32 bytes by the DMA write operation. However, when the PDREQ 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 signal is negated, is 16 bytes or less.

In this mode, the PDREQ signal is negated before the endpoint becomes completely full.

When the area of the endpoint set by the EP{a,b,c,d}StartAdrs 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}StartAdrs register.

D[6:4] Reserved**D3 SingleWord**

Sets the handshake mode in the Asynchronous (handshake) mode.

In the Single Word mode, the PDREQ signal is negated every time when one word is transferred.

In the Multi-Word mode, the PDREQ signal is not negated if the next data communication is possible when one word is transferred.

D[2:1] 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.

0x300997: DMA_Latency (DMA Latency)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Latency (DMA latency)	00300997 (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.

D[7:4] Reserved

D[3:0] DMA_Latency[3:0]

If a value between 0x1 and 0xF is written, the PDREQ 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 signal is not be asserted for $(130 \times N)$ ns period.

0x300998: DMA_Remain_H (DMA FIFO Remain HIGH)**0x300999: DMA_Remain_L (DMA FIFO Remain LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Remain_H (DMA FIFO remain high)	00300998 (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)	00300999 (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_Remain[11:0]

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.

0x30099C: DMA_Count_HH (DMA Transfer Byte Counter HIGH/HIGH)**0x30099D: DMA_Count_HL (DMA Transfer Byte Counter HIGH/LOW)****0x30099E: DMA_Count_LH (DMA Transfer Byte Counter LOW/HIGH)****0x30099F: 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)	0030099C (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)	0030099D (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)	0030099E (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)	0030099F (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		

DMA_Count[31:0]

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 DMA_IntStat 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.

S1C33E08 Technical Manual

X PERIPHERAL MODULES 8 (MP3)

X.1 MP3 Decoder (MP3)

X.1.1 Overview

The S1C33E08 incorporates an MP3 decoder module for decoding MP3 data and playback of the decoded data through the I²S module. The MP3 decoder module consists of a hardware MP3 accelerator and an MP3 decoder BIOS located in the specific ROM (Area 1). The MP3 decoder BIOS provides high-level MP3 API functions to realize an MP3 player and low-level MP3 API functions to realizes an MP3 decoder using the hardware MP3 accelerator. The high-level MP3 API functions include codes for decoding MP3 data using the low-level MP3 API functions and for playback using the I²S module allowing the programmer to implement an MP3 player easily by creating MP3 data input and event processing functions. When creating a user-specific MP3 decoder/player program, the low-level MP3 API functions may be directly used to calculate equations for decoding MP3 data.

Note: The EPSON S1C33E08 incorporates MP3 technology of which Thomson SA in France holds the patent. Manufacturers using the EPSON S1C33E08 to develop MP3 products must pay royalties to Thomson SA in order to procure the license for the MP3 technology.

The following shows the MP3 and I²S specifications supported in the MP3 decoder BIOS:

MP3 data format supported

Sampling frequency:

32, 44.1, and 48 kHz (MPEG1 Audio Layer-III)
16, 22.05, and 24 kHz (MPEG2 Audio Layer-III LSF)

* The MP3 decoder BIOS does not support MPEG2.5 (8, 11.025, and 12 kHz) data with the Fraunhofer-Gesellschaft expanded.

The higher the sampling frequency, the more the CPU occupancy rate during playback increases. The CPU clock frequency should be set as high as possible to perform parallel processing using the RTOS, etc.

Bit-rate:

32 to 320 kbps (MPEG1 Audio Layer-III, CBR/VBR/ABR)
8 to 160 kbps (MPEG2 Audio Layer-III LSF, CBR/VBR/ABR)

Channel mode:

Stereo (joint stereo and dual channel) and monaural

I²S output specifications

Quality mode:

Three quality modes (High, Middle, and Low) available

* The output sound quality depends on the quality mode selected, note, however, that higher quality modes increase the CPU occupancy rate.

PCM data sampling frequency:

Automatically switched according to the MP3 data for playback.

MP3 decoder API functions

High-level MP3 API functions

The following control functions are provided:

- Open MP3 decoder (Open)
- Close MP3 decoder (Close)
- Start playback (Start)
- Stop playback (Stop)
- Pause playback (Pause)
- Resume playback (Resume) For playback from pause
- Synchronize and playback (Resync) For forward and rewind

Low-level MP3 API functions

The following calculation functions are provided:

- Anti alias processing (AntiAlias)
- Inverse discrete cosine transform (IdctLong, IdctShort)
- Window function for inverse-MDCT (IdctWindow)
- Stereo processing (StereoProcess, StereoProcessLSF)
- Fast-DCT for sub-band synthesis (SubbandSynthesisFdct)
- Window function for sub-band synthesis (SubbandSynthesisWindow/2/4/8)

X.1.2 Composition of MP3 Decoder System

The S1C33E08 provides a hardware MP3 accelerator and an MP3 decoder BIOS to implement an MP3 decoder/player with an I²S interface and HSDMA for controlling data stream.

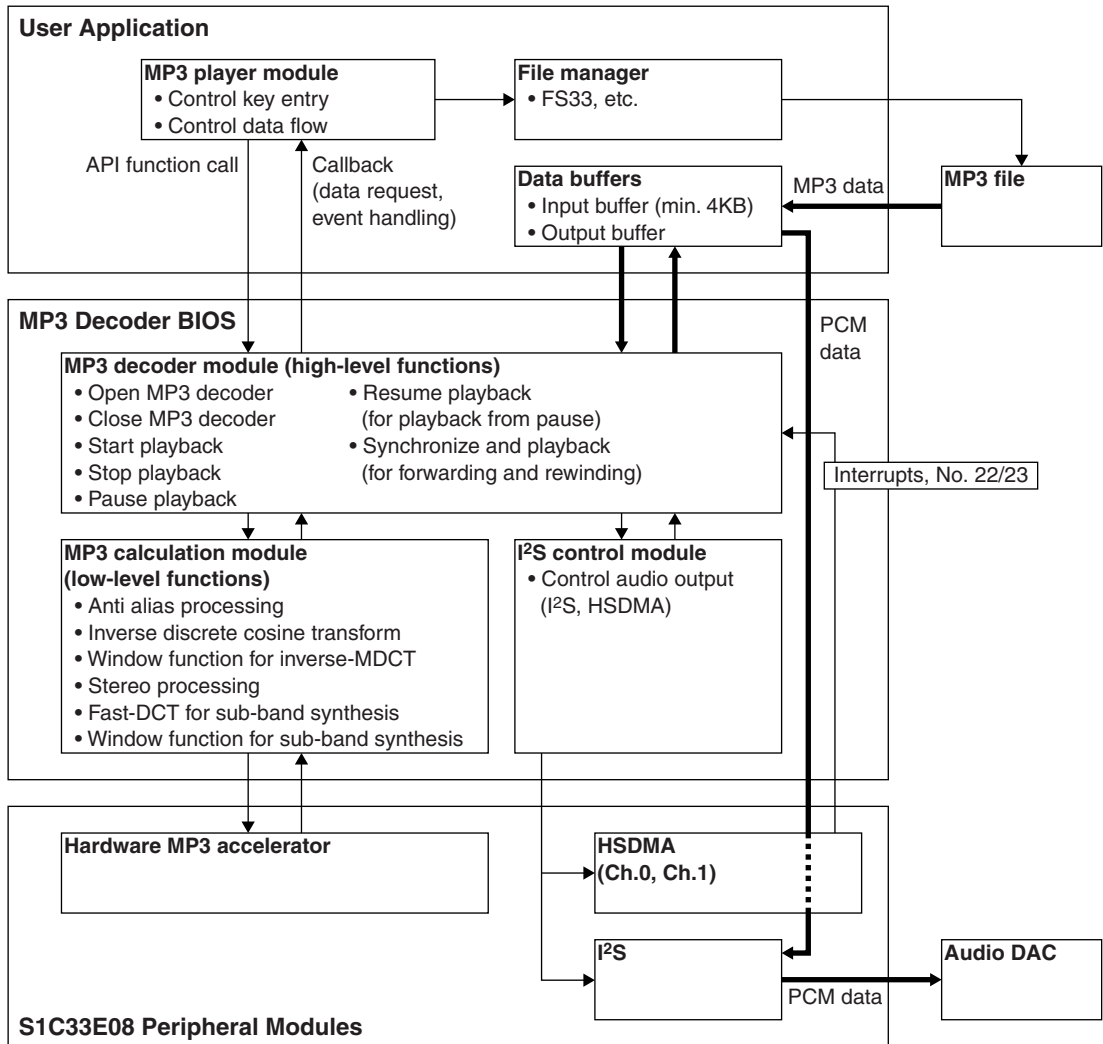


Figure X.1.2.1 Structure of MP3 Decoding System 1 (when MP3 decoder module is used)

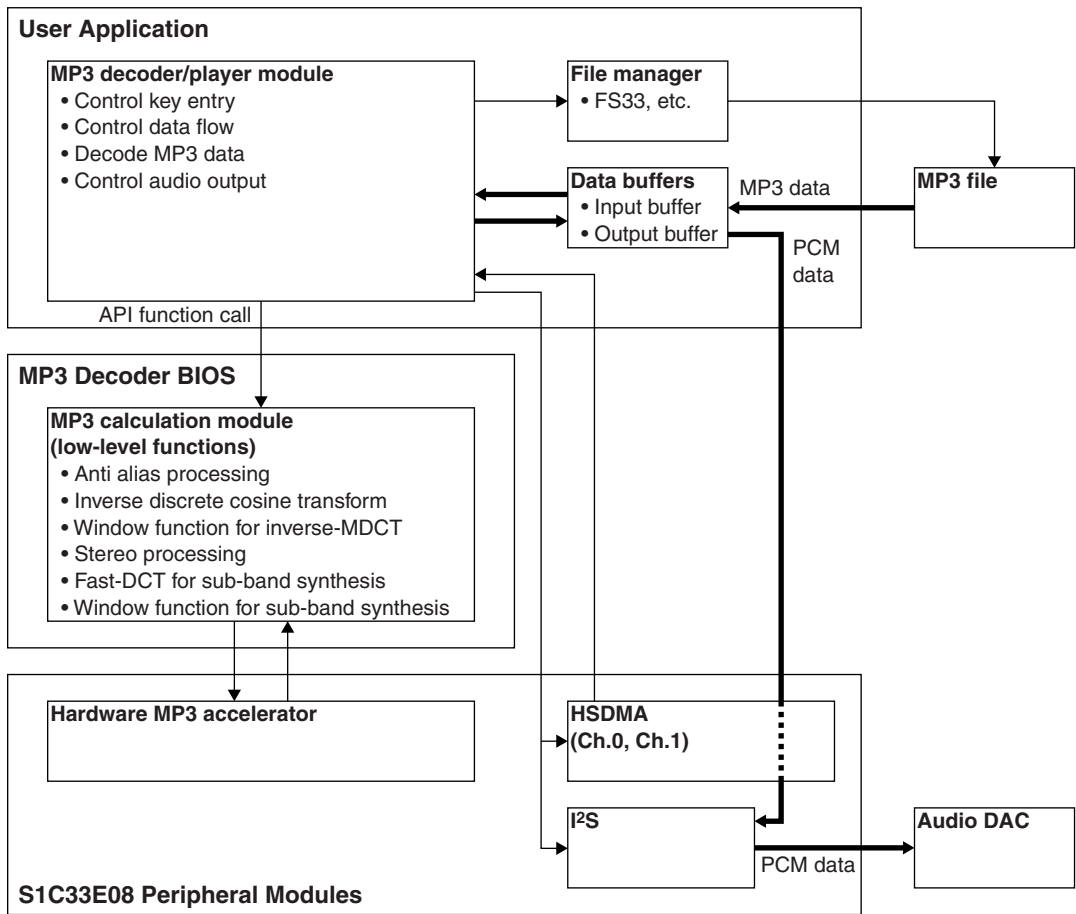


Figure X.1.2.2 Structure of MP3 Decoding System 2 (when MP3 calculation module only is used)

X.1.2.1 MP3 Decoder BIOS

The S1C33E08 contains an MP3 decoder BIOS that provides APIs to realize an MP3 decoder system. The BIOS consists of three modules shown below and located in Area 1.

(1) MP3 decoder module

This module provides high-level MP3 API functions that allow the user to implement an MP3 player easily. The module decodes MP3 data using the MP3 calculation module (2) and sends the decoded data to an external audio DAC via the I²S module using the I²S control module (3).

(2) MP3 calculation module

This module provides low-level MP3 API functions used for calculations to decode MP3 data. When using the MP3 decoder module (1) to implement an MP3 player, the user program does not need to call the low-level functions because they are called from the MP3 decoder module (1). When implementing an MP3 player with a user original control program, call the low-level functions directly from the user program to decode MP3 data.

(3) I²S control module

This module is positioned under the MP3 decoder module (1) and it controls the I²S module and HSDMA to send the decoded data to the external audio DAC connected via the I²S interface. The module will not be used when the MP3 decoder module (1) is not used.

X.1.2.2 Hardware Resource Requirements

Table X.1.2.2.1 lists the memory and internal peripheral modules required for an MP3 decoding system.

Table X.1.2.2.1 Hardware Resource Requirements

Resource	When MP3 decoder module is used	When MP3 calculation module only is used
AORAM	7KB (0x0400–0x1FFF) *1	–
IVRAM	12KB (0x2000–0x4FFF) *1	–
External RAM	20KB min. *2	–
Stack area	40 bytes *3	40 bytes *3
I ² S module	Used	–
HSDMA module	Ch.0 and Ch.1 *4	–
ITC module	Vector No. 22, 23 and 94 *4	–

- *1: The MP3 decoder module (high-level MP3 API functions) occupies 19K bytes of the internal RAM (0x0400–0x4FFF) in Area 0 including the IVRAM. The remaining 1K bytes of internal RAM (0x0000–0x03FF) is available for the user program. Note that an external SDRAM must be used as the VRAM for the LCD controller.
- *2: A 16K-byte work area and at least 4K bytes of MP3 input data buffer are required for the MP3 decoder module (high-level MP3 API functions) in addition to the internal RAM. The 16K-byte work area can be located in an external RAM from any given address in word alignment. The input data buffer can be located anywhere independently of the work area, note, however, that the buffer size must be a multiple of eight bytes and the start address must be aligned on a word boundary.
- *3: The MP3 calculation module uses 40 bytes in the stack configured in the user application.
- *4: The MP3 decoder module (I²S control module) uses HSDMA Ch.0 and Ch.1 to fill the I²S FIFO with the PCM data to be output and their interrupts to control data transfer.

Note: RSTONLY (D0/0x30029F), which is used to select a cause-of-interrupt flag reset method, must be set to 1 (reset-only method) when the MP3 decoder function is used.

* **RSTONLY:** Cause-of-Interrupt Flag Reset Method Select Bit in the Flag Set/Reset Method Select Register (D0/0x30029F)

Figure X.1.2.2.1 shows the memory maps according to the MP3 module used.

(1) When MP3 decoder module is used

External RAM	
InputBufferAddr + (0xFFFF)	MP3 input buffer
InputBufferAddr	(4KB min.)
WorkAreaAddr + 0x3FFF	MP3 work area (16KB)
WorkAreaAddr	

Internal Areas		
Area 6	0x003F FFFF	(Reserved)
	0x0030 2000	IP and peripherals
	0x0030 1FFF	
	0x0030 0000	(Reserved)
Area 3	0x000F FFFF	
	0x0008 4800	DST RAM (2KB)
	0x0008 47FF	
	0x0008 4000	(Reserved)
	0x0008 3FFF	
	0x0008 3000	
	0x0008 2FFF	Reserved for debugging
Area 2	0x0007 FFFF	
	0x0006 0000	(Reserved)
Area 1	0x0005 FFFF	
	0x0003 0000	Specific ROM
	0x0002 FFFF	
	0x0002 0000	(Reserved for firmware)
Area 0	0x0001 FFFF	
	0x0000 8000	MP3 work area (19KB)
	0x0000 4FFF	
	0x0000 0400	IVRAM (12KB)
	0x0000 03FF	
	0x0000 0000	AORAM (1KB)

(2) When MP3 calculation module only is used

External RAM		
Depends on the user application.		
Internal Areas		
Area 6	0x003F FFFF	(Reserved)
	0x0030 2000	IP and peripherals
	0x0030 1FFF	
	0x0030 0000	(Reserved)
Area 3	0x000F FFFF	
	0x0008 4800	DST RAM (2KB)
	0x0008 47FF	
	0x0008 4000	(Reserved)
	0x0008 3FFF	
	0x0008 3000	
	0x0008 2FFF	IVRAM (12KB)
Area 2	0x0007 FFFF	
	0x0006 0000	Reserved for debugging
Area 1	0x0005 FFFF	
	0x0003 0000	(Reserved)
	0x0002 FFFF	
	0x0002 0000	Specific ROM
	0x0001 FFFF	
	0x0001 FFFF	(Reserved for firmware)
Area 0	0x0001 FFFF	
	0x0000 8000	AORAM (12KB)
	0x0000 4FFF	
	0x0000 2000	AORAM (8KB)
	0x0000 1FFF	
	0x0000 0000	

□ Can be accessed from the user program
 ■ Cannot be accessed from the user program

Figure X.1.2.2.1 Memory Map

Refer to Section X.1.6, “Performance,” for the CPU sharing time.

X.1.3 Usage of MP3 API Functions

The MP3 decoder BIOS provides the API functions that can be called from the user program to realize an MP3 decoder/player.

The API functions are divided into two categories, high-level MP3 API functions and low-level MP3 API functions. Use the high-level MP3 API functions to realize an MP3 player easily as well as the MP3 decoder. The low-level MP3 API functions provide an MP3 decoding feature only, so use them when creating a user's MP3 player module.

X.1.3.1 High-Level MP3 API Functions

The high-level MP3 API functions allow the user to implement an MP3 player easily.

List of high-level MP3 API functions

Table X.1.3.1.1 lists the high-level MP3 API functions and their start address.

Table X.1.3.1.1 List of High-level MP3 API Functions

Address	API function	Description
0x0002_3000	c33MP3DecOpen()	Opens MP3 decoder.
0x0002_3004	c33MP3DecClose()	Closes MP3 decoder.
0x0002_3008	c33MP3DecStart()	Starts playback of MP3 data.
0x0002_300C	c33MP3DecStop()	Stops playback of MP3 data.
0x0002_3010	c33MP3DecPause()	Pauses playback of MP3 data.
0x0002_3014	c33MP3DecResume()	Resumes playback of paused MP3 data.
0x0002_3018	c33MP3DecResync()	Resynchronizes MP3 data for forwarding or rewinding.

- Notes:**
- The addresses may be changed in a BIOS update in the future.
 - Includes the “mp3.h” header file, which contains the prototype declarations for the functions, into the user application source file.

Control procedure

The following shows an MP3 decoder initialization and control procedure:

- (1) Enable the clock supply to the required peripheral modules.

Table X.1.3.1.2 lists the clocks that must be enabled.

Table X.1.3.1.2 Clock Control

Clock	Peripheral	Control bit
SDRAMC CPU APP clock (HALT mode) *1	SDRAMC	SDAPCPU_HCKE (D7/0x301B00)
SDRAMC CPU APP clock *1	SDRAMC	SDAPCPU_CKE (D6/0x301B00)
SDRAMC LCDC APP clock *1	SDRAMC	SDAPLCDC_CKE (D5/0x301B00)
SDRAMC SAPB bus interface clock *1	SDRAMC	SDSAPB_CKE (D4/0x301B00)
DST RAM clock *2	On-chip debugger	DSTRAM_CKE (D3/0x301B00)
LCDC SAPB bus interface clock	LCDC	LCDCSAPB_CKE (D1/0x301B00)
I ² S clock	I ² S	I2S_CKE (D11/0x301B04)
GPIO clock	GPIO	GPIO_CKE (D8/0x301B04)
ITC clock	ITC	ITC_CKE (D2/0x301B04)
DMA clock	HSDMA	DMA_CKE (D1/0x301B04)

*1: These clocks must be enabled when the MP3 work area/input buffer is located in an SDRAM.

*2: This clock must be enabled when debugging the program.

- (2) Define a `T_MP3DEC_PARAM` type variable and set the members.

The `T_MP3DEC_PARAM` structure is used as the argument of the `c33MP3DecOpen()` function to initialize the MP3 decoder.

The `T_MP3DEC_PARAM` structure is defined in the “mp3.h” header file. Include this file in the user program source file.

Definition of the `T_MP3DEC_PARAM` structure

```
typedef struct {
    unsigned long    WorkAreaAddr;
    unsigned long    InputBufferAddr;
    unsigned long    InputBufferSize;
    unsigned long    SystemClock;
    unsigned long    Quality;
    unsigned long    I2SConfig;
    unsigned long    (*TransferData)(void*, unsigned long);
    void             (*Event)(long);
} T_MP3DEC_PARAM;
```

WorkAreaAddr

Set the top address of the work area used by the MP3 decoder.

The work area needs a 16K-byte area in an external RAM. The address must be aligned on a word boundary.

InputBufferAddr

Set the top address of the MP3 input data buffer. The MP3 decoder BIOS gets MP3 data from this buffer.

The buffer needs an external RAM area of the size designated with `InputBufferSize`. The address must be aligned on a word boundary.

InputBufferSize

Set the size of the MP3 input data buffer.

The size should be 4K bytes or more in eight-byte increments.

SystemClock

Set the frequency [Hz] of the MCLK clock that is supplied from the CMU to the I²S module.

The practical I²S output clock is decided by this parameter.

$$I^2S \text{ output clock} = \frac{\text{System clock}}{\text{ROUND}\left(\frac{\text{System clock}}{\text{Sampling frequency} \times 64}\right) \times 64}$$

“*Sampling frequency*” is determined by the setting in the MP3 file to be decoded.

Table X.1.3.1.3 lists examples of the I²S output clock frequencies.

Table X.1.3.1.3 Examples of I²S Output Clock Frequencies

CPU clock	Sampling frequency					
	48 kHz	44.1 kHz	32 kHz	24 kHz	22.05 kHz	16 kHz
60 MHz	46.875 (+2.3%)	44.642 (-1.2%)	32.327 (-1.0%)	24.038 (-0.2%)	21.802 (+1.1%)	15.889 (+0.7%)
48 MHz	46.875 (+2.3%)	44.117 (-0.0%)	32.608 (-1.9%)	24.193 (-0.8%)	22.058 (-0.0%)	15.957 (+0.3%)
30 MHz	46.875 (+2.3%)	42.613 (+3.4%)	31.250 (+2.3%)	23.437 (+2.6%)	22.321 (-1.2%)	16.163 (-1.0%)
24 MHz	46.875 (+2.3%)	41.666 (+5.5%)	31.250 (+2.3%)	23.437 (+2.4%)	22.058 (-0.0%)	16.304 (-1.9%)
12 MHz	46.875 (+2.3%)	46.875 (-6.3%)	31.250 (+2.3%)	23.437 (+2.6%)	20.833 (+5.5%)	15.625 (+2.3%)
8 MHz	41.666 (+13.2%)	41.666 (+5.5%)	31.250 (+2.3%)	25.000 (-4.2%)	20.833 (+5.5%)	15.625 (+2.3%)

Values mean “Practical sampling frequency [kHz] (error[%])”

Quality

Specify the quality mode for playing back MP3 data using a constant shown below.

MP3DEC_MODE_H (0) High quality mode using high CPU power

MP3DEC_MODE_M (1) Middle quality mode using middle CPU power

MP3DEC_MODE_L (2) Low quality mode using low CPU power

I2SConfig

Set the I²S output configuration value to be written to the I²S Control Register (0x301C00).

Use the constants shown below. Two or more constants can be specified by separating each of them with an operator '|'.

MP3DEC_WCLKMD_L_HIGH	(0x8)	Left = High level
MP3DEC_BCLKPOL_NEGATIVE	(0x4)	Trigger = Negative Edge
MP3DEC_DTTMG_NOT_DELAYED	(0x2)	Not delay
MP3DEC_DTFORM_LSB	(0x1)	LSB first

Refer to the “I²S Interface” section for details of the I²S Control Register (0x301C00).

TransferData

Set the pointer to the user function to transfer MP3 data to the MP3 decoder.

Refer to “Callback functions” below or Section X.1.5.9, “TransferData (Callback Function),” for details.

Event

Set the pointer to the user function to handle events issued from the MP3 decoder BIOS.

When set to 0, no event will be issued.

Refer to “Callback functions” below or Section X.1.5.10, “Event (Callback Function),” for details.

- (3) Call the `c33MP3DecOpen()` function.

Example:

```
#include "mp3.h"

T_MP3DEC_PARAM param;

param.WorkAreaAddr = 0x10040000;
param.InputBufferAddr = 0x10046000;
param.InputBufferSize = 0x1000; // 4 [KB]
param.SystemClock = 48000000; // 48 [MHz]
param.Quality = MP3DEC_MODE_M;
param.I2SConfig = MP3DEC_BCLKPOL_NEGATIVE | MP3DEC_DTTMG_NOT_DELAYED;
param.TransferData = UserTransferFunc; // unsigned long UserTransferFunc(void*,
// unsigned long)
param.Event = UserEventFunc; // void UserEventFunc(long)

c33MP3DecOpen(&param);
```

The `c33MP3DecOpen()` function must be called to initialize the MP3 decoder and peripheral module settings before other API functions can be used. Do not call the `c33MP3DecOpen()` function twice or more before the MP3 decoder has been closed.

- (4) To start playback of MP3 data, call the `c33MP3DecStart()` function.

The `c33MP3DecStart()` function is used to start playback from the beginning of the MP3 data.

This function gets MP3 data into the input data buffer by calling the users data transfer function and decodes it using the MP3 calculate module, then outputs the decoded data via the I²S module. The playback sequence continues until the end of the MP3 data or a control function is called.

- (5) To stop playback, call the `c33MP3DecStop()` function.

This function stops data transfer to the MP3 decoder and clears buffer conditions. Therefore, a playback cannot be resumed from the stopped position after it is stopped with this function. Use the `c33MP3DecPause()` function to stop playback temporarily.

- (6) To pause playback, call the `c33MP3DecPause()` function.

This function stops data transfer to the MP3 decoder similar to the `c33MP3DecStop()` function. However, this function holds the buffer conditions allowing the program to resume playback from the paused position.

To resume playback, call the `c33MP3DecResume()` function. The `c33MP3DecStart()` function cannot be used for resuming.

- (7) To playback MP3 data from a desired position (or to restart playback after forwarding or rewinding the audio data), call the `c33MP3DecResync()` function.

This function synchronizes the decoder with the playback data using the sync word located in the MP3 frame header before starting playback. Therefore, the audio data in the buffer before the resynchronization has completed is not played back.

- (8) To finish MP3 decoding, call the `c33MP3DecClose()` function.

This function releases the resources used for the MP3 decoder.

Figure X.1.3.1.1 shows the MP3 decoder state transition diagram.

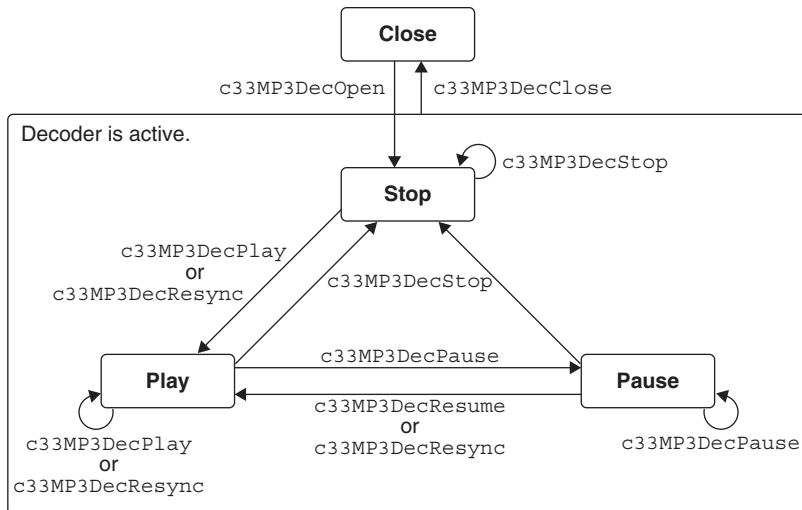


Figure X.1.3.1.1 MP3 Decoder State Transition Diagram

Callback functions

When using the high-level MP3 API functions to implement an MP3 player, the functions shown below must be created in the user program.

Data transfer function

The MP3 decoder BIOS gets MP3 data through the input data buffer of which the location and size are designated in the initial parameters (`T_MP3DEC_PARAM` structure). During playback, the MP3 decoder BIOS calls a data transfer function to fill the buffer with the MP3 data to be decoded when the input buffer becomes empty.

Create a data transfer function and set its pointer to the `TransferData` parameter in the `T_MP3DEC_PARAM` structure.

Example:

```

unsigned long UserTransferFunc(void*, unsigned long);
. . .
param.TransferData = UserTransferFunc;    // Set initial parameter
. . .
unsigned long UserTransferFunc(void *DstAddr, unsigned long Size){
    return fread(DstAddr, 1, Size, fin);  // Fill the input buffer
}
  
```

This function should fill the input buffer located from the address (`DstAddr`) with the specified size (`Size`) or less of MP3 data and return the size actually transferred. The pointer to the MP3 data to be transferred must be managed in the user program.

Event handler function

The MP3 decoder BIOS calls an event handler function when an error occurs during decoding if the pointer to the event handler function is set to the `Event` parameter in the `T_MP3DEC_PARAM` structure.

Create an event handler function as necessary.

If event handling is not necessary, set the `Event` parameter to 0. In this case, the MP3 decoder BIOS will not call an event handler function even if an error occurs during decoding.

The following shows the values that will be sent to the event handler function.

<code>DEC_FILE_END</code>	(10)	The MP3 data being decoded reached end of the file.
<code>DEC_ERR_CANT_FIND_SYNC</code>	(-1)	No sync-word can be found.
<code>DEC_ERR_INVALID_HEADER</code>	(-2)	The header contains invalid information.
<code>DEC_ERR_BAD_LAYER</code>	(-3)	The header contains information other than Layer III.

Example:

```
void UserEventFunc(long);

. . .

param.Event = UserEventFunc;

. . .

void UserEventFunc(long EventFlag){
    switch (EventFlag){
        case MP3DEC_FILE_END:
            // Go to next track.

            . . .

            break;

        case DEC_ERR_CANT_FIND_SYNC:
        case DEC_ERR_INVALID_HEADER:
        case DEC_ERR_BAD_LAYER:
            // Show error message "This is not MP3 format."

            . . .

            break;

        default:
            // Undefined event
            break;
    }
}
```

X.1.3.2 Low-Level MP3 API Functions

The low-level MP3 API functions provide an MP3 decoding feature. When creating an MP3 player module in the user program, use these functions for decoding MP3 data.

It is not necessary to call these functions when using the high-level MP3 API functions.

List of low-level MP3 API functions

Table X.1.3.2.1 lists the low-level MP3 API functions and their start addresses.

Table X.1.3.2.1 List of Low-level MP3 API Functions

Address	API function	Description
0x0002_3080	c33MP3CalcAntiAlias()	Performs butterfly calculation processing.
0x0002_3084	c33MP3CalcIdctLong()	Performs inverse MDCT calculation processing for long block.
0x0002_3088	c33MP3CalcIdctWindow()	Performs window calculation processing.
0x0002_308C	c33MP3CalcIdctShort()	Performs inverse MDCT calculation processing for short block.
0x0002_3090	c33MP3CalcStereoProcess()	Performs stereo processing for MPEG1 Layer III data.
0x0002_3094	c33MP3CalcStereoProcessLSF()	Performs stereo processing for MPEG2 Layer III LSF data.
0x0002_3098	c33MP3CalcSubbandSynthesisFdct()	Performs Fast-DCT calculation processing.
0x0002_309C	c33MP3CalcSubbandSynthesisWindow()	Performs window calculation for sub-band synthesis.
0x0002_30a0	c33MP3CalcSubbandSynthesisWindow2()	Performs window calculation for sub-band synthesis.
0x0002_30a4	c33MP3CalcSubbandSynthesisWindow4()	Performs window calculation for sub-band synthesis.
0x0002_30a8	c33MP3CalcSubbandSynthesisWindow8()	Performs window calculation for sub-band synthesis.

These functions may be used without any initial processing.

Note: The addresses may be changed in BIOS update in the future.

MP3 decoding flowchart

Figure X.1.3.2.1 shows an MP3 decoding flowchart. Refer to the sample program supplied for how to use the low-level MP3 API functions.

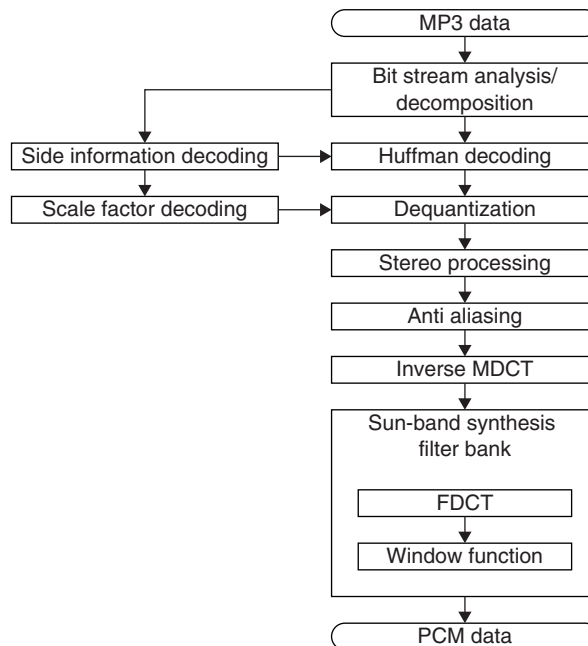


Figure X.1.3.2.1 MP3 Decoding Flowchart

X.1.4 MP3 Format

X.1.4.1 MP3 File Format

The MP3 decoder BIOS can treat only the fundamental MP3 format (hereafter, called “MP3 bit stream”) that was standardized with ISO/IEC.

However, extended formats that include MP3 bit stream are widely diffused, because general MP3 players require various information in addition to MP3 bit stream. When the MP3 decoder BIOS is used, you should skip the extended part and send only the MP3 bit stream part to the MP3 decoder BIOS.

The following shows the MP3 file formats that are generally used:

MP3 file (file extension: “.mp3”)

The most popular formats in various MP3 files are shown below.

Format A



The MP3 decoder BIOS can skip the header after finding the sync-word (continuous 12 bits of ‘1’).

Format B



<table> <tr><td>ID3v2</td><td>3 bytes</td><td>“ID3”</td></tr> <tr><td></td><td>1 byte</td><td>Version</td></tr> <tr><td></td><td>1 byte</td><td>Revision</td></tr> <tr><td></td><td>1 byte</td><td>Flag</td></tr> <tr><td></td><td>4 bytes</td><td>Size of ID3 header</td></tr> </table>	ID3v2	3 bytes	“ID3”		1 byte	Version		1 byte	Revision		1 byte	Flag		4 bytes	Size of ID3 header	<table> <tr><td>ID3v1</td><td>3 bytes</td><td>“TAG”</td></tr> <tr><td></td><td>30 bytes</td><td>Title</td></tr> <tr><td></td><td>30 bytes</td><td>Artist</td></tr> <tr><td></td><td>30 bytes</td><td>Album</td></tr> <tr><td></td><td>4 bytes</td><td>Release year</td></tr> <tr><td></td><td>30 bytes</td><td>Comment</td></tr> <tr><td></td><td>1 byte</td><td>Genre</td></tr> </table>	ID3v1	3 bytes	“TAG”		30 bytes	Title		30 bytes	Artist		30 bytes	Album		4 bytes	Release year		30 bytes	Comment		1 byte	Genre
ID3v2	3 bytes	“ID3”																																			
	1 byte	Version																																			
	1 byte	Revision																																			
	1 byte	Flag																																			
	4 bytes	Size of ID3 header																																			
ID3v1	3 bytes	“TAG”																																			
	30 bytes	Title																																			
	30 bytes	Artist																																			
	30 bytes	Album																																			
	4 bytes	Release year																																			
	30 bytes	Comment																																			
	1 byte	Genre																																			

The MP3 decoder BIOS cannot skip the ID3-tag, because the ID3-tag may have the sync-word.

Figure X.1.4.1.1 MP3 File Format (“.mp3”)

RIFF wave file (file extension: “.wav”)

This is an MP3 file that is supported by Windows. A data chunk in this Windows RIFF format has either “Format A” or “Format B” that are mentioned above.

RIFF MP3 file (file extension: “.rmp”)

This is an MP3 file that is supported by Windows. A data chunk in this Windows RIFF format has “Format A” that is mentioned above.

X.1.4.2 MP3 Bit Stream Format

The composition of an MP3 bit stream is shown below.

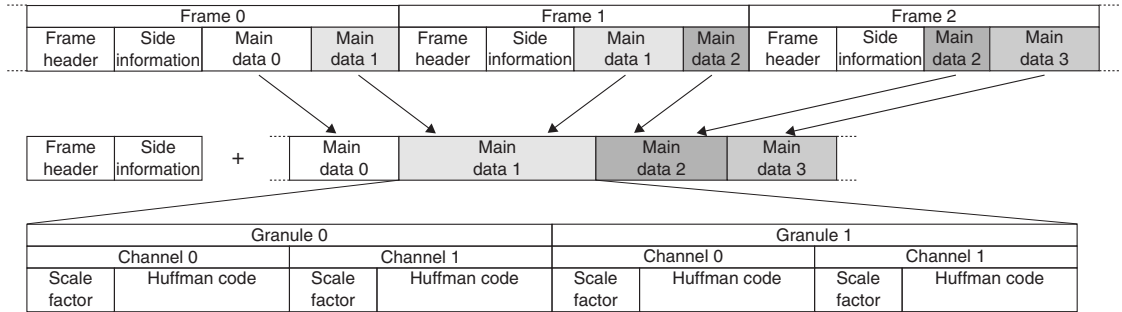


Figure X.1.4.2.1 MP3 Bit Stream Format

One frame data in an MP3 bit stream is composed of a frame header, side information and main data. The main data is variable length data compressed as Huffman code.

One frame is composed of two granules in MPEG1 Layer III or one granule in MPEG2 Layer III. One granule is composed of 576 samples for two channels (L and R).

Table X.1.4.2.1 Construction of Frame and Granule on MPEG Layer III

Content	MPEG2 Layer III			MPEG1 Layer III		
Sampling frequency [kHz]	16	22.05	24	32	44.1	48
Number of granule / frame	1 [granule]			2 [granules]		
Number of sample / granule	576 [samples]					
Number of channel / granule	2 [channels]					

X.1.5 Details of MP3 Decoder BIOS API

X.1.5.1 Return Values and Event Values

Table X.1.5.1.1 lists the return and event values sent from the high-level MP3 API functions.

Table X.1.5.1.1 Return Values and Event Values

Constant (return/event value)	Description
MP3DEC_SUCCESS (0)	Process was terminated successfully.
MP3DEC_FILE_END (10)	Decoding process reached end of the MP3 file.
MP3DEC_ERR_CANT_FIND_SYNC (-1)	No sync word can be found in the MP3 data. The MP3 data may be invalid.
MP3DEC_ERR_INVALID_HEADER (-2)	The frame header contains an invalid parameter. The MP3 data may be invalid.
MP3DEC_ERR_BAD_LAYER (-3)	The MP3 data is for the unsupported layer.
MP3DEC_ERR_NOT_OPEN (-10)	The decoder module cannot be opened. Check to see if the vector table is located in the RAM.
MP3DEC_ERR_INVALID_PARAM (-11)	The decoder found invalid parameter. Check to see if any invalid parameter is designated.
MP3DEC_ERR_INVALID_COMMAND (-12)	Invalid command. This command cannot be executed.

X.1.5.2 c33MP3DecOpen — Open MP3 Decoder

[Category]

High-level function

[Declaration]

```
long c33MP3DecOpen(T_MP3DEC_PARAM *Parameter)
```

[Argument]

Parameter <in> Pointer to the initial parameter structure (see the description below.)

[Return value]

MP3DEC_SUCCESS	(0)	Success
MP3DEC_ERR_NOT_OPEN	(-10)	Failure (The MP3 decoder cannot be opened.)
MP3DEC_ERR_INVALID_PARAM	(-11)	Failure (An invalid parameter was designated.)

[Description]

Opens the MP3 decoder module.

This function initializes the various parameters and hardware resources to activate the MP3 decoder. After that, the user program will be able to use the high-level API functions in the MP3 decoder BIOS.

Processing by this function is as follows:

1. Relocates IVRAM to Area 0.
2. Overwrites the vector table to change the HSDMA Ch.0 and Ch.1 vectors to the interrupt handler functions in the MP3 decoder BIOS.
3. Connects the P04–P07 ports to the I²S signals.
4. Enables the interrupt of HSDAM Ch.0, and set its priority level to seven.
5. Configures the I²S L and R channels to trigger HSDMA Ch.0 and Ch.1 for requesting output data.
6. Initializes the decoder resources.

If any configuration in Steps 1 to 5 fails, the decoder returns to closed status.

Note: Be sure not to call the `c33MP3DecOpen()` function twice before calling the `c33MP3DecClose()` function.

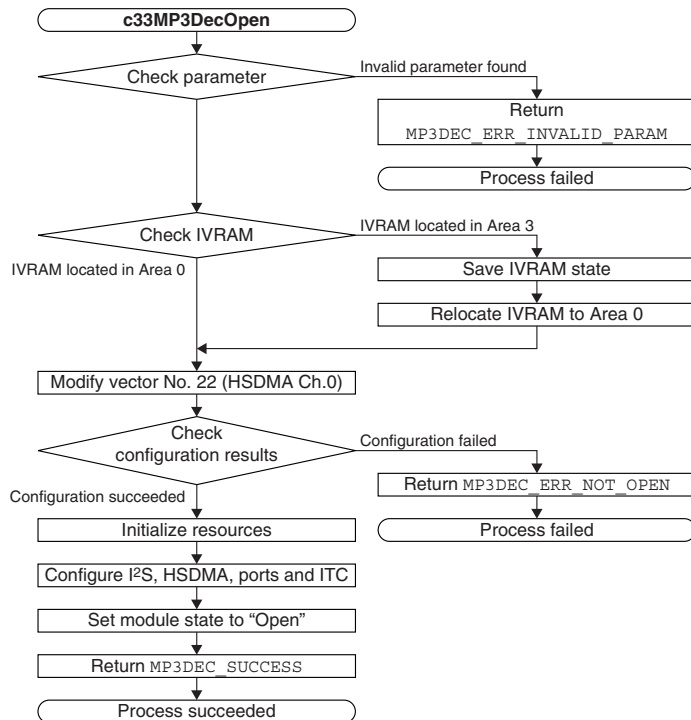


Figure X.1.5.2.1 Open Process Flow

T_MP3DEC_PARAM parameter

The argument “Parameter” is the pointer to a T_MP3DEC_PARAM type variable used to initialize the MP3 decoder.

The T_MP3DEC_PARAM structure is defined in the “mp3.h” header file. Include this file in the user program source file to create a T_MP3DEC_PARAM type variable and set the members before calling the c33MP3DecOpen() function.

Definition of the T_MP3DEC_PARAM structure

```
typedef struct {
    unsigned long    WorkAreaAddr;
    unsigned long    InputBufferAddr;
    unsigned long    InputBufferSize;
    unsigned long    SystemClock;
    unsigned long    Quality;
    unsigned long    I2SConfig;
    unsigned long    (*TransferData)(void*, unsigned long);
    void             (*Event)(long);
} T_MP3DEC_PARAM;
```

WorkAreaAddr

Set the top address of the work area used by the MP3 decoder.

The work area needs a 16K-byte area in an external RAM. The address must be aligned on a word boundary.

InputBufferAddr

Set the top address of the MP3 input data buffer. The MP3 decoder BIOS gets MP3 data from this buffer.

The buffer needs an external RAM area of the size designated with InputBufferSize. The address must be aligned on a word boundary.

InputBufferSize

Set the size of the MP3 input data buffer.

The size should be 4K bytes or more in eight-byte increments.

SystemClock

Set the frequency [Hz] of the MCLK clock that is supplied from the CMU to the I²S module.

The practical I²S output clock is decided by this parameter.

$$I^2S \text{ output clock} = \frac{\text{System clock}}{\text{ROUND}\left(\frac{\text{System clock}}{\text{Sampling frequency} \times 64}\right) \times 64}$$

“Sampling frequency” is determined by the setting in the MP3 file to be decoded.

Table X.1.5.2.1 lists examples of the I²S output clock frequencies.

Table X.1.5.2.1 Examples of I²S Output Clock Frequencies

CPU clock	Sampling frequency					
	48 kHz	44.1 kHz	32 kHz	24 kHz	22.05 kHz	16 kHz
60 MHz	46.875 (+2.3%)	44.642 (-1.2%)	32.327 (-1.0%)	24.038 (-0.2%)	21.802 (+1.1%)	15.889 (+0.7%)
48 MHz	46.875 (+2.3%)	44.117 (-0.0%)	32.608 (-1.9%)	24.193 (-0.8%)	22.058 (-0.0%)	15.957 (+0.3%)
30 MHz	46.875 (+2.3%)	42.613 (+3.4%)	31.250 (+2.3%)	23.437 (+2.6%)	22.321 (-1.2%)	16.163 (-1.0%)
24 MHz	46.875 (+2.3%)	41.666 (+5.5%)	31.250 (+2.3%)	23.437 (+2.4%)	22.058 (-0.0%)	16.304 (-1.9%)
12 MHz	46.875 (+2.3%)	46.875 (-6.3%)	31.250 (+2.3%)	23.437 (+2.6%)	20.833 (+5.5%)	15.625 (+2.3%)
8 MHz	41.666 (+13.2%)	41.666 (+5.5%)	31.250 (+2.3%)	25.000 (-4.2%)	20.833 (+5.5%)	15.625 (+2.3%)

Values mean “Practical sampling frequency [kHz] (error[%])”

Quality

Specify the quality mode for playing back MP3 data using a constant shown below.

MP3DEC_MODE_H (0) High quality mode using high CPU power

MP3DEC_MODE_M (1) Middle quality mode using middle CPU power

MP3DEC_MODE_L (2) Low quality mode using low CPU power

I2SConfig

Set the I²S output configuration value to be written to the I²S Control Register (0x301C00).

Use the constants shown below. Two or more constants can be specified by separating each of them with an operator '|'.

MP3DEC_WCLKMD_L_HIGH	(0x8)	Left = High level
MP3DEC_BCLKPOL_NEGATIVE	(0x4)	Trigger = Negative Edge
MP3DEC_DTTMG_NOT_DELAYED	(0x2)	Not delay
MP3DEC_DTFORM_LSB	(0x1)	LSB first

Refer to the “I²S Interface” section for details of the I²S Control Register (0x301C00).

TransferData

Set the pointer to the user function to transfer MP3 data to the MP3 decoder.

Refer to Section X.1.5.9, “TransferData (Callback Function),” for details.

Event

Set the pointer to the user function to handle events issued from the MP3 decoder BIOS.

When set to 0, no event will be issued.

Refer to Section X.1.5.10, “Event (Callback Function),” for details.

[Reference]

X.1.5.3 c33MP3DecClose (), X.1.5.9 TransferData (), X.1.5.10 Event ()

X.1.5.3 c33MP3DecClose — Close MP3 Decoder

[Category]

High-level function

[Declaration]

void c33MP3DecClose(void)

[Argument]

(none)

[Return value]

(none)

[Description]

Closes the MP3 decoder module.

This function deactivates the MP3 decoder and it releases all resources used for the MP3 decoder to restore the conditions to those before the MP3 decoder opened.

Be sure to call this function before terminating the MP3 application.

Processing by this function is as follows:

1. Stops playback if the decoder is in playback.
2. Disables the I²S module and HSDMA Ch.0 and Ch.1.
3. Restores the HSDMA Ch.0 interrupt vector.
4. Restores IVRAM to Area 3 if IVRAM was located in Area 3 before the MP3 decoder opened.

Call the c33MP3DecOpen () function to use the MP3 decoder once again after it is closed.

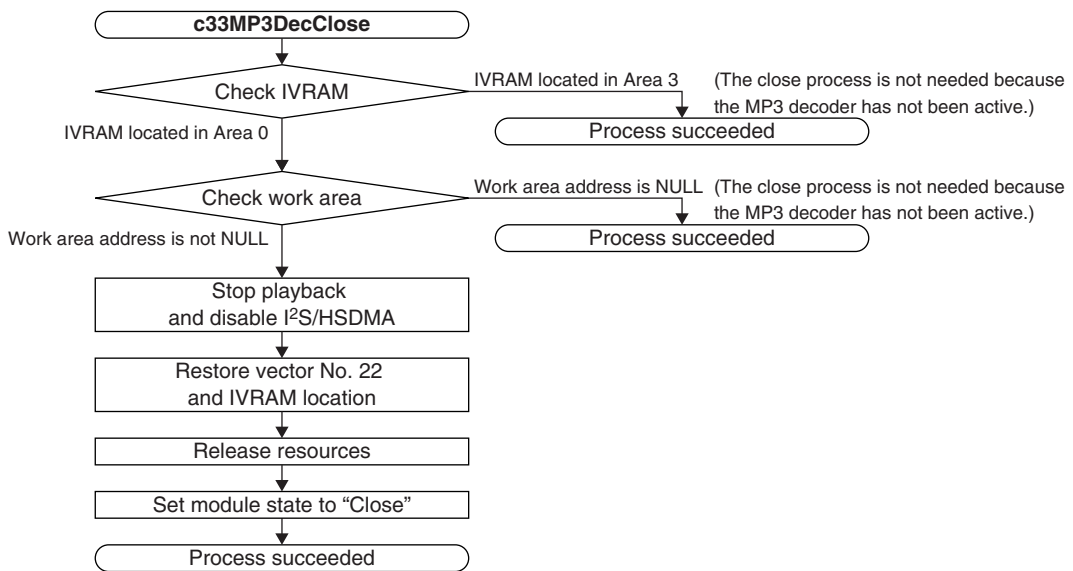


Figure X.1.5.3.1 Close Process Flow

[Reference]

X.1.5.2 c33MP3DecOpen ()

X.1.5.4 c33MP3DecStart — Start Playback

[Category]

High-level function

[Declaration]

```
long c33MP3DecStart(void)
```

[Argument]

(none)

[Return value]

MP3DEC_SUCCESS	(0)	Success
MP3DEC_FILE_END	(10)	End of file
MP3DEC_ERR_CANT_FIND_SYNC	(-1)	Failure (No sync word can be found in an audio frame.)
MP3DEC_ERR_INVALID_HEADER	(-2)	Failure (Data has an invalid header.)
MP3DEC_ERR_BAD_LAYER	(-3)	Failure (Data is a bad layer.)

[Description]

Starts decoding and playback.

Processing by this function is as follows:

1. Stops playback if the decoder is in playback.
2. Initializes the software and hardware resources.
3. Reads MP3 data to fill the input buffer.
4. Decodes the MP3 data until the output buffer is filled up.
5. Enables the I²S and HSDMA to activate and starts playback.

This function does not skip “ID3 Tag,” because it may contain a sync word. When decoding MP3 data with an ID3 Tag, the ID3 Tag should be removed before sending MP3 data to the decoder.

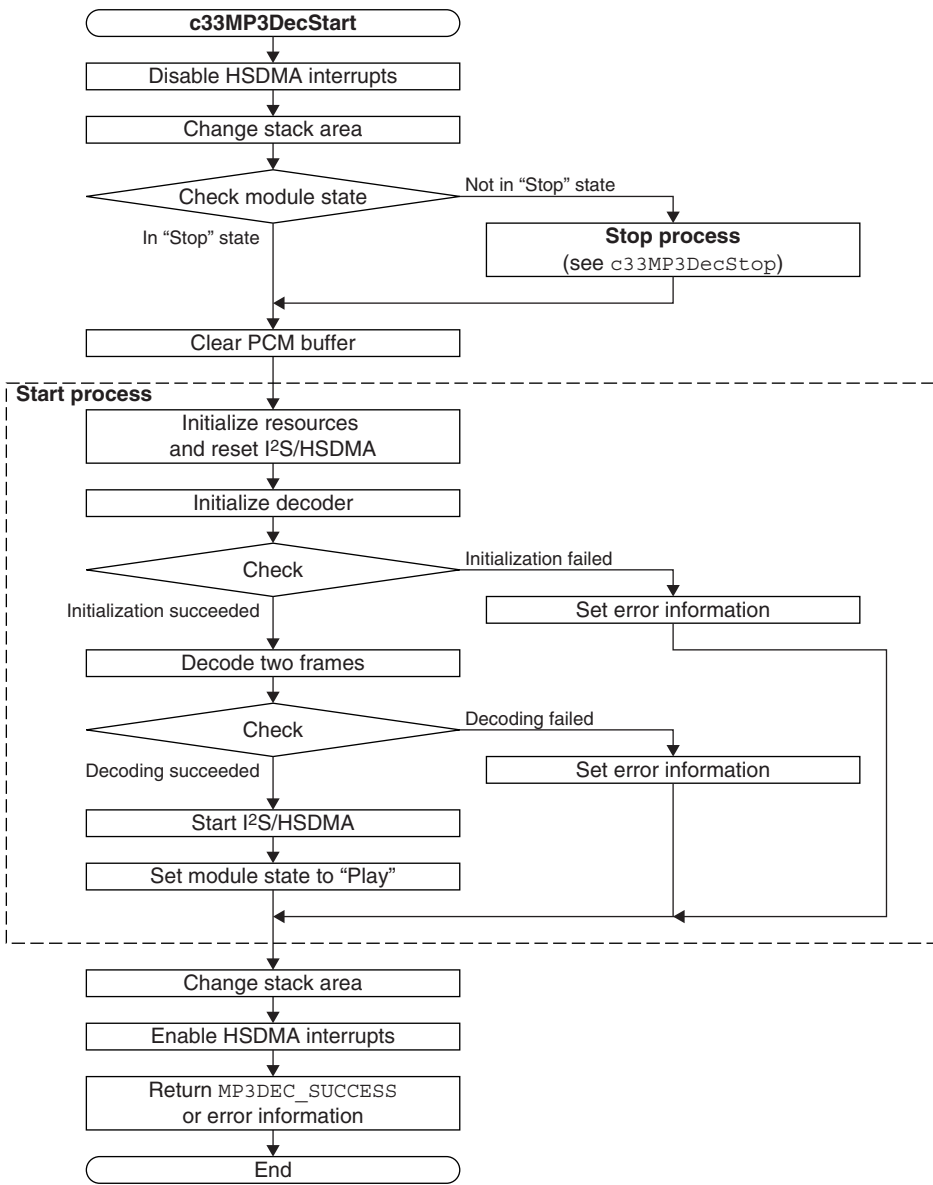


Figure X.1.5.4.1 Start Process Flow

[Reference]

X.1.5.5 c33MP3DecStop (), X.1.5.6 c33MP3DecPause ()

X.1.5.5 c33MP3DecStop — Stop Playback

[Category]

High-level function

[Declaration]

```
long c33MP3DecStop(void)
```

[Argument]

(none)

[Return value]

MP3DEC_SUCCESS (0) Success
 MP3DEC_ERR_INVALID_COMMAND (-12) Invalid command. This command cannot be executed.

[Description]

Stops playback.

This function stops hardware operations and clears the interrupt flag.

The difference between `c33MP3DecStop()` and `c33MP3DecPause()` is as follows:

- The `c33MP3DecStop()` function flushes the condition of the input and output buffers.
- The `c33MP3DecPause()` function holds the condition of the input and output buffers.

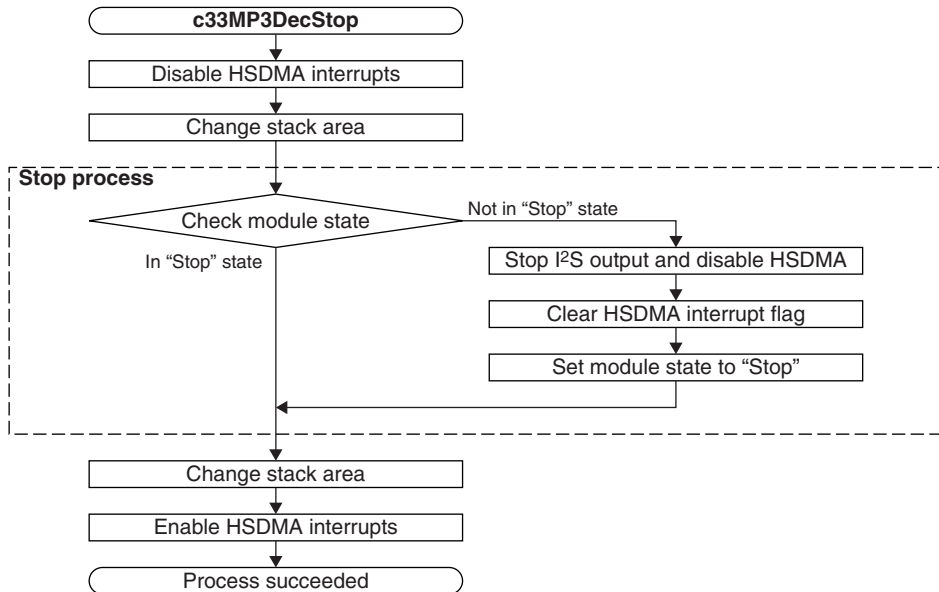


Figure X.1.5.5.1 Stop Process Flow

[Reference]

X.1.5.4 `c33MP3DecStart()`, X.1.5.6 `c33MP3DecPause()`

X.1.5.6 c33MP3DecPause — Pause Playback

[Category]

High-level function

[Declaration]

long c33MP3DecPause(void)

[Argument]

(none)

[Return value]

MP3DEC_SUCCESS (0) Success
 MP3DEC_ERR_INVALID_COMMAND (-12) Invalid command. This command cannot be executed.

[Description]

Pauses playback.

This pause function allows resuming of playback from the stopped position. Call the c33MP3DecResume () function to release pause status.

The difference between c33MP3DecPause () and c33MP3DecStop () is as follows:

- The c33MP3DecPause () function holds the condition of the input and output buffers.
- The c33MP3DecStop () function flushes the condition of the input and output buffers.

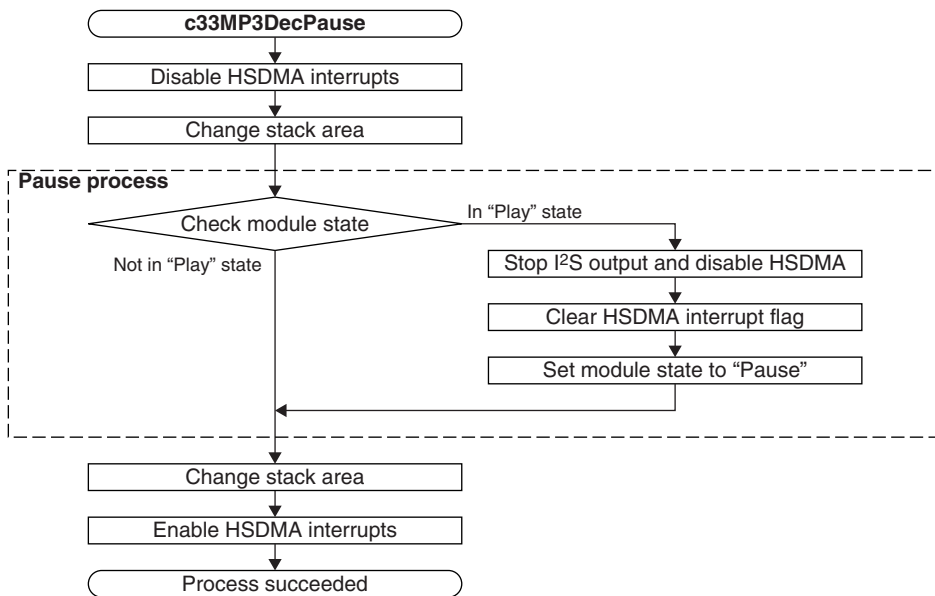


Figure X.1.5.6.1 Pause Process Flow

[Reference]

X.1.5.7 c33MP3DecResume (), X.1.5.5 c33MP3DecStop ()

X.1.5.7 c33MP3DecResume — Resume Playback

[Category]

High-level function

[Declaration]

```
long c33MP3DecResume(void)
```

[Argument]

(none)

[Return value]

MP3DEC_SUCCESS (0) Success
 MP3DEC_ERR_INVALID_COMMAND (-12) Invalid command. This command cannot be executed.

[Description]

Releases pause status to resume playback.

This function starts clock output from the I²S module to restart playback.

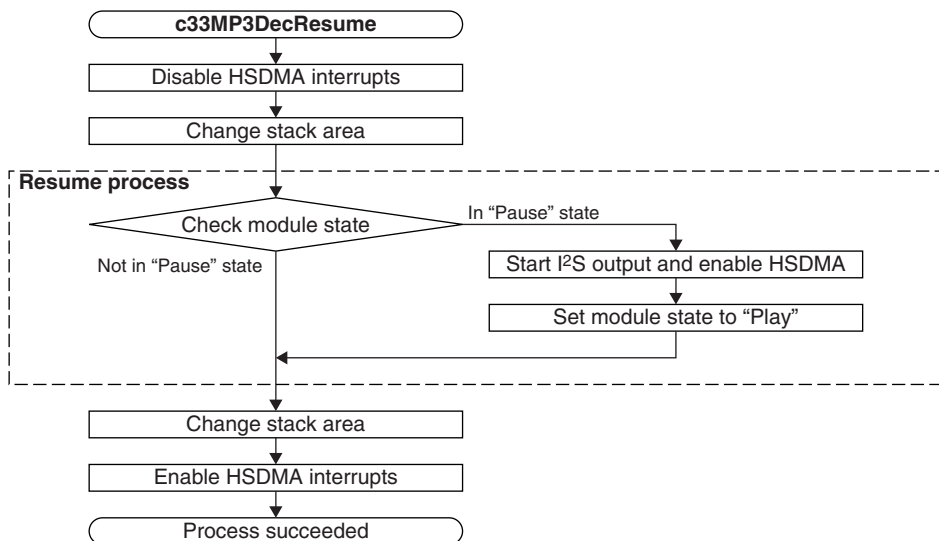


Figure X.1.5.7.1 Resume Process Flow

[Reference]

X.1.5.6 c33MP3DecPause()

X.1.5.8 c33MP3DecResync — Resynchronization

[Category]

High-level function

[Declaration]

```
long c33MP3DecResync(void)
```

[Argument]

(none)

[Return value]

MP3DEC_SUCCESS	(0)	Success
MP3DEC_FILE_END	(10)	End of file
MP3DEC_ERR_CANT_FIND_SYNC	(-1)	Failure (No sync word can be found in an audio frame.)
MP3DEC_ERR_INVALID_HEADER	(-2)	Failure (Data has an invalid header.)
MP3DEC_ERR_BAD_LAYER	(-3)	Failure (Data is a bad layer.)

[Description]

Synchronizes with the data stored in the input buffer to start playback from a new position. Use this function for forwarding or rewinding playback.

This function skips data until a first sync-word appears and it starts playback from the data that follows the sync-word.

Processing by this function is as follows:

1. Stops playback if the decoder is in playback.
2. Initializes the software and hardware resources.
3. Reads and decodes MP3 data. The decoder regards that the re-sync process has succeeded when an audio frame can be continuously decoded three times.
4. Enables the I²S and HSDMA to activate and starts playback.

The difference between `c33MP3DecStart()` and `c33MP3DecResync()` is as follows:

- The `c33MP3DecStart()` function treats MP3 data assuming that the data is in sync.
- The `c33MP3DecResync()` function treats MP3 data assuming that the data is out of sync.

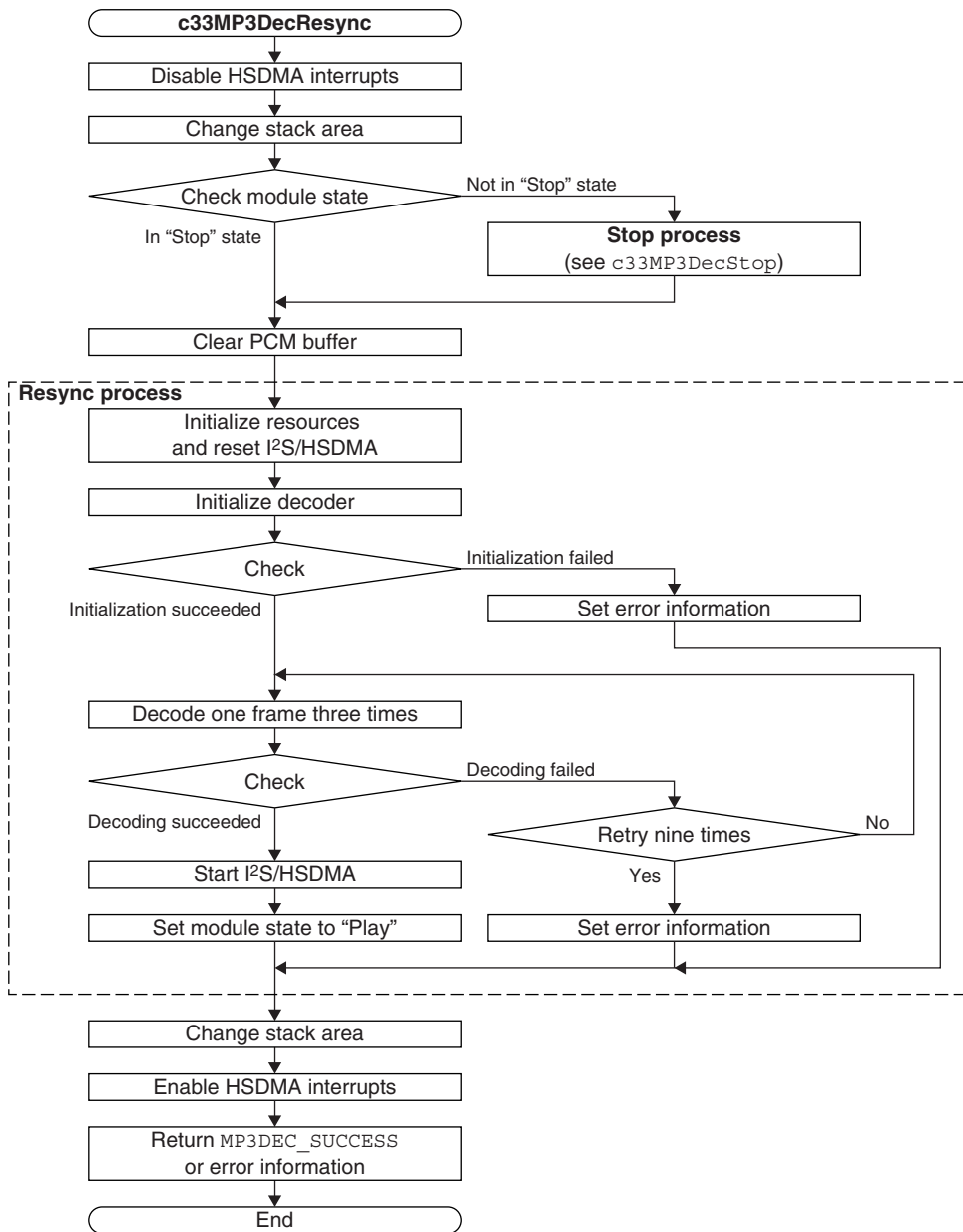


Figure X.1.5.8.1 Resync Process Flow

[Reference]

X.1.5.4 c33MP3DecStart ()

X.1.5.9 TransferData (Callback Function) — Set MP3 Data

[Category]

High-level function (User function called from the MP3 decoder BIOS)

[Declaration]

```
unsigned long TransferData(void *DstAddr, unsigned long Size)
```

[Argument]

DstAddr <in> Destination address
Size <in> Transfer size [Byte]

[Return value]

TransQuant Data size actually transferred [Byte]

[Description]

Transfers MP3 data to the MP3 input buffer.

This function is called from the MP3 decoder BIOS every time the MP3 input buffer becomes empty.

The user must create this function in accordance with the specification below.

- Write MP3 data of the size (*Size*) to the address (*DstAddr*).
- Return the actual data size transferred as its return value.

The MP3 decoder BIOS does not save registers (R10–R15). It is necessary to write a wrapper code that saves registers. (Refer to the example below.)

Set the pointer to this function as the initial parameter (*TransferData* member of the *T_MP3DEC_PARAM* structure).

The following is an example of the function to transfer data from a file:

Example of *TransferData* ()

```
// Prototype definition
extern void _TransferData(void);
extern unsigned long TransferData(void *DstAddr, unsigned long Size);

// Initialize Decoder I/F API
:
:
param.WorkAreaAddr      = workAddress;
param.InputBufferAddr   = mp3BuffAddress;
param.InputBufferSize   = mp3BuffSize;
param.SystemClock       = AudioClock;
param.Quality            = mode;
param.I2SConfig         = AudioDacConfigure;
param.TransferData      = (unsigned long(*) (void*, unsigned long))_TransferData;
param.Event              = AudioEventsProc;

if(c33MP3DecOpen(&param) != MP3DEC_SUCCESS){
    return AudioSetEvent(AUDIO_ERR_DEC_NOT_OPEN);
}

:
:

// Wrapper for saving R10-R15 (It is necessary.)
void _TransferData(void)
{
    asm("push %r10");
    asm("push %r11");
    asm("push %r12");
    asm("push %r13");
    asm("push %r14");
    asm("push %r15");
    asm("xcall TransferData");
    asm("pop %r15");
}
```

```
asm("pop %r14");
asm("pop %r13");
asm("pop %r12");
asm("pop %r11");
asm("pop %r10");
}

// Main process that transfers data
unsigned long TransferData(void *DstAddr, unsigned long Size)
{
    return fread(DstAddr, 1, Size, fin);
}
```

[Reference]

X.1.5.2 c33MP3DecOpen()

X.1.5.10 Event (Callback Function) — Event Handler

[Category]

High-level function (User function called from the MP3 decoder BIOS)

[Declaration]

```
void Event (long EventFlag)
```

[Argument]

EventFlag	<in>	Event flag	
		DEC_FILE_END	(10) The MP3 data being decoded reached end of the file.
		DEC_ERR_CANT_FIND_SYNC	(-1) No sync-word can be found.
		DEC_ERR_INVALID_HEADER	(-2) The header contains invalid information. (The decoder cannot decode data.)
		DEC_ERR_BAD_LAYER	(-3) The header contains information other than Layer-III. (The decoder cannot decode data.)

[Return value]

(none)

[Description]

Handles an event that has occurred during decoding.

This function is called from the MP3 decoder BIOS when an error causes the MP3 decoder to stop processing. At this time the MP3 decoder BIOS pass an argument `EventFlag` to notify the user that an event occurred.

The user can create an `Event ()` function when the application needs an event handler. To use the `Event ()` function, set its pointer as the initial parameter (`Event` member of the `T_MP3DEC_PARAM` structure). When not using an `Event ()` function, set the `Event` parameter to 0. Setting the `Event` parameter 0 disables the MP3 decoder BIOS to call the `Event ()` function even when an event occurs.

Note: Be sure to avoid calling MP3 API functions from this function.

The following is an example of the function:

Example of `Event ()`

```
void UserEventFunc (long EventFlag) {
    switch (EventFlag) {
        case MP3DEC_FILE_END:
            // Go to next track.
            . . .
            break;

        case DEC_ERR_CANT_FIND_SYNC:
        case DEC_ERR_INVALID_HEADER:
        case DEC_ERR_BAD_LAYER:
            // Show error message "This is not MP3 format."
            . . .
            break;

        default:
            // Undefined event
            break;
    }
}
```

[Reference]

X.1.5.2 `c33MP3DecOpen ()`

X.1.5.11 c33MP3CalcStereoProcess — Stereo Process

[Category]

Low-level function

[Declaration]

```
void c33MP3CalcStereoProcess(long *out_data, long *in_data, short *is_pos,
int sblimit, int ms_stereo)
```

[Argument]

- out_data <out> Output block data (out_data [2] [sblimit] [18])
- in_data <in> Input block data (in_data [2] [sblimit] [18])
- is_pos <in> Intensity stereo position (is_pos [sblimit] [18])
- sblimit <in> Sub-band limit (sblimit = 1-32)
- ms_stereo <in> MS-stereo flag
 0: MS-stereo
 Other: LR-stereo

[Return value]

(none)

[Data structure]

The input block data (in_data) must have an array structure with [2][32][18] elements regardless of the sblimit value.

If the sblimit is smaller than 32, no redundant area is used.

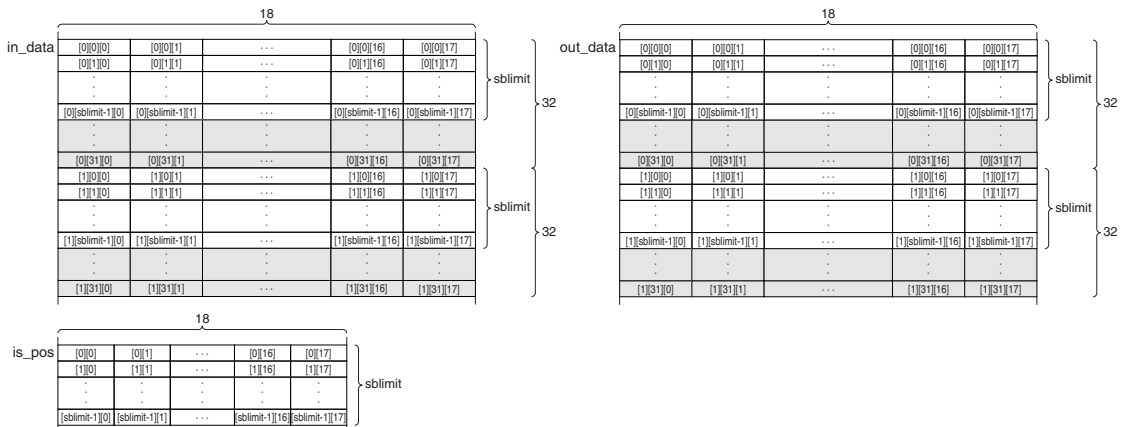


Figure X.1.5.11.1 Data Structure Supported by the `c33MP3CalcStereoProcess()` Function

[Restriction]

The operation of this function cannot be guaranteed if the `sblimit` is out of the valid range.

[Description]

Performs stereo processing for one block of MPEG1 Layer III data that has been dequantized by the previous process.

This function transforms the block data to L and R stereo block data. Outline of this function is shown below.

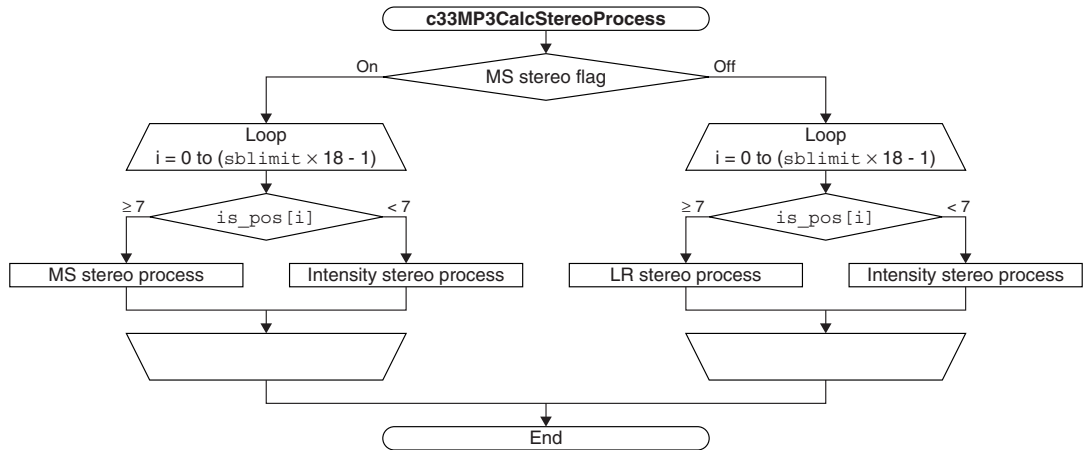


Figure X.1.5.11.2 Flowchart of Stereo Process

LR stereo process

$$L = l, R = r$$

MS stereo process

$$L = \frac{(l+r)}{\sqrt{2}}, R = \frac{(l-r)}{\sqrt{2}}$$

Intensity stereo process

$$is_ratio = \tan\left(\frac{\pi}{12} \times is_pos\right)$$

$$L = m \times \frac{is_ratio}{1 + is_ratio}, R = m \times \frac{1}{1 + is_ratio}$$

Resource requirements

Stack size: 24 [bytes]

Register: %r4, %r5, %alr	Work register
%r6	Argument out_data
%r7	Argument in_data
%r8	Argument is_pos
%r9	Argument sblimit
[%sp+1]	Argument ms_stereo

[Reference]

X.1.5.12 c33MP3CalcStereoProcessLSF ()

X.1.5.12 c33MP3CalcStereoProcessLSF — Stereo Process for LSF

[Category]

Low-level function

[Declaration]

```
void c33MP3CalcStereoProcessLSF(long *out_data0, long *in_data0,
    long *out_data1, long *in_data1, short *is_pos, int sblimit, int ms_stereo,
    int io)
```

[Argument]

out_data0 <out> Ch.0 output block data (out_data0[sblimit][18])
in_data0 <in> Ch.0 input block data (in_data0[sblimit][18])
out_data1 <out> Ch.1 output block data (out_data1[sblimit][18])
in_data1 <in> Ch.1 input block data (in_data1[sblimit][18])
is_pos <in> Intensity stereo position (is_pos[sblimit][18])
sblimit <in> Sub-band limit (sblimit = 1–32)
ms_stereo <in> MS-stereo flag
0: MS-stereo
Other: LR-stereo
io <in> Table number (0 or 1)

[Return value]

(none)

[Data structure]

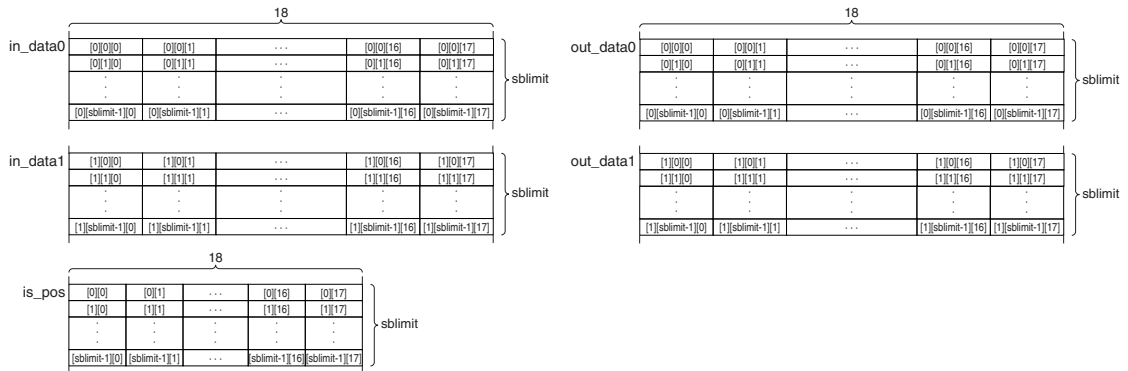


Figure X.1.5.12.1 Data Structure Supported by the `c33MP3CalcStereoProcessLSF()` Function

[Restriction]

The operation of this function cannot be guaranteed if the `sblimit` is out of the valid range.

[Description]

Performs stereo processing for one block of MPEG2 Layer III LSF data that has been dequantized by the previous process.

This function transforms the block data to L and R stereo block data. Outline of this function is shown below.

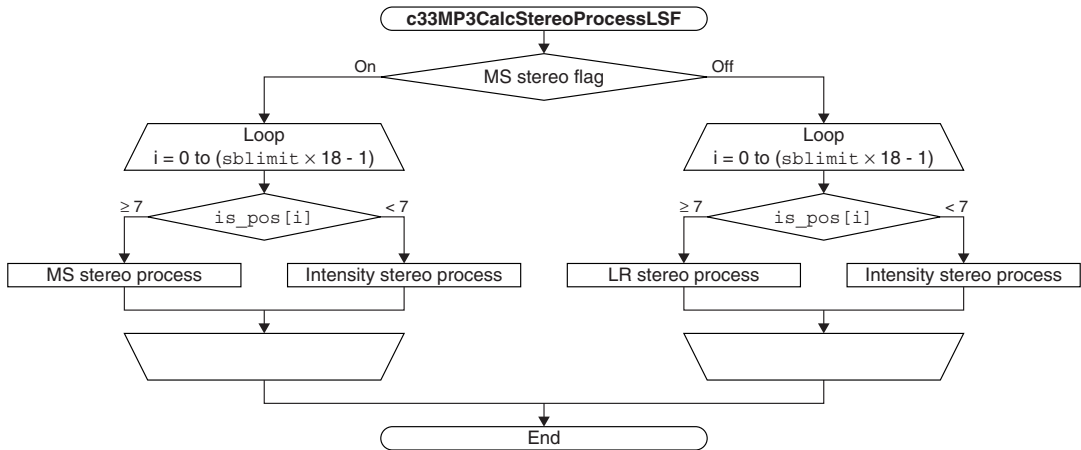


Figure X.1.5.12.2 Flowchart of Stereo Process (LSF)

LR stereo process

$$L = l, R = r$$

MS stereo process

$$L = \frac{(l + r)}{\sqrt{2}}, R = \frac{(l - r)}{\sqrt{2}}$$

Intensity stereo process (for LSF data)

Table X.1.5.12.1 Equations of Intensity Stereo Process (LSF)

is_pos	In case of scalefac_compress = 0		In case of scalefac_compress = 1	
	L-ch.	R-ch.	L-ch.	R-ch.
0	$L = m$	$R = m$	$L = m$	$R = m$
1	$L = m \times 2^{(-1/2)}$	$R = m$	$L = m \times 2^{(-1/4)}$	$R = m$
2	$L = m$	$R = m \times 2^{(-1/2)}$	$L = m$	$R = m \times 2^{(-1/4)}$
3	$L = m \times 2^{(-2/2)}$	$R = m$	$L = m \times 2^{(-2/4)}$	$R = m$
4	$L = m$	$R = m \times 2^{(-2/2)}$	$L = m$	$R = m \times 2^{(-2/4)}$
5	$L = m \times 2^{(-3/2)}$	$R = m$	$L = m \times 2^{(-3/4)}$	$R = m$
6	$L = m$	$R = m \times 2^{(-3/2)}$	$L = m$	$R = m \times 2^{(-3/4)}$

Resource requirements

Stack size: 16 [bytes]

Register: %r4, %r5, %r10-14, %alr Work register
 %r6 Argument out_data0
 %r7 Argument in_data0
 %r8 Argument out_data1
 %r9 Argument in_data1
 [%SP+1] Argument is_pos
 [%SP+2] Argument sblimit
 [%SP+3] Argument ms_stereo
 [%SP+4] Argument io

[Reference]

X.1.5.11 c33MP3CalcStereoProcess ()

X.1.5.13 c33MP3CalcAntiAlias — Anti Alias Process

[Category]

Low-level function

[Declaration]

```
void c33MP3CalcAntiAlias(long *in_data, long *out_data, int sblimit)
```

[Argument]

`in_data[sblimit][18]` <in> Array of input sub-band data
 One sub-band data consists of 18 samples (one sample = 32 bits).

`out_data[sblimit][18]` <out> Array of output sub-band data
 One sub-band data consists of 18 samples (one sample = 32 bits).

`sblimit` <in> Number of sub-band - 1 (`sblimit` = 1–31)
 For example, if there are 32 sub-bands, this parameter is 31.

[Return value]

(none)

[Restriction]

The operation of this function cannot be guaranteed if the `sblimit` is out of the valid range.

[Description]

Performs butterfly calculation processing to plural sub-band data.

The butterfly calculation method is shown below.

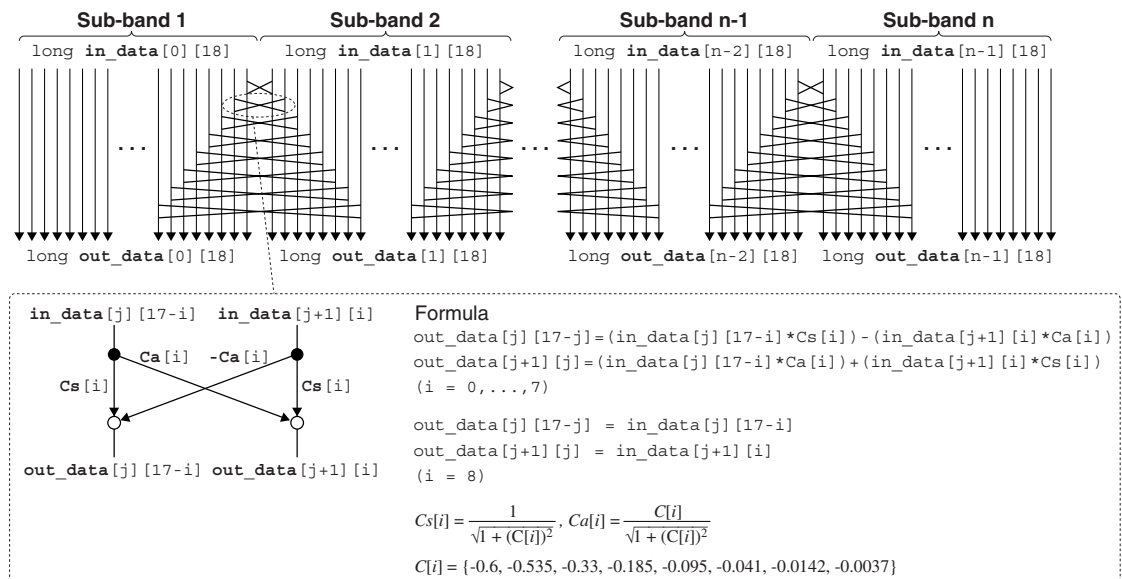


Figure X.1.5.13.1 Diagram of Anti Alias Processing

Resource requirements

Stack size: 12 [bytes]

Register: %r4, %r5, %r9 Work register
 %r6 Argument `in_data`
 %r7 Argument `out_data`
 %r8 Argument `sblimit`

[Reference]

(none)

X.1.5.14 c33MP3CalcIdctLong — Inverse MDCT Calculation for Long Block

[Category]

Low-level function

[Declaration]

```
void c33MP3CalcIdctLong(long in_data[18], long out_data[18])
```

[Argument]

```
in_data[18]    <in>    Input sub-band data
out_data[18]   <out>   Output sub-band data
```

[Return value]

(none)

[Description]

Performs inverse MDCT calculation processing to one sub-band data of a long block. This calculation is based on Lee 84' FCT.

(Reference) Inverse MDCT

$$x_i = \sum_{k=0}^{\frac{n}{2}-1} X_k \cos\left(\frac{\pi}{2n}\left(2i+1+\frac{n}{2}\right)(2k+1)\right)$$

for $i = 0, 1, \dots, n-1$ ($n = 36$)

Generally, this function is used with the `c33MP3CalcIdctWindow()` function.

An example is shown below.

Example: Usage of `c33MP3CalcIdctLong()` and `c33MP3CalcIdctWindow()`

```
unsigned long   in_data[18];           // Input subband data
unsigned long   tmp[36];
unsigned long   out_data[36];         // Output subband data

c33MP3CalcIdctLong(in_data, tmp);
c33MP3CalcIdctWindow(out_data, tmp, 0); // Window table is type '0'
```

Resource requirements

Stack size: 104 [bytes]

```
Register:  %r4, %r5, %r8, %r9  Work register
           %r6                  Argument in_data
           %r7                  Argument out_data
```

[Reference]

X.1.5.15 `c33MP3CalcIdctWindow()`, X.1.5.16 `c33MP3CalcIdctShort()`

X.1.5.15 c33MP3CalcIdctWindow — Window Calculation

[Category]

Low-level function

[Declaration]

```
void c33MP3CalcIdctWindow(long out_data[36], long in_data[18],
    int block_type)
```

[Argument]

out_data[36]	<out>	Output sub-band data
in_data[18]	<in>	Input sub-band data
block_type	<in>	Block table No. (No. = 0-3)

[Return value]

(none)

[Description]

Performs window calculation processing after inverse MDCT to one sub-band data for a long block. The window calculation method is shown below.

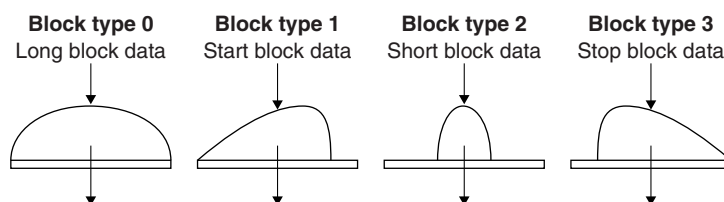


Figure X.1.5.15.1 Diagram of Window Function After Inverse MDCT

Resource requirements

Stack size: 0 [bytes]

Register:	%r4, %r5, %alr	Work register
	%r6	Argument out_data
	%r7	Argument in_data
	%r8	Argument block_type

[Reference]

X.1.5.14 c33MP3CalcIdctLong(), X.1.5.16 c33MP3CalcIdctShort()

X.1.5.16 c33MP3CalcIdctShort — Inverse MDCT Calculation for Short Block

[Category]

Low-level function

[Declaration]

```
void c33MP3CalcIdctShort(long in_data[18], long out_data[36])
```

[Argument]

```
in_data[18]    <in>   Input sub-band data  
out_data[36]   <out>  Output sub-band data
```

[Return value]

(none)

[Description]

Performs inverse MDCT calculation processing to one sub-band data of a short block. This calculation is based on Lee 84' FCT.

Resource requirements

Stack size: 72 [bytes]

Register: %r4, %r5, %r8, %r9 Work register
 %r6 Argument in_data
 %r7 Argument out_data

[Reference]

X.1.5.14 c33MP3CalcIdctLong(), X.1.5.15 c33MP3CalcIdctWindow()

X.1.5.17 c33MP3CalcSubbandSynthesisFdct — Fast-DCT Calculation

[Category]

Low-level function

[Declaration]

```
void c33MP3CalcSubbandSynthesisFdct(long in_data[32],
    long out_data_even[16], long out_data_odd[16])
```

[Argument]

in_data [32]	<in>	A block data on a sample. The block data has elements of 32 sub-bands.
out_data_even [16]	<out>	Even DCT output coefficients
out_data_odd [16]	<out>	Odd DCT output coefficients

[Return value]

(none)

[Description]

Performs Fast-DCT calculation processing to block data on one sample. This calculation is based on a 32 point IDCT as derived by LEE '84 & Konstantinedes.

Resource requirements

Stack size: 104 [bytes]

Register: %r4, %r5 Work register
 %r6 Argument in_data
 %r7 Argument out_data_even
 %r8 Argument out_data_odd

[Reference]

X.1.5.18 c33MP3CalcSubbandSynthesisWindow(),
 X.1.5.19 c33MP3CalcSubbandSynthesisWindow2/4/8()

X.1.5.18 c33MP3CalcSubbandSynthesisWindow — Window Calculation for Sub-band Synthesis

[Category]

Low-level function

[Declaration]

```
void c33MP3CalcSubbandSynthesisWindow(long *in_ptr, short out_data[32],
long in_data[512])
```

[Argument]

in_ptr	<in>	Current position on in_data
out_data[32]	<out>	Output PCM data
in_data[512]	<in>	Input data

[Return value]

(none) In the case of c33MP3CalcSubbandSynthesisWindow()

[Description]

Performs window calculation for sub-band synthesis.

Resource requirements

Stack size: 0 [bytes]

Register:	%r4, %r5	Work register
	%r6	Argument in_ptr
	%r7	Argument out_data
	%r8	Argument in_data

[Reference]

X.1.5.17 c33MP3CalcSubbandSynthesisFdct()

X.1.5.19 c33MP3CalcSubbandSynthesisWindow2/4/8 — Window Calculation for Sub-band Synthesis 2/4/8

[Category]

Low-level function

[Declaration]

```
short c33MP3CalcSubbandSynthesisWindow2(long *in_ptr, short out_data[32],
long in_data[512], short carry_in)

short c33MP3CalcSubbandSynthesisWindow4(long *in_ptr, short out_data[32],
long in_data[512], short carry_in)

short c33MP3CalcSubbandSynthesisWindow8(long *in_ptr, short out_data[32],
long in_data[512], short carry_in)
```

[Argument]

in_ptr	<in>	Current position on in_data
out_data[32]	<out>	Output PCM data
in_data[512]	<in>	Input data
carry_in	<in>	A value that was carried on the last time. Set 0 at the first time.

[Return value]

carry_out Carried value to set into carry_in at the next time

[Description]

Performs window calculation for sub-band synthesis.

The c33MP3CalcSubbandSynthesisWindow2/4/8() functions are compatible with the c33MP3CalcSubbandSynthesisWindow() function. However, the c33MP3CalcSubbandSynthesisWindow2/4/8() functions decrease calculation quantity by thinning and interpolation.

The c33MP3CalcSubbandSynthesisWindow2(), c33MP3CalcSubbandSynthesisWindow4(), and c33MP3CalcSubbandSynthesisWindow8() functions perform thinning and interpolation with 2, 4, and 8 multiples, respectively.

Resource requirements

Stack size: 0 [bytes]

Register:	%r4, %r5	Work register
	%r6	Argument in_ptr
	%r7	Argument out_data
	%r8	Argument in_data
	%r9	Argument carry_in

[Reference]

X.1.5.17 c33MP3CalcSubbandSynthesisFdct()

X.1.6 Performance

X.1.6.1 Support Formats

The following shows the MP3 formats supported in the MP3 decoder BIOS:

MPEG1 Layer III

Sampling frequency [kHz]	Bit-rate [kbps]													
	32	40	48	56	64	80	96	112	128	160	192	224	256	320
48	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
44.1	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
32	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

* All bit rates support VBR.

MPEG2 Layer III (LSF)

Sampling frequency [kHz]	Bit-rate [kbps]													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
24	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
22.05	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
16	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

* All bit rates support VBR.

MPEG2.5

Sampling frequency [kHz]	Bit-rate [kbps]													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11.025	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-

* This is the Fraunhofer-Gesellschaft original format.

- OK: Supported.
- ?: Not tested.
- : Not supported.

X.1.6.2 CPU Occupancy Ratio

The CPU time shared by the decoder (HSDMA interrupt process) depends on the sampling frequency and bit rate of the MP3 data to be played back.

For example, when the CPU/SDRAM clock is 48 MHz and the decoder is playing back 44.1 kHz sampled 128-kbps MP3 data in middle quality mode, the CPU time shared by the decoder is about 70% in 26.1 ms periods (*).

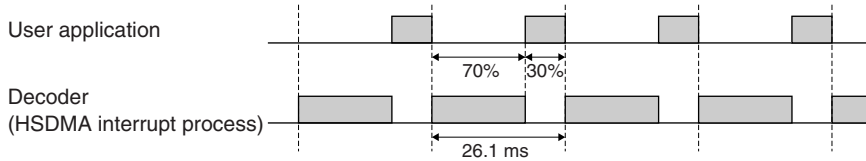
* $(1/\text{sampling frequency}) \times (\text{number of samples in an MP3 data frame})$

Number of samples: 576 when sampling frequency = 16–24 kHz

1152 when sampling frequency = 32–48 kHz

Example: When the sampling frequency = 44.1 kHz and the bit rate = 128 kbps

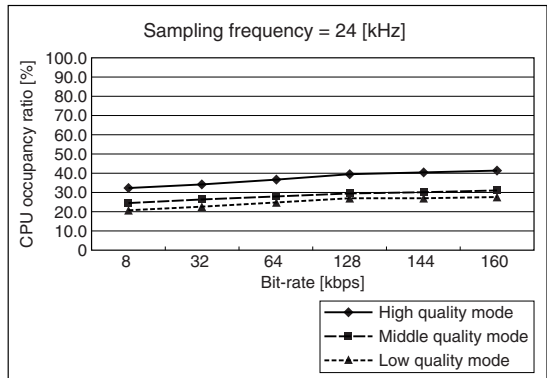
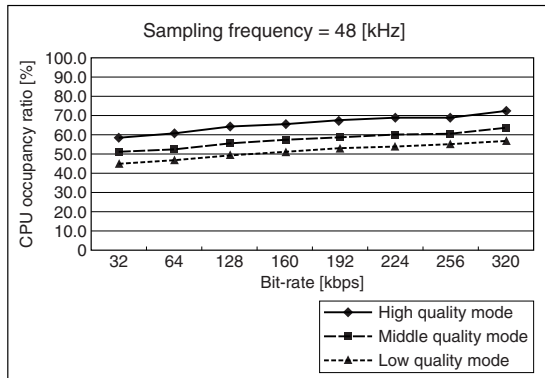
$$1/441000 \times 1152 = 22.6 \mu\text{s} \times 1152 = 26.1 \text{ ms}$$



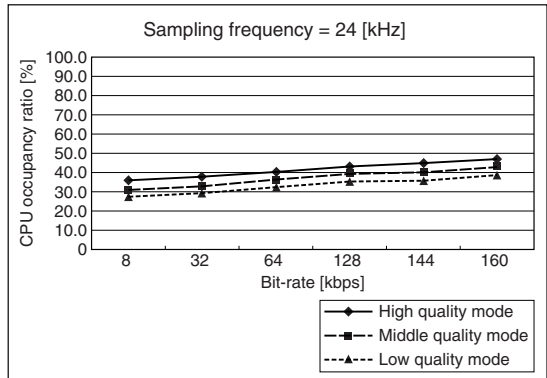
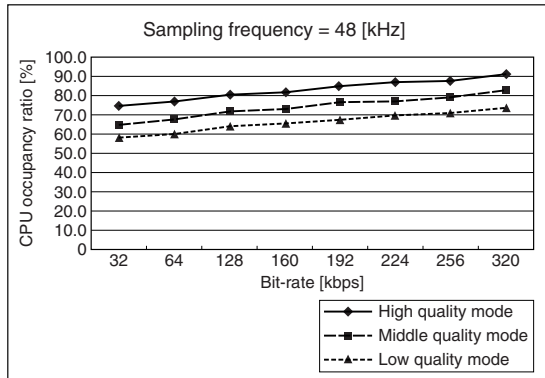
44.1 kHz, 128 kbps, middle quality mode (CPU/SDRAM clock = 48 MHz)

Figure X.1.6.2.1 CPU Occupancy Ratio

System clock: 60 [MHz], SDRAM clock: 60 [MHz]



System clock: 48 [MHz], SDRAM clock: 48 [MHz]



THIS PAGE IS BLANK.

S1C33E08 Technical Manual

Appendix

Appendix A I/O Map

0x300010–0x300020	Misc Register (1)	AP-A-2
0x300260–0x3002AF	Interrupt Controller	AP-A-3
0x300300–0x30031B	Card Interface	AP-A-11
0x300380–0x3003D5	I/O Ports	AP-A-13
0x300520–0x30055E	A/D Converter	AP-A-24
0x300660–0x30066C	Watchdog Timer	AP-A-27
0x300780–0x3007EA	16-bit Timer	AP-A-29
0x300900–0x30099F	USB Function Controller	AP-A-37
0x300B00–0x300B4F	Serial Interface	AP-A-47
0x300C00–0x300C25	Extended Ports	AP-A-52
0x300C40–0x300C4D	Misc Register (2)	AP-A-54
0x301100–0x301105	Intelligent DMA	AP-A-56
0x301120–0x30119E	High-Speed DMA	AP-A-57
0x301500–0x301510	SRAM Controller	AP-A-70
0x301600–0x301610	SDRAM Controller	AP-A-71
0x301700–0x30171C	SPI	AP-A-72
0x301800–0x30181C	DCSIO	AP-A-73
0x301900–0x301928	Real Time Clock	AP-A-74
0x301A00–0x301AAC	LCD Controller	AP-A-76
0x301B00–0x301B24	Clock Management Unit	AP-A-83
0x301C00–0x301C20	I ² S Interface	AP-A-87

Note:

(B), (HW), and (W) in [Address] indicate an 8-bit register, a 16-bit register, and a 32-bit register, respectively.

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.

0x300010–0x300020 **Misc Register**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC wait control register (pMISC_RTCWT) Protected	00300010 (B)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	RTCWT2	RTC register access wait control	0 to 7 (cycles)	1	R/W	
		D1 D0	RTCWT1 RTCWT0			1 1		
USB wait control register (pMISC_USBWT) Protected	00300012 (B)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	USBSNZ	USB snooze control	1 Enabled 0 Disabled	0	R/W	
		D4–3	–	reserved	–	–	–	0 when being read.
		D2 D1 D0	USBWT2 USBWT1 USBWT0	USB register access wait control	0 to 7 (cycles)	1 1 1	R/W	
Debug port MUX register (pMISC_PMUX) Protected	00300014 (B)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	TRCMUX	P15–17, P34–36 debug function selection	1 Debug 0 GPIO, etc.	1	R/W	
Performance analyzer control register (pMISC_PAC) Protected	00300016 (B)	D7–2	–	reserved	–	–	–	Do not access in the user program.
		D1	PARUN	Test bit	–	–	–	
		D0	PACLR	Test bit	–	–	–	
Boot register (pMISC_BOOT) Protected	00300018 (B)	D7	BOOT3	Boot mode indicator	BOOT[3:0] 1000 SPI 0100 NOR Flash/ROM 0010 reserved 0001 NAND Flash	X X X X	R/W	Depend on the BOOT1 and BOOT0 pin status at initial reset
		D6	BOOT2					
		D5	BOOT1					
		D4	BOOT0					
		D3–2	–					
D1	BOOT_ENA	#CE10 area boot enable	1 Internal 0 External	1	R/W			
D0	CE10_SIZE	#CE10 area size	1 16 bits 0 8 bits	1	R/W			
COROM switch register (pMISC_COROM) Protected	0030001A (B)	D7–1	–	reserved	–	–	–	Do not access in the user program.
		D0	COROM_SW	Test bit	–	–	–	
Misc protect register (pMISC_PROT)	00300020 (B)	D7	PROT7	Misc register protect flag	Writing 10010110 (0x96) removes the write protection of the Misc registers (0x300010–0x30001A). Writing another value set the write protection.	0 0 0 0 0 0 0 0	R/W	
		D6	PROT6					
		D5	PROT5					
		D4	PROT4					
		D3	PROT3					
		D2	PROT2					
		D1	PROT1					
		D0	PROT0					

0x300260–0x300268

Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Port input 0–1 interrupt priority register (pINT_PP01L)	00300260 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PP1L2	Port input 1 interrupt level	0 to 7	X	R/W		
		D5	PP1L1						
		D4	PP1L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PP0L2	Port input 0 interrupt level	0 to 7	X	R/W		
D1	PP0L1								
D0	PP0L0								
Port input 2–3 interrupt priority register (pINT_PP23L)	00300261 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PP3L2	Port input 3 interrupt level	0 to 7	X	R/W		
		D5	PP3L1						
		D4	PP3L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PP2L2	Port input 2 interrupt level	0 to 7	X	R/W		
D1	PP2L1								
D0	PP2L0								
Key input interrupt priority register (pINT_PK01L)	00300262 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PK1L2	Key input 1 interrupt level	0 to 7	X	R/W		
		D5	PK1L1						
		D4	PK1L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PK0L2	Key input 0 interrupt level	0 to 7	X	R/W		
D1	PK0L1								
D0	PK0L0								
HSDMA Ch.0–1 interrupt priority register (pINT_PHSD01L)	00300263 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PHSD1L2	HSDMA Ch.1 interrupt level	0 to 7	X	R/W		
		D5	PHSD1L1						
		D4	PHSD1L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PHSD0L2	HSDMA Ch.0 interrupt level	0 to 7	X	R/W		
D1	PHSD0L1								
D0	PHSD0L0								
HSDMA Ch.2–3 interrupt priority register (pINT_PHSD23L)	00300264 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PHSD3L2	HSDMA Ch.3 interrupt level	0 to 7	X	R/W		
		D5	PHSD3L1						
		D4	PHSD3L0						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	PHSD2L2	HSDMA Ch.2 interrupt level	0 to 7	X	R/W		
D1	PHSD2L1								
D0	PHSD2L0								
IDMA interrupt priority register (pINT_PDM)	00300265 (B)	D7–3	–	reserved	–	–	–	0 when being read.	
		D2	PDM2	IDMA interrupt level	0 to 7	X	R/W		
		D1	PDM1						
		D0	PDM0						
16-bit timer 0–1 interrupt priority register (pINT_P16T01)	00300266 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	P16T12	16-bit timer 1 interrupt level	0 to 7	X	R/W		
		D5	P16T11						
		D4	P16T10						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	P16T02	16-bit timer 0 interrupt level	0 to 7	X	R/W		
D1	P16T01								
D0	P16T00								
16-bit timer 2–3 interrupt priority register (pINT_P16T23)	00300267 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	P16T32	16-bit timer 3 interrupt level	0 to 7	X	R/W		
		D5	P16T31						
		D4	P16T30						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	P16T22	16-bit timer 2 interrupt level	0 to 7	X	R/W		
D1	P16T21								
D0	P16T20								
16-bit timer 4–5 interrupt priority register (pINT_P16T45)	00300268 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	P16T52	16-bit timer 5 interrupt level	0 to 7	X	R/W		
		D5	P16T51						
		D4	P16T50						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	P16T42	16-bit timer 4 interrupt level	0 to 7	X	R/W		
D1	P16T41								
D0	P16T40								

0x300269–0x300272

Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
LCDC, serial I/F Ch.0 interrupt priority register (pINT_PLCDC_PSI00)	00300269 (B)	D7	–	reserved	–	–	–	0 when being read.			
		D6	PSIO02	Serial interface Ch.0 interrupt level	0 to 7	X	R/W				
		D5	PSIO01			X					
		D4	PSIO00			X					
		D3	–	reserved	–	–	–	0 when being read.			
		D2	PLCDC2	LCDC interrupt level	0 to 7	X	R/W				
D1	PLCDC1	X									
D0	PLCDC0	X									
Serial I/F Ch.1, A/D interrupt priority register (pINT_PSI01_PAD)	0030026A (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									
RTC interrupt priority register (pINT_PRTC)	0030026B (B)	D7–3	–	reserved	–	–	–	Writing 1 not allowed.			
		D2	PRTC2	RTC interrupt level	0 to 7	X	R/W				
		D1	PRTC1			X					
		D0	PRTC0			X					
D7–3	–	reserved	–			–	–		0 when being read.		
Port input 4–5 interrupt priority register (pINT_PP45L)	0030026C (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 (pINT_PP67L)	0030026D (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, SPI interrupt priority register (pINT_PSI02_PSPI)	0030026E (B)	D7	–	reserved	–	–	–	0 when being read.			
		D6	PSPi2	SPI interrupt level	0 to 7	X	R/W				
		D5	PSPi1			X					
		D4	PSPi0			X					
		D3	–	reserved	–	–	–	0 when being read.			
		D2	PSIO22	Serial interface Ch.2 interrupt level	0 to 7	X	R/W				
D1	PSIO21	X									
D0	PSIO20	X									
Key input, port input 0–3 interrupt enable register (pINT_EK01_EP03)	00300270 (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 (pINT_EDMA)	00300271 (B)	D7–5	–	reserved	–	–	–	–	0 when being read.		
		D4	EIDMA	IDMA	1	Enabled	0	Disabled	0	R/W	
		D3	EHDM3	HSDMA Ch.3					0	R/W	
		D2	EHDM2	HSDMA Ch.2					0	R/W	
		D1	EHDM1	HSDMA Ch.1					0	R/W	
		D0	EHDM0	HSDMA Ch.0					0	R/W	
16-bit timer 0–1 interrupt enable register (pINT_E16T01)	00300272 (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.	

0x300273–0x300283

Interrupt Controller

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 2–3 interrupt enable register (pINT_E16T23)	00300273 (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 (pINT_E16T45)	00300274 (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.
Serial I/F Ch.0–1 interrupt enable register (pINT_ESIF01)	00300276 (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, RTC, A/D interrupt enable register (pINT_EP47_RTC_EAD)	00300277 (B)	D7	–	reserved					–	–	0 when being read.
		D6	EP7	Port input 7	1	Enabled	0	Disabled	0	R/W	
		D5	EP6	Port input 6					0	R/W	
		D4	EP5	Port input 5					0	R/W	
		D3	EP4	Port input 4					0	R/W	
		D2	ERTC	RTC					0	R/W	
		D1	EADE	A/D conversion completion					0	R/W	
		D0	EADC	A/D out-of-range					0	R/W	
LCDC interrupt enable register (pINT_ELCDC)	00300278 (B)	D7–2	–	reserved					–	–	0 when being read.
		D1	ELCDC	LCDC frame end	1	Enabled	0	Disabled	0	R/W	
		D0	–	reserved					–	–	Do not write 1.
Serial I/F Ch.2, SPI interrupt enable register (pINT_ESIF2_ESPI)	00300279 (B)	D7–6	–	reserved					–	–	0 when being read.
		D5	ESPITX	SPI transmit DMA	1	Enabled	0	Disabled	0	R/W	
		D4	ESPIRX	SPI receive DMA					0	R/W	
		D3	–	reserved					–	–	Do not write 1.
		D2	ESTX2	SIF Ch.2 transmit buffer empty	1	Enabled	0	Disabled	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 cause flag register (pINT_FK01_FP03)	00300280 (B)	D7–6	–	reserved					–	–	0 when being read.
		D5	FK1	Key input 1	1	Occurred	0	Not occurred	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 cause flag register (pINT_FDMA)	00300281 (B)	D7–5	–	reserved					–	–	0 when being read.
		D4	FIDMA	IDMA	1	Occurred	0	Not occurred	X	R/W	
		D3	FHDM3	HSDMA Ch.3					X	R/W	
		D2	FHDM2	HSDMA Ch.2					X	R/W	
		D1	FHDM1	HSDMA Ch.1					X	R/W	
		D0	FHDM0	HSDMA Ch.0					X	R/W	
16-bit timer 0–1 interrupt cause flag register (pINT_F16T01)	00300282 (B)	D7	F16TC1	16-bit timer 1 comparison A	1	Occurred	0	Not occurred	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	Occurred	0	Not occurred	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 cause flag register (pINT_F16T23)	00300283 (B)	D7	F16TC3	16-bit timer 3 comparison A	1	Occurred	0	Not occurred	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	Occurred	0	Not occurred	X	R/W	
		D2	F16TU2	16-bit timer 2 comparison B					X	R/W	
D1–0	–	reserved					–	–	0 when being read.		

0x300284–0x300293

Interrupt Controller

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
16-bit timer 4–5 interrupt cause flag register (pINT_F16T45)	00300284 (B)	D7	F16TC5	16-bit timer 5 comparison A	1	Occurred	0	Not occurred	X	R/W	0 when being read.
		D6	F16TU5	16-bit timer 5 comparison B					X	R/W	
		D5–4	–	reserved					–	–	
		D3	F16TC4	16-bit timer 4 comparison A	1	Occurred	0	Not occurred	X	R/W	
		D2	F16TU4	16-bit timer 4 comparison B					X	R/W	
D1–0	–	reserved					–	–			
Serial I/F Ch.0–1 interrupt cause flag register (pINT_FSIF01)	00300286 (B)	D7–6	–	reserved					–	–	0 when being read.
		D5	FSTX1	SIF Ch.1 transmit buffer empty	1	Occurred	0	Not occurred	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	
D7–2	–	reserved									–
Port input 4–7, RTC, A/D interrupt cause flag register (pINT_FP47_FRTC_FAD)	00300287 (B)	D7	–	reserved					–	–	0 when being read.
		D6	FP7	Port input 7	1	Occurred	0	Not occurred	X	R/W	
		D5	FP6	Port input 6					X	R/W	
		D4	FP5	Port input 5					X	R/W	
		D3	FP4	Port input 4					X	R/W	
		D2	FRTC	RTC					X	R/W	
		D1	FADE	A/D conversion completion					X	R/W	
		D0	FADC	A/D out-of-range					X	R/W	
D7–2	–	reserved									–
LCDC interrupt cause flag register (pINT_FLCDC)	00300288 (B)	D1	FLCDC	LCDC frame end	1	Occurred	0	Not occurred	X	R/W	
		D0	–	reserved					–	–	0 when being read.
Serial I/F Ch.2, SPI interrupt cause flag register (pINT_FSIF2_FSPI)	00300289 (B)	D7–6	–	reserved					–	–	0 when being read.
		D5	FSPITX	SPI transmit DMA	1	Occurred	0	Not occurred	X	R/W	
		D4	FSPIRX	SPI receive DMA					X	R/W	
		D3	–	reserved					–	–	0 when being read.
		D2	FSTX2	SIF Ch.2 transmit buffer empty	1	Occurred	0	Not occurred	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, HSDMA Ch.0–1, 16-bit timer 0 IDMA request register (pIDMAREQ_RP03_RHS_R16T0)	00300290 (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	HSDMA Ch.1					0	R/W	
		D4	RHDM0	HSDMA 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	
		D7	R16TC4	16-bit timer 4 comparison A					1	IDMA request	0
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, serial I/F Ch.0 IDMA request register (pIDMAREQ_R16T5_RSIF0)	00300292 (B)	D7	RSTX0	SIF Ch.0 transmit buffer empty	1	IDMA request	0	Interrupt request			
		D6	RSRX0	SIF Ch.0 receive buffer full					0	R/W	
		D5–2	–	reserved					–	–	0 when being read.
		D1	R16TC5	16-bit timer 5 comparison A	1	IDMA request	0	Interrupt request	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 (pIDMAREQ_RSIF1_RAD_RP47)	00300293 (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 conversion completion	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	
D7	R16TC5	16-bit timer 5 comparison A									–

0x300294–0x300298

Interrupt Controller

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks						
Port input 0–3, HSDMA Ch.0–1, 16-bit timer 0 IDMA enable register (pIDMAEN_DEP03_DEHS_DE16T0)	00300294 (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	HSDMA Ch.1					0	R/W					
		D4	DEHDM0	HSDMA 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 (pIDMAEN_DE16T14)	00300295 (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, serial I/F Ch.0 IDMA enable register (pIDMAEN_DE16T5_DESIF0)	00300296 (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					
		D5–2	–	reserved					–	–		0 when being read.			
		D1	DE16TC5	16-bit timer 5 comparison A					1	IDMA enabled		0	IDMA disabled	0	R/W
		D0	DE16TU5	16-bit timer 5 comparison B					1	IDMA enabled		0	IDMA disabled	0	R/W
Serial I/F Ch.1, A/D, port input 4–7 IDMA enable register (pIDMAEN_DESIF1_DEAD_DEP47)	00300297 (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 conversion completion					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					
HSDMA Ch.0–1 trigger set-up register (pHSDMA_HTGR1)	00300298 (B)	D7	HSD1S3	HSDMA Ch.1 trigger set-up	0	Software trigger	0	R/W							
		D6	HSD1S2		1	#DMAREQ1 input (falling edge)	0								
		D5	HSD1S1		2	#DMAREQ1 input (rising edge)	0								
		D4	HSD1S0		3	Port 1 input	0								
					4	Port 5 input	0								
					5	(reserved)									
					6	16-bit timer 1 compare B									
					7	16-bit timer 1 compare A									
					8	16-bit timer 5 compare B									
					9	I ² S right									
				A	SI/F Ch.1 Rx buffer full										
				B	SI/F Ch.1 Tx buffer empty										
				C	A/D conversion completion										
				D	Port 9 input (USB PDREQ)										
				E	Port 13 input										
		D3	HSD0S3	HSDMA Ch.0 trigger set-up	0	Software trigger	0	R/W							
		D2	HSD0S2		1	#DMAREQ0 input (falling edge)	0								
		D1	HSD0S1		2	#DMAREQ0 input (rising edge)	0								
		D0	HSD0S0		3	Port 0 input	0								
					4	Port 4 input	0								
5	(reserved)														
6	16-bit timer 0 compare B														
		7	16-bit timer 0 compare A												
		8	16-bit timer 4 compare B												
		9	I ² S left												
		A	SI/F Ch.0 Rx buffer full												
		B	SI/F Ch.0 Tx buffer empty												
		C	A/D conversion completion												
		D	Port 8 input (SPI interrupt)												
		E	Port 12 input												

0x300299–0x30029F **Interrupt Controller**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks							
HSDMA Ch.2-3 trigger set-up register (pHSDMA_HTGR2)	00300299 (B)	D7	HSD3S3	HSDMA Ch.3 trigger set-up	0	Software trigger	0	R/W							
		D6	HSD3S2		1	#DMAREQ3 input (falling edge)	0								
		D5	HSD3S1		2	#DMAREQ3 input (rising edge)	0								
		D4	HSD3S0		3	Port 3 input	0								
					4	Port 7 input									
					5	(reserved)									
					6	16-bit timer 3 compare B									
					7	16-bit timer 3 compare A									
					8	(reserved)									
					9	SPI Rx									
					A	(reserved)									
					B	(reserved)									
					C	A/D conversion completion									
					D	Port 11 input (DCSIO interrupt)									
		E	Port 15 input												
HSDMA software trigger register (pHSDMA_HSOFTTGR)	0030029A (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				
		LCDC, serial I/F Ch.2, SPI IDMA request register (pIDMAREQ_RLCDC_RSIF2_RSPI)	0030029B (B)	D7-6	–	reserved	–	–		–	–	0 when being read.			
				D5	RSPITX	SPI transmit DMA	1	IDMA request		0	Interrupt request		0	R/W	
				D4	RSPIRX	SPI receive DMA							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	RLCDC	LCDC frame end							0	R/W	
				D0	–	reserved	–	–		–	–		–	–	
				LCDC, serial I/F Ch.2, SPI IDMA enable register (pIDMAEN_DELCDC_HESIF2_DESPI)	0030029C (B)	D7-6	–	reserved		–	–		–	–	0 when being read.
						D5	DESPITX	SPI transmit DMA		1	IDMA enabled		0	IDMA disabled	
D4	DESPIRX					SPI receive DMA					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	DELCDC	LCDC frame end							0	R/W					
D0	–	reserved	–			–	–	–	–	–					
Flag set/reset method select register (pRST_RESET)	0030029F (B)	D7-3	–	reserved	–	–	–	–	0 when being read.						
		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	Cause-of-interrupt flag reset method selection	1	Reset only	0	RD/WR		1	R/W				

0x3002A0–0x3002AA

Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Port input 8–9 interrupt priority register (pINT_PP89L)	003002A0 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PP9L2	Port input 9/USB PDREQ interrupt level	0 to 7	X	R/W		
		D5	PP9L1		X				
		D4	PP9L0		X				
		D3	–	reserved	–	–	–	0 when being read.	
		D2	PP8L2	Port input 8/SPI interrupt level	0 to 7	X	R/W		
		D1	PP8L1		X				
D0	PP8L0	X							
Port input 10–11 interrupt priority register (pINT_PP1011L)	003002A1 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PP11L2	Port input 11/DCSIO interrupt level	0 to 7	X	R/W		
		D5	PP11L1		X				
		D4	PP11L0		X				
		D3	–	reserved	–	–	–	0 when being read.	
		D2	PP10L2	Port input 10/USB interrupt level	0 to 7	X	R/W		
		D1	PP10L1		X				
D0	PP10L0	X							
Port input 12–13 interrupt priority register (pINT_PP1213L)	003002A2 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PP13L2	Port input 13 interrupt level	0 to 7	X	R/W		
		D5	PP13L1		X				
		D4	PP13L0		X				
		D3	–	reserved	–	–	–	0 when being read.	
		D2	PP12L2	Port input 12 interrupt level	0 to 7	X	R/W		
		D1	PP12L1		X				
D0	PP12L0	X							
Port input 14–15 interrupt priority register (pINT_PP1415L)	003002A3 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	PP15L2	Port input 15 interrupt level	0 to 7	X	R/W		
		D5	PP15L1		X				
		D4	PP15L0		X				
		D3	–	reserved	–	–	–	0 when being read.	
		D2	PP14L2	Port input 14 interrupt level	0 to 7	X	R/W		
		D1	PP14L1		X				
D0	PP14L0	X							
I ² S interrupt priority register (pINT_PI2S)	003002A4 (B)	D7–3	–	reserved	–	–	–	0 when being read.	
		D2	PI2S2	I ² S interrupt level	0 to 7	X	R/W		
		D1	PI2S1		X				
		D0	PI2S0		X				
Port input 8–15 interrupt enable register (pINT_EP815)	003002A6 (B)	D7	EP15	Port input 15	1 Enabled	0 Disabled	0	R/W	
		D6	EP14	Port input 14			0	R/W	
		D5	EP13	Port input 13			0	R/W	
		D4	EP12	Port input 12			0	R/W	
		D3	EP11	Port input 11/DCSIO			0	R/W	
		D2	EP10	Port input 10/USB			0	R/W	
		D1	EP9	Port input 9/USB PDREQ			0	R/W	
		D0	EP8	Port input 8/SPI			0	R/W	
I ² S interrupt enable register (pINT_EI2S)	003002A7 (B)	D7–3	–	reserved	–	–	–	0 when being read.	
		D2	EI2S	I ² S	1 Enabled	0 Disabled	0	R/W	
		D1–0	–	reserved	–	–	–	0 when being read.	
Port input 8–15 interrupt cause flag register (pINT_FP815)	003002A9 (B)	D7	FP15	Port input 15	1 Occurred	0 Not occurred	X	R/W	
		D6	FP14	Port input 14			X	R/W	
		D5	FP13	Port input 13			X	R/W	
		D4	FP12	Port input 12			X	R/W	
		D3	FP11	Port input 11/DCSIO			X	R/W	
		D2	FP10	Port input 10/USB			X	R/W	
		D1	FP9	Port input 9/USB PDREQ			X	R/W	
		D0	FP8	Port input 8/SPI			X	R/W	
I ² S interrupt cause flag register (pINT_FI2S)	003002AA (B)	D7–3	–	reserved	–	–	–	0 when being read.	
		D2	FI2S	I ² S	1 Occurred	0 Not occurred	X	R/W	
		D1–0	–	reserved	–	–	–	0 when being read.	

0x3002AC–0x3002AF **Interrupt Controller**

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Port input 8–15 IDMA request register (pIDMAREQ _RP815)	003002AC (B)	D7	RP15	Port input 15	1	IDMA request	0	Interrupt request	0	R/W	
		D6	RP14	Port input 14					0	R/W	
		D5	RP13	Port input 13					0	R/W	
		D4	RP12	Port input 12					0	R/W	
		D3	RP11	Port input 11/DCSIO					0	R/W	
		D2	RP10	Port input 10/USB					0	R/W	
		D1	RP9	Port input 9/USB PDREQ					0	R/W	
		D0	RP8	Port input 8/SPI					0	R/W	
I ² S IDMA request register (pIDMAREQ_RI2S)	003002AD (B)	D7–1	–	reserved	–		–	–	0 when being read.		
		D0	RI2S	I ² S	1	IDMA request	0	Interrupt request	0	R/W	
Port input 8–15 IDMA enable register (pIDMAEN_DEP815)	003002AE (B)	D7	DEP15	Port input 15	1	IDMA enabled	0	IDMA disabled	0	R/W	
		D6	DEP14	Port input 14					0	R/W	
		D5	DEP13	Port input 13					0	R/W	
		D4	DEP12	Port input 12					0	R/W	
		D3	DEP11	Port input 11/DCSIO					0	R/W	
		D2	DEP10	Port input 10/USB					0	R/W	
		D1	DEP9	Port input 9/USB PDREQ					0	R/W	
		D0	DEP8	Port input 8/SPI					0	R/W	
I ² S IDMA enable register (pIDMAEN_DEI2S)	003002AF (B)	D7–1	–	reserved	–		–	–	0 when being read.		
		D0	DEI2S	I ² S	1	IDMA enabled	0	IDMA disabled	0	R/W	

0x300300–0x300318

Card Interface

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Card I/F area configuration register (pCARDSETUP)	00300300 (B)	D7	CARDPC21	PC card 2 area configuration	bit[1:0]	#CE area	0	R/W		
		D6	CARDPC20				0			
		D5	CARDPC11	PC card 1 area configuration	11	#CE11 (Area 11, 12)	0	R/W		
		D4	CARDPC10				0			
		D3	CARDCF1	CF area configuration	10	#CE9 (Area 9, 22)	0	R/W		
		D2	CARDCF0				0			
D1	CARDSMT1	SmartMedia/NAND flash area configuration	01	#CE7 (Area 7, 19)	0	R/W				
D0	CARDSMT0				0					
Card I/F output port configuration register (pCARDFUNCSEL05)	00300302 (B)	D7–6	–	reserved	–	–	–	0 when being read.		
		D5	CARDIO5	CARD5 port function select	1	#CFCE2	0	#WE	0	R/W
		D4	CARDIO4	CARD4 port function select	1	#CFCE1	0	#OE	0	R/W
		D3	CARDIO3	CARD3 port function select	1	#SMWR	0	#IOWR	0	R/W
		D2	CARDIO2	CARD2 port function select	1	#SMRD	0	#IORD	0	R/W
		D1	CARDIO1	CARD1 port function select	1	#CFCE2	0	#SMWR	0	R/W
D0	CARDIO0	CARD0 port function select	1	#CFCE1	0	#SMRD	0	R/W		
ECC trigger area select register (pECCTRIGSEL)	00300310 (B)	D7–3	–	reserved	–	–	–	0 when being read.		
		D2	ECCSEL2	External memory area select for triggering ECC	ECCSEL[2:0]	#CE area	0	R/W		
		D1	ECCSEL1				111	#CE11	0	
		D0	ECCSEL0				110	#CE10	0	
							101	#CE9		
							100	#CE8		
		011	#CE7							
		010	#CE6							
		001	#CE5							
		000	#CE4							
ECC reset/ready register (pECCRSTRDY)	00300311 (B)	D7–2	–	reserved	–	–	–	0 when being read.		
		D1	ECCRDY1	Area 1 parity data ready status	1	Ready	0	Busy	0	R
		D0	ECCRST	ECC circuit reset	1	Reset	0	Invalid	–	W
ECC enable register (pECCENA)	00300312 (B)	D7–1	–	reserved	–	–	–	0 when being read.		
		D0	ECCEN	ECC circuit enable	1	Enabled	0	Disabled	0	R/W
ECC mode register (pECCMD)	00300313 (B)	D7–1	–	reserved	–	–	–	0 when being read.		
		D0	MODE	Card device mode	1	16 bits	0	8 bits	0	R/W
Area 0 ECC column parity data register (pECC0CP)	00300314 (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
D0	–	Unused bit	–	1	R					
Area 0 ECC line parity register 0 (pECC0LPL)	00300316 (B)	D7	LP07	Area 0 ECC line parity LP00 = LSB	0x0 to 0xFF (low-order 8 bits)	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 (pECC0LPH)	00300317 (B)	D7	LP015	Area 0 ECC line parity LP015 = MSB	0x0 to 0xFF (high-order 8 bits)	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 (pECC1CP)	00300318 (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
D0	–	Unused bit	–	1	R					

0x30031A–0x30031B **Card Interface**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Area 1 ECC line parity register 0 (pECC1LPL)	0030031A (B)	D7	LP17	Area 1 ECC line parity LP10 = LSB	0x0 to 0xFF (low-order 8 bits)	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 (pECC1LPH)	0030031B (B)	D7	LP115	Area 1 ECC line parity LP115 = MSB	0x0 to 0xFF (high-order 8 bits)	1	R	
		D6	LP114			1		
		D5	LP113			1		
		D4	LP112			1		
		D3	LP111			1		
		D2	LP110			1		
		D1	LP19			1		
		D0	LP18			1		

0x300380–0x300387

I/O Ports

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P0 port data register (pP0_P0D)	00300380 (B)	D7	P07D	P07 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D6	P06D	P06 I/O port data					Ext.	R/W	
		D5	P05D	P05 I/O port data					Ext.	R/W	
		D4	P04D	P04 I/O port data					Ext.	R/W	
		D3	P03D	P03 I/O port data					Ext.	R/W	
		D2	P02D	P02 I/O port data					Ext.	R/W	
		D1	P01D	P01 I/O port data					Ext.	R/W	
		D0	P00D	P00 I/O port data					Ext.	R/W	
P0 I/O control register (pP0_IOC0)	00300381 (B)	D7	IOC07	P07 I/O control	1	Output	0	Input	0	R/W	
		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 port data register (pP1_P1D)	00300382 (B)	D7	P17D	P17 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D6	P16D	P16 I/O port data					Ext.	R/W	
		D5	P15D	P15 I/O port data					Ext.	R/W	
		D4	P14D	P14 I/O port data					Ext.	R/W	
		D3	P13D	P13 I/O port data					Ext.	R/W	
		D2	P12D	P12 I/O port data					Ext.	R/W	
		D1	P11D	P11 I/O port data					Ext.	R/W	
		D0	P10D	P10 I/O port data					Ext.	R/W	
P1 I/O control register (pP1_IOC1)	00300383 (B)	D7	IOC17	P17 I/O control	1	Output	0	Input	0	R/W	
		D6	IOC16	P16 I/O control					0	R/W	
		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	
P2 port data register (pP2_P2D)	00300384 (B)	D7	P27D	P27 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D6	P26D	P26 I/O port data					Ext.	R/W	
		D5	P25D	P25 I/O port data					Ext.	R/W	
		D4	P24D	P24 I/O port data					Ext.	R/W	
		D3	P23D	P23 I/O port data					Ext.	R/W	
		D2	P22D	P22 I/O port data					Ext.	R/W	
		D1	P21D	P21 I/O port data					Ext.	R/W	
		D0	P20D	P20 I/O port data					Ext.	R/W	
P2 I/O control register (pP2_IOC2)	00300385 (B)	D7	IOC27	P27 I/O control	1	Output	0	Input	0	R/W	
		D6	IOC26	P26 I/O control					0	R/W	
		D5	IOC25	P25 I/O control					0	R/W	
		D4	IOC24	P24 I/O control					0	R/W	
		D3	IOC23	P23 I/O control					0	R/W	
		D2	IOC22	P22 I/O control					0	R/W	
		D1	IOC21	P21 I/O control					0	R/W	
		D0	IOC20	P20 I/O control					0	R/W	
P3 port data register (pP3_P3D)	00300386 (B)	D7	–	reserved	1	High	0	Low	–	–	0 when being read.
		D6	P36D	P36 I/O port data					Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D5	P35D	P35 I/O port data					Ext.	R/W	
		D4	P34D	P34 I/O port data					Ext.	R/W	
		D3	P33D	P33 I/O port data					Ext.	R/W	
		D2	P32D	P32 I/O port data					Ext.	R/W	
		D1	P31D	P31 I/O port data					Ext.	R/W	
		D0	P30D	P30 I/O port data					Ext.	R/W	
P3 I/O control register (pP3_IOC3)	00300387 (B)	D7	–	reserved	1	Output	0	Input	–	–	0 when being read.
		D6	IOC36	P36 I/O control					0	R/W	
		D5	IOC35	P35 I/O control					0	R/W	
		D4	IOC34	P34 I/O control					0	R/W	
		D3	IOC33	P33 I/O control					0	R/W	
		D2	IOC32	P32 I/O control					0	R/W	
		D1	IOC31	P31 I/O control					0	R/W	
		D0	IOC30	P30 I/O control					0	R/W	

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks			
P4 port data register (pP4_P4D)	00300388 (B)	D7	P47D	P47 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.	
		D6	P46D	P46 I/O port data					Ext.	R/W		
		D5	P45D	P45 I/O port data					Ext.	R/W		
		D4	P44D	P44 I/O port data					Ext.	R/W		
		D3	P43D	P43 I/O port data					Ext.	R/W		
		D2	P42D	P42 I/O port data					Ext.	R/W		
		D1	P41D	P41 I/O port data					Ext.	R/W		
		D0	P40D	P40 I/O port data					Ext.	R/W		
P4 I/O control register (pP4_IOC4)	00300389 (B)	D7	IOC47	P47 I/O control	1	Output	0	Input	0	R/W		
		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 port data register (pP5_P5D)	0030038A (B)	D7	P57D	P57 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.	
		D6	P56D	P56 I/O port data					Ext.	R/W		
		D5	P55D	P55 I/O port data					Ext.	R/W		
		D4	P54D	P54 I/O port data					Ext.	R/W		
		D3	P53D	P53 I/O port data					Ext.	R/W		
		D2	P52D	P52 I/O port data					Ext.	R/W		
		D1	P51D	P51 I/O port data					Ext.	R/W		
		D0	P50D	P50 I/O port data					Ext.	R/W		
P5 I/O control register (pP5_IOC5)	0030038B (B)	D7	IOC57	P57 I/O control	1	Output	0	Input	0	R/W		
		D6	IOC56	P56 I/O control					0	R/W		
		D5	IOC55	P55 I/O control					0	R/W		
		D4	IOC54	P54 I/O control					0	R/W		
		D3	IOC53	P53 I/O control					0	R/W		
		D2	IOC52	P52 I/O control					0	R/W		
		D1	IOC51	P51 I/O control					0	R/W		
		D0	IOC50	P50 I/O control					0	R/W		
P6 port data register (pP6_P6D)	0030038C (B)	D7	P67D	P67 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.	
		D6	P66D	P66 I/O port data					Ext.	R/W		
		D5	P65D	P65 I/O port data					Ext.	R/W		
		D4	P64D	P64 I/O port data					Ext.	R/W		
		D3	P63D	P63 I/O port data					Ext.	R/W		
		D2	P62D	P62 I/O port data					Ext.	R/W		
		D1	P61D	P61 I/O port data					Ext.	R/W		
		D0	P60D	P60 I/O port data					Ext.	R/W		
P6 I/O control register (pP6_IOC6)	0030038D (B)	D7	IOC67	P67 I/O control	1	Output	0	Input	0	R/W		
		D6	IOC66	P66 I/O control					0	R/W		
		D5	IOC65	P65 I/O control					0	R/W		
		D4	IOC64	P64 I/O control					0	R/W		
		D3	IOC63	P63 I/O control					0	R/W		
		D2	IOC62	P62 I/O control					0	R/W		
		D1	IOC61	P61 I/O control					0	R/W		
		D0	IOC60	P60 I/O control					0	R/W		
P7 port data register (pP7_P7D)	0030038E (B)	D7–5	–	reserved	1	High	0	Low	–	–	0 when being read.	
		D4	P74D	P74 input port data					Ext.	R		Ext.: The initial value depends on the external pin status.
		D3	P73D	P73 input port data					Ext.	R		
		D2	P72D	P72 input port data					Ext.	R		
		D1	P71D	P71 input port data					Ext.	R		
D0	P70D	P70 input port data	Ext.	R								
P8 port data register (pP8_P8D)	00300390 (B)	D7–6	–	reserved	1	High	0	Low	–	–	0 when being read.	
		D5	P85D	P85 I/O port data					Ext.	R/W		Ext.: The initial value depends on the external pin status.
		D4	P84D	P84 I/O port data					Ext.	R/W		
		D3	P83D	P83 I/O port data					Ext.	R/W		
		D2	P82D	P82 I/O port data					Ext.	R/W		
		D1	P81D	P81 I/O port data					Ext.	R/W		
D0	P80D	P80 I/O port data	Ext.	R/W								
P8 I/O control register (pP8_IOC8)	00300391 (B)	D7–6	–	reserved	1	Output	0	Input	–	–	0 when being read.	
		D5	IOC85	P85 I/O control					0	R/W		
		D4	IOC84	P84 I/O control					0	R/W		
		D3	IOC83	P83 I/O control					0	R/W		
		D2	IOC82	P82 I/O control					0	R/W		
		D1	IOC81	P81 I/O control					0	R/W		
D0	IOC80	P80 I/O control	0	R/W								

0x300392–0x3003A1

I/O Ports

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
P9 port data register (pP9_P9D)	00300392 (B)	D7	P97D	P97 I/O port data	1	High	0	Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D6	P96D	P96 I/O port data					Ext.	R/W	
		D5	P95D	P95 I/O port data					Ext.	R/W	
		D4	P94D	P94 I/O port data					Ext.	R/W	
		D3	P93D	P93 I/O port data					Ext.	R/W	
		D2	P92D	P92 I/O port data					Ext.	R/W	
		D1	P91D	P91 I/O port data					Ext.	R/W	
		D0	P90D	P90 I/O port data					Ext.	R/W	
P9 I/O control register (pP9_IOC9)	00300393 (B)	D7	IOC97	P97 I/O control	1	Output	0	Input	0	R/W	
		D6	IOC96	P96 I/O control					0	R/W	
		D5	IOC95	P95 I/O control					0	R/W	
		D4	IOC94	P94 I/O control					0	R/W	
		D3	IOC93	P93 I/O control					0	R/W	
		D2	IOC92	P92 I/O control					0	R/W	
		D1	IOC91	P91 I/O control					0	R/W	
		D0	IOC90	P90 I/O control					0	R/W	
P00–P03 port function select register (pP0_03_CFP)	003003A0 (B)	D7	CFP031	P03 port extended function	CFP03[1:0]	Function	0	R/W	reserved		
		D6	CFP030							11	reserved
										10	#DMAEND3
										01	#SRDY0
				00	P03						
		D5	CFP021	P02 port extended function	CFP02[1:0]	Function	0	R/W	reserved		
		D4	CFP020							11	reserved
										10	#DMAEND2
										01	#SCLK0
				00	P02						
		D3	CFP011	P01 port extended function	CFP01[1:0]	Function	0	R/W	reserved		
		D2	CFP010							11	reserved
		10	#DMAACK3								
		01	SOUT0								
		00	P01								
D1	CFP001	P00 port extended function	CFP00[1:0]	Function	0	R/W	reserved				
D0	CFP000							11	reserved		
								10	#DMAACK2		
								01	SIN0		
		00	P00								
P04–P07 port function select register (pP0_47_CFP)	003003A1 (B)	D7	CFP071	P07 port extended function	CFP07[1:0]	Function	0	R/W	reserved		
		D6	CFP070							11	reserved
										10	I2S_MCLK
										01	#SRDY1
				00	P07						
		D5	CFP061	P06 port extended function	CFP06[1:0]	Function	0	R/W	reserved		
		D4	CFP060							11	reserved
										10	I2S_SCK
										01	#SCLK1
				00	P06						
		D3	CFP051	P05 port extended function	CFP05[1:0]	Function	0	R/W	reserved		
		D2	CFP050							11	reserved
		10	I2S_WS								
		01	SOUT1								
		00	P05								
D1	CFP041	P04 port extended function	CFP04[1:0]	Function	0	R/W	reserved				
D0	CFP040							11	reserved		
								10	I2S_SDO		
								01	SIN1		
		00	P04								

0x3003A2–0x3003A4

I/O Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P10–P13 port function select register (pP1_03_CFP)	003003A2 (B)	D7 D6	CFP131	P13 port extended function	CFP13[1:0]	Function	0	R/W	
			CFP130		11	#DMAACK1	0		
					10	#SRDY0			
					01	TM3			
			00	P13					
		D5 D4	CFP121 CFP120	P12 port extended function	CFP12[1:0]	Function	0	R/W	
					11	#DMAACK0	0		
					10	#SCLK0			
					01	TM2			
			00	P12					
		D3 D2	CFP111 CFP110	P11 port extended function	CFP11[1:0]	Function	0	R/W	
					11	#DMAEND1	0		
	10				SOUT0				
	01				TM1				
		00	P11						
	D1 D0	CFP101 CFP100	P10 port extended function	CFP10[1:0]	Function	0	R/W		
				11	#DMAEND0	0			
				10	SIN0				
				01	TM0				
		00	P10						
P14–P17 port function select register (pP1_47_CFP)	003003A3 (B)	D7 D6	CFP171	P17 port extended function	CFP17[1:0]	Function	0	R/W	When TRCMUX (D0/0x300014) is set to 1 (default), this register becomes ineffective and the port is configured for debugging.
			CFP170		11	TFT_CTL2	0		
					10	#SRDY1			
					01	DCSIO1			
			00	P17					
		D5 D4	CFP161 CFP160	P16 port extended function	CFP16[1:0]	Function	0	R/W	
					11	TFT_CTL3	0		
					10	#SCLK1			
					01	DCSIO0			
			00	P16					
		D3 D2	CFP151 CFP150	P15 port extended function	CFP15[1:0]	Function	0	R/W	
					11	TFT_CTL0	0		
	10				SOUT1				
	01				TM5				
		00	P15						
	D1 D0	CFP141 CFP140	P14 port extended function	CFP14[1:0]	Function	0	R/W		
				11	reserved	0			
				10	SIN1				
				01	TM4				
		00	P14						
P20–P23 port function select register (pP2_03_CFP)	003003A4 (B)	D7 D6	CFP231	P23 port extended function	CFP23[1:0]	Function	0	R/W	
			CFP230		11	reserved	0		
					10	TFT_CTL1			
					01	#SDRAS			
			00	P23					
		D5 D4	CFP221 CFP220	P22 port extended function	CFP22[1:0]	Function	0	R/W	
					1*	reserved	0		
					01	#SDCS			
					00	P22			
		D3 D2	CFP211 CFP210	P21 port extended function	CFP21[1:0]	Function	0	R/W	
					1*	reserved	1		
	01				SDCLK				
	00				P21				
	D1 D0	CFP201 CFP200	P20 port extended function	CFP20[1:0]	Function	0	R/W		
				1*	reserved	0			
				01	SDCKE				
				00	P20				

0x3003A5–0x3003A8

I/O Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P24–P27 port function select register (pP2_47_CFP)	003003A5 (B)	D7 D6	CFP271	P27 port extended function	CFP27[1:0]	Function	0	R/W	
			CFP270		1*	reserved	0		
					01	DQMH			
		00	P27						
		D5 D4	CFP261	P26 port extended function	CFP26[1:0]	Function	0	R/W	
			CFP260		1*	reserved	0		
					01	DQML			
		00	P26						
		D3 D2	CFP251	P25 port extended function	CFP25[1:0]	Function	0	R/W	
			CFP250		1*	reserved	0		
					01	#SDWE			
		00	P25						
D1 D0	CFP241	P24 port extended function	CFP24[1:0]	Function	0	R/W			
	CFP240		1*	reserved	0				
			01	#SDCAS					
00	P24								
P30–P33 port function select register (pP3_03_CFP)	003003A6 (B)	D7 D6	CFP331	P33 port extended function	CFP33[1:0]	Function	0	R/W	
			CFP330		11	reserved	0		
					10	#DMAREQ3			
					01	CARD5			
		00	P33						
		D5 D4	CFP321	P32 port extended function	CFP32[1:0]	Function	0	R/W	
			CFP320		11	reserved	0		
					10	#DMAREQ2			
					01	CARD4			
		00	P32						
		D3 D2	CFP311	P31 port extended function	CFP31[1:0]	Function	0	R/W	
			CFP310		11	reserved	0		
10	#DMAREQ1								
01	CARD3								
00	P31								
D1 D0	CFP301	P30 port extended function	CFP30[1:0]	Function	0	R/W			
	CFP300		11	reserved	0				
			10	#DMAREQ0					
			01	CARD2					
00	P30								
P34–P36 port function select register (pP3_46_CFP)	003003A7 (B)	D7–6	–	reserved	–	–	–	0 when being read.	
		D5 D4	CFP361	P36 port extended function	CFP36[1:0]	Function	0	R/W	When TRCMUX (D0/0x300014) is set to 1 (default), this register becomes ineffective and the port is configured for debugging.
			CFP360		1*	reserved	0		
		01	P36						
		00	DST2						
		D3 D2	CFP351	P35 port extended function	CFP35[1:0]	Function	0	R/W	
			CFP350		1*	reserved	0		
01	P35								
00	DCLK								
D1 D0	CFP341	P34 port extended function	CFP34[1:0]	Function	0	R/W			
	CFP340		1*	reserved	0				
			01	P34					
00	DSIO								
P40–P43 port function select register (pP4_03_CFP)	003003A8 (B)	D7 D6	CFP431	P43 port extended function	CFP43[1:0]	Function	0	R/W	
			CFP430		11	reserved	0		
					10	FPDAT9			
					01	P43			
		00	A21						
		D5 D4	CFP421	P42 port extended function	CFP42[1:0]	Function	0	R/W	
			CFP420		11	reserved	0		
					10	FPDAT8			
					01	P42			
		00	A22						
		D3 D2	CFP411	P41 port extended function	CFP41[1:0]	Function	0	R/W	
			CFP410		11	EXCL3			
10	#SDRAS								
01	P41								
00	A23								
D1 D0	CFP401	P40 port extended function	CFP40[1:0]	Function	0	R/W			
	CFP400		11	EXCL4					
			10	#SDCAS					
			01	P40					
00	A24								

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P44–P47 port function select register (pP4_47_CFP)	003003A9 (B)	D7	CFP471	P47 port extended function	CFP47[1:0]	Function	0	R/W
		D6	CFP470		1*	reserved	0	
					01	P47		
					00	A11		
		D5	CFP461	P46 port extended function	CFP46[1:0]	Function	0	R/W
	D4	CFP460	11		reserved	0		
			10		TFT_CTL2			
			01		P46			
			00	A18				
		D3	CFP451	P45 port extended function	CFP45[1:0]	Function	0	R/W
	D2	CFP450	11		reserved	0		
			10		FPDAT11			
		01	P45					
		00	A19					
	D1	CFP441	P44 port extended function	CFP44[1:0]	Function	0	R/W	
D0	CFP440	11		reserved	0			
		10		FPDAT10				
		01		P44				
		00	A20					
P50–P53 port function select register (pP5_03_CFP)	003003AA (B)	D7	CFP531	P53 port extended function	CFP53[1:0]	Function	0	R/W
		D6	CFP530		11	reserved	0	
					10	SDA10		
					01	P53		
			00	#CE7				
		D5	CFP521	P52 port extended function	CFP52[1:0]	Function	0	R/W
	D4	CFP520	11		CMU_CLK	0		
			10		#CE6			
			01		BCLK			
			00	P52				
		D3	CFP511	P51 port extended function	CFP51[1:0]	Function	0	R/W
	D2	CFP510	11		reserved	0		
		10	CARD1					
		01	P51					
		00	#CE5					
	D1	CFP501	P50 port extended function	CFP50[1:0]	Function	0	R/W	
D0	CFP500	11		reserved	0			
		10		CARD0				
		01		P50				
		00	#CE4					
P54–P57 port function select register (pP5_47_CFP)	003003AB (B)	D7	CFP571	P57 port extended function	CFP57[1:0]	Function	0	R/W
		D6	CFP570		1*	reserved	0	
					01	P57		
					00	#CE10		
		D5	CFP561	P56 port extended function	CFP56[1:0]	Function	0	R/W
	D4	CFP560	1*		reserved	0		
			01		P56			
			00		#CE11			
		D3	CFP551	P55 port extended function	CFP55[1:0]	Function	0	R/W
	D2	CFP550	11		reserved	0		
			10		CARD0			
			01		P55			
		00	#CE9					
	D1	CFP541	P54 port extended function	CFP54[1:0]	Function	0	R/W	
D0	CFP540	11		reserved	0			
		10		CARD1				
		01		P54				
		00	#CE8					

0x3003AC–0x3003AF

I/O Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P60–P63 port function select register (pP6_03_CFP)	003003AC (B)	D7 D6	CFP631 CFP630	P63 port extended function	CFP63[1:0]	Function	0	R/W	
					11	#WDT_NMI	0		
					10	WDT_CLK			
					01	#SRDY2			
		D5 D4	CFP621 CFP620	P62 port extended function	CFP62[1:0]	Function	0	R/W	
					11	CMU_CLK	0		
					10	#ADTRG			
					01	#SCLK2			
		D3 D2	CFP611 CFP610	P61 port extended function	CFP61[1:0]	Function	0	R/W	
					11	EXCL1	0		
					10	DCSIO1			
					01	SOUT2			
D1 D0	CFP601 CFP600	P60 port extended function	CFP60[1:0]	Function	0	R/W			
			11	EXCL0	0				
			10	DCSIO0					
			01	SIN2					
P64–P67 port function select register (pP6_47_CFP)	003003AD (B)	D7 D6	CFP671 CFP670	P67 port extended function	CFP67[1:0]	Function	0	R/W	
					11	reserved	0		
					10	FPDAT10			
					01	SPL_CLK			
		D5 D4	CFP661 CFP660	P66 port extended function	CFP66[1:0]	Function	0	R/W	
					11	reserved	0		
					10	FPDAT9			
					01	SDO			
		D3 D2	CFP651 CFP650	P65 port extended function	CFP65[1:0]	Function	0	R/W	
					11	reserved	0		
					10	FPDAT8			
					01	SDI			
D1 D0	CFP641 CFP640	P64 port extended function	CFP64[1:0]	Function	0	R/W			
			11	reserved	0				
			10	EXCL2					
			01	#WAIT					
P70–P73 port function select register (pP7_03_CFP)	003003AE (B)	D7 D6	CFP731 CFP730	P73 port extended function	CFP73[1:0]	Function	0	R/W	
					1*	reserved	0		
					01	AIN3			
					00	P73			
		D5 D4	CFP721 CFP720	P72 port extended function	CFP72[1:0]	Function	0	R/W	
					1*	reserved	0		
					01	AIN2			
					00	P72			
		D3 D2	CFP711 CFP710	P71 port extended function	CFP71[1:0]	Function	0	R/W	
					1*	reserved	0		
					01	AIN1			
					00	P71			
D1 D0	CFP701 CFP700	P70 port extended function	CFP70[1:0]	Function	0	R/W			
			1*	reserved	0				
			01	AIN0					
			00	P70					
P74 port function select register (pP7_4_CFP)	003003AF (B)	D7–2	–	reserved	–	–	–	0 when being read.	
		D1 D0	CFP741 CFP740	P74 port extended function	CFP74[1:0]	Function	0	R/W	
11	reserved				0				
10	EXCL5								
01	AIN4								
					00	P74			

0x3003B0–0x3003B3

I/O Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P80–P83 port function select register (pP8_03_CFP)	003003B0 (B)	D7 D6	CFP831	P83 port extended function	CFP83[1:0]	Function	0	R/W	
			CFP830		11	BCLK	0		
					10	TFT_CTL1			
					01	FPDRDY			
			00	P83					
		D5 D4	CFP821	P82 port extended function	CFP82[1:0]	Function	0	R/W	
	CFP820		1*		reserved	0			
			01	FPSHIFT					
			00	P82					
		D3 D2	CFP811	P81 port extended function	CFP81[1:0]	Function	0	R/W	
	CFP810		1*		reserved	0			
			01		FPLINE				
		00	P81						
	D1 D0	CFP801	P80 port extended function	CFP80[1:0]	Function	0	R/W		
CFP800		1*		reserved	0				
		01		FPFRAME					
		00		P80					
P84–P85 port function select register (pP8_45_CFP)	003003B1 (B)	D7–4	–	reserved	–	–	–	0 when being read.	
			D3 D2	CFP851 CFP850	P85 port extended function	CFP85[1:0]	Function	0	R/W
					1*	reserved	0		
					01	DCSIO1			
					00	P85			
		D1 D0	CFP841	P84 port extended function	CFP84[1:0]	Function	0	R/W	
CFP840	11		reserved		0				
	10		FPDAT11						
	01		DCSIO0						
				00	P84				
P90–P93 port function select register (pP9_03_CFP)	003003B2 (B)	D7 D6	CFP931	P93 port extended function	CFP93[1:0]	Function	0	R/W	
			CFP930		1*	reserved	0		
					01	FPDAT3			
					00	P93			
		D5 D4	CFP921	P92 port extended function	CFP92[1:0]	Function	0	R/W	
	CFP920		1*		reserved	0			
			01	FPDAT2					
			00	P92					
		D3 D2	CFP911	P91 port extended function	CFP91[1:0]	Function	0	R/W	
	CFP910		1*		reserved	0			
			01		FPDAT1				
			00	P91					
	D1 D0	CFP901	P90 port extended function	CFP90[1:0]	Function	0	R/W		
CFP900		1*		reserved	0				
		01		FPDAT0					
		00		P90					
P94–P97 port function select register (pP9_47_CFP)	003003B3 (B)	D7 D6	CFP971	P97 port extended function	CFP97[1:0]	Function	0	R/W	
			CFP970		1*	reserved	0		
					01	FPDAT7			
					00	P97			
		D5 D4	CFP961	P96 port extended function	CFP96[1:0]	Function	0	R/W	
	CFP960		1*		reserved	0			
			01	FPDAT6					
			00	P96					
		D3 D2	CFP951	P95 port extended function	CFP95[1:0]	Function	0	R/W	
	CFP950		1*		reserved	0			
			01		FPDAT5				
			00	P95					
	D1 D0	CFP941	P94 port extended function	CFP94[1:0]	Function	0	R/W		
CFP940		1*		reserved	0				
		01		FPDAT4					
		00		P94					

0x3003C0–0x3003C3

I/O Ports

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Port input interrupt select register 1 (pPINTSEL_SPT03)	003003C0 (B)	D7	SPT31	FPT3 interrupt input port selection	SPT3[1:0]		Port	0	R/W	
		D6	SPT30		11	P33	0			
					10	P13				
					01	P23				
					00	P03				
		D5	SPT21	FPT2 interrupt input port selection	SPT2[1:0]		Port	0	R/W	
		D4	SPT20		11	P32	0			
					10	P12				
					01	P22				
					00	P02				
		D3	SPT11	FPT1 interrupt input port selection	SPT1[1:0]		Port	0	R/W	
		D2	SPT10		11	P31	0			
		10	P11							
		01	P21							
		00	P01							
D1	SPT01	FPT0 interrupt input port selection	SPT0[1:0]		Port	0	R/W			
D0	SPT00		11	P30	0					
			10	P10						
			01	P20						
			00	P00						
Port input interrupt select register 2 (pPINTSEL_SPT47)	003003C1 (B)	D7	SPT71	FPT7 interrupt input port selection	SPT7[1:0]		Port	0	R/W	
		D6	SPT70		11	P63	0			
					10	P17				
					01	P27				
					00	P07				
		D5	SPT61	FPT6 interrupt input port selection	SPT6[1:0]		Port	0	R/W	
		D4	SPT60		11	P62	0			
					10	P16				
					01	P26				
					00	P06				
		D3	SPT51	FPT5 interrupt input port selection	SPT5[1:0]		Port	0	R/W	
		D2	SPT50		11	P61	0			
		10	P15							
		01	P25							
		00	P05							
D1	SPT41	FPT4 interrupt input port selection	SPT4[1:0]		Port	0	R/W			
D0	SPT40		11	P60	0					
			10	P14						
			01	P24						
			00	P04						
Port input interrupt polarity select register 1 (pPINTPOL_SPP07)	003003C2 (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 1 (pPINTEL_SEPT07)	003003C3 (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

0x3003C4–0x3003C7

I/O Ports

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Port input interrupt select register 3 (pPINTSEL_SPT811)	003003C4 (B)	D7	SPTB1	FPT11 interrupt input port selection	SPTB[1:0]	Port	0	R/W		
		D6	SPTB0		11	P93				
					10	INT_DCSIO				
					01	P83				
			00	P73						
	D5	SPTA1	FPT10 interrupt input port selection	SPTA[1:0]	Port	0	R/W			
	D4	SPTA0		11	P92					
				10	INT_USB					
				01	P82					
			00	P72						
	D3	SPT91	FPT9 interrupt input port selection	SPT9[1:0]	Port	0	R/W			
	D2	SPT90		11	P91					
		10		USB_PDREQ						
		01		P81						
		00	P71							
D1	SPT81	FPT8 interrupt input port selection	SPT8[1:0]	Port	0	R/W				
D0	SPT80		11	P90						
			10	INT_SPI						
			01	P80						
		00	P70							
Port input interrupt select register 4 (pPINTSEL_SPT1215)	003003C5 (B)	D7	SPTF1	FPT15 interrupt input port selection	SPTF[1:0]	Port	0	R/W		
		D6	SPTF0		11	P97				
					10	P67				
					01	P53				
			00	P43						
	D5	SPT E1	FPT14 interrupt input port selection	SPT E[1:0]	Port	0	R/W			
	D4	SPT E0		11	P96					
				10	P66					
				01	P52					
			00	P42						
	D3	SPTD1	FPT13 interrupt input port selection	SPTD[1:0]	Port	0	R/W			
	D2	SPTD0		11	P95					
		10		P65						
		01		P51						
		00	P41							
D1	SPTC1	FPT12 interrupt input port selection	SPTC[1:0]	Port	0	R/W				
D0	SPTC0		11	P94						
			10	P64						
			01	P50						
		00	P40							
Port input interrupt polarity select register 2 (pPINTPOL_SPP815)	003003C6 (B)	D7	SPPTF	FPT15 input polarity selection	1	High level or Rising edge	0	Low level or Falling edge	1	R/W
		D6	SPPTE	FPT14 input polarity selection					1	R/W
		D5	SPPTD	FPT13 input polarity selection					1	R/W
		D4	SPPTC	FPT12 input polarity selection					1	R/W
		D3	SPPTB	FPT11 input polarity selection					1	R/W
		D2	SPPTA	FPT10 input polarity selection					1	R/W
		D1	SPPT9	FPT9 input polarity selection					1	R/W
		D0	SPPT8	FPT8 input polarity selection					1	R/W
Port input interrupt edge/level select register 2 (pPINTEL_SEPT815)	003003C7 (B)	D7	SEPTF	FPT15 edge/level selection	1	Edge	0	Level	1	R/W
		D6	SEPTE	FPT14 edge/level selection					1	R/W
		D5	SEPTD	FPT13 edge/level selection					1	R/W
		D4	SEPTC	FPT12 edge/level selection					1	R/W
		D3	SEPTB	FPT11 edge/level selection					1	R/W
		D2	SEPTA	FPT10 edge/level selection					1	R/W
		D1	SEPT9	FPT9 edge/level selection					1	R/W
		D0	SEPT8	FPT8 edge/level selection					1	R/W

0x3003D0–0x3003D5

I/O Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Key input interrupt select register (pKINTSEL_SPPK01)	003003D0 (B)	D7	–	reserved	–	–	–	0 when being read.	
		D6	SPPK12	FPK1 interrupt input port selection	SPPK1[2:0] Port	0	R/W		
		D5	SPPK11			111	P9[7:4]		0
		D4	SPPK10			110	P8[5:4]		0
						101	P7[3:0]		0
						100	P6[7:4]		0
						011	P3[3:0]		0
						010	P2[7:4]		0
						001	P1[7:4]		0
		000	P0[7:4]	0					
D3	–	reserved	–	–	–	0 when being read.			
D2	SPPK02	FPK0 interrupt input port selection	SPPK0[2:0] Port	0	R/W				
D1	SPPK01			111	P9[4:0]		0		
				110	P8[4:0]		0		
				101	P5[4:0]		0		
				100	P6[4:0]		0		
D0	SPPK00	011	P4[4:0]	0					
		010	P2[4:0]	0					
		001	P1[4:0]	0					
000	P0[4:0]	0							
Key input interrupt (FPK0) input comparison register (pKINTCOMP_SCPK0)	003003D2 (B)	D7–5	–	reserved	–	–	–	0 when being read.	
		D4	SCPCK04	FPK04 input comparison	1 High	0 Low	0	R/W	
		D3	SCPCK03	FPK03 input comparison			0	R/W	
		D2	SCPCK02	FPK02 input comparison			0	R/W	
		D1	SCPCK01	FPK01 input comparison			0	R/W	
		D0	SCPCK00	FPK00 input comparison			0	R/W	
0	R/W								
Key input interrupt (FPK1) input comparison register (pKINTCOMP_SCPK1)	003003D3 (B)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3	SCPCK13	FPK13 input comparison	1 High	0 Low	0	R/W	
		D2	SCPCK12	FPK12 input comparison			0	R/W	
		D1	SCPCK11	FPK11 input comparison			0	R/W	
		D0	SCPCK10	FPK10 input comparison			0	R/W	
0	R/W								
Key input interrupt (FPK0) input mask register (pKINTCOMP_SMPK0)	003003D4 (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	
0	R/W								
Key input interrupt (FPK1) input mask register (pKINTCOMP_SMPK1)	003003D5 (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	
0	R/W								

0x300520–0x300544

A/D Converter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
A/D clock control register (pAD_CLKCTL)	00300520 (HW)	D15–4	–	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	
		D1	PSAD1		111 MCLK/256	0			
		D0	PSAD0		110 MCLK/128	0			
					101 MCLK/64				
					100 MCLK/32				
					011 MCLK/16				
		010 MCLK/8							
		001 MCLK/4							
		000 MCLK/2							
A/D conversion result register (pAD_ADD)	00300540 (HW)	D15–10	–	reserved	–	–	–	0 when being read.	
		D9	ADD9	A/D converted data ADD9 = MSB ADD0 = LSB	0x0 to 0x3FF	0	R		
		D8	ADD8		0				
		D7	ADD7		0				
		D6	ADD6		0				
		D5	ADD5		0				
		D4	ADD4		0				
		D3	ADD3		0				
		D2	ADD2		0				
		D1	ADD1		0				
		D0	ADD0		0				
A/D trigger/channel select register (pAD_TRIG_CHNL)	00300542 (HW)	D15–14	–	reserved	–	–	–	0 when being read.	
		D13	CE2	A/D converter end channel selection	0 to 4	0	R/W		
		D12	CE1		0				
		D11	CE0		0				
		D10	CS2	A/D converter start channel selection	0 to 4	0	R/W		
		D9	CS1		0				
		D8	CS0		0				
		D7–6	–	reserved	–	–	–	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		11 #ADTRG pin	0			
		10 reserved							
		01 16-bit timer							
		00 Software							
D2	CH2	A/D conversion channel status	0 to 4	0	R				
D1	CH1		0						
D0	CH0		0						
A/D control/status register (pAD_EN_SMPPL_STAT)	00300544 (HW)	D15	ADCMPE	Upper/lower-limit comparison enable	1 Enabled 0 Disabled	0	R/W	Can be used when ADCADV = "1".	
		D14	ADCMP2	Upper/lower-limit comparison channel selection	0 to 4	0	R/W		
		D13	ADCMP1		0				
		D12	ADCMP0		0				
		D11	ADUPRST	Upper-limit comparison status	1 Out of range 0 Within range	0	R		
		D10	ADLWRST	Lower-limit comparison status	1 Out of range 0 Within range	0	R		
		D9	ST1	Input signal sampling time setup	ST[1:0] Sampling time	1	R/W	Use with 9 clocks.	
		D8	ST0		11 9 clocks	1			
					10 7 clocks				
					01 5 clocks				
				00 3 clocks					
		D7	–	reserved	–	–	–	0 when being read.	
		D6	INTMODE	Interrupt signal mode	1 Complete only 0 OR	0	R/W	Can be used when	
		D5	CMPINTEN	Out-of-range int. enable	1 Enabled 0 Disabled	0	R/W	ADCADV = "1".	
		D4	CNVINTEN	Conversion-complete int. enable	1 Enabled 0 Disabled	1	R/W		
D3	ADF	Conversion-complete flag	1 Completed 0 Run/Standby	0	R	Reset when ADD is read.			
D2	ADE	A/D enable	1 Enabled 0 Disabled	0	R/W				
D1	ADST	A/D conversion control/status	1 Start/Run 0 Stop	0	R/W				
D0	OWE	Overwrite error flag	1 Error 0 Normal	0	R/W	Reset by writing 0.			

0x300546–0x300550

A/D Converter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
A/D channel status flag register (pAD_END)	00300546 (HW)	D15–13	–	reserved	–	–	–	0 when being read.		
		D12	OWE4	Ch.4 overwrite error flag	1 Error	0 Normal	0	R/W	Can be used when ADCADV = "1".	
		D11	OWE3	Ch.3 overwrite error flag			0	R/W		
		D10	OWE2	Ch.2 overwrite error flag			0	R/W	Reset by writing 0.	
		D9	OWE1	Ch.1 overwrite error flag			0	R/W		
		D8	OWE0	Ch.0 overwrite error flag			0	R/W		
		D7–5	–	reserved	–	–	–	–		0 when being read.
			D4	ADF4	Ch.4 conversion-complete flag	1 Completed	0 Run/Standby	0	R	Can be used when ADCADV = "1".
			D3	ADF3	Ch.3 conversion-complete flag			0	R	
			D2	ADF2	Ch.2 conversion-complete flag			0	R	Reset when ADBUFx is read.
			D1	ADF1	Ch.1 conversion-complete flag			0	R	
			D0	ADF0	Ch.0 conversion-complete flag			0	R	
A/D Ch.0 conversion result buffer register (pAD_CH0_BUF)	00300548 (HW)	D15–10	–	reserved	–	–	–	0 when being read.		
		D9	AD0BUF9	A/D Ch.0 converted data	0x0 to 0x3FF	0	R	Can be used when ADCADV = "1".		
		D8	AD0BUF8	AD0BUF9 = MSB						
		D7	AD0BUF7	AD0BUF0 = LSB						
		D6	AD0BUF6							
		D5	AD0BUF5							
		D4	AD0BUF4							
		D3	AD0BUF3							
		D2	AD0BUF2							
		D1	AD0BUF1							
		D0	AD0BUF0							
A/D Ch.1 conversion result buffer register (pAD_CH1_BUF)	0030054A (HW)	D15–10	–	reserved					–	–
		D9	AD1BUF9	A/D Ch.1 converted data	0x0 to 0x3FF	0	R	Can be used when ADCADV = "1".		
		D8	AD1BUF8	AD1BUF9 = MSB						
		D7	AD1BUF7	AD1BUF0 = LSB						
		D6	AD1BUF6							
		D5	AD1BUF5							
		D4	AD1BUF4							
		D3	AD1BUF3							
		D2	AD1BUF2							
		D1	AD1BUF1							
		D0	AD1BUF0							
A/D Ch.2 conversion result buffer register (pAD_CH2_BUF)	0030054C (HW)	D15–10	–	reserved					–	–
		D9	AD2BUF9	A/D Ch.2 converted data	0x0 to 0x3FF	0	R	Can be used when ADCADV = "1".		
		D8	AD2BUF8	AD2BUF9 = MSB						
		D7	AD2BUF7	AD2BUF0 = LSB						
		D6	AD2BUF6							
		D5	AD2BUF5							
		D4	AD2BUF4							
		D3	AD2BUF3							
		D2	AD2BUF2							
		D1	AD2BUF1							
		D0	AD2BUF0							
A/D Ch.3 conversion result buffer register (pAD_CH3_BUF)	0030054E (HW)	D15–10	–	reserved					–	–
		D9	AD3BUF9	A/D Ch.3 converted data	0x0 to 0x3FF	0	R	Can be used when ADCADV = "1".		
		D8	AD3BUF8	AD3BUF9 = MSB						
		D7	AD3BUF7	AD3BUF0 = LSB						
		D6	AD3BUF6							
		D5	AD3BUF5							
		D4	AD3BUF4							
		D3	AD3BUF3							
		D2	AD3BUF2							
		D1	AD3BUF1							
		D0	AD3BUF0							
A/D Ch.4 conversion result buffer register (pAD_CH4_BUF)	00300550 (HW)	D15–10	–	reserved					–	–
		D9	AD4BUF9	A/D Ch.4 converted data	0x0 to 0x3FF	0	R	Can be used when ADCADV = "1".		
		D8	AD4BUF8	AD4BUF9 = MSB						
		D7	AD4BUF7	AD4BUF0 = LSB						
		D6	AD4BUF6							
		D5	AD4BUF5							
		D4	AD4BUF4							
		D3	AD4BUF3							
		D2	AD4BUF2							
		D1	AD4BUF1							
		D0	AD4BUF0							

0x300558–0x30055E

A/D Converter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
A/D upper limit value register (pAD_UPPER)	00300558 (HW)	D15–10	–	reserved	–	–	–	0 when being read.		
		D9	ADUPR9	A/D conversion upper limit value	0x0 to 0x3FF	0	R/W	Can be used when ADCADV = "1".		
		D8	ADUPR8	ADUPR9 = MSB		0				
		D7	ADUPR7	ADUPR0 = LSB		0				
		D6	ADUPR6			0				
		D5	ADUPR5			0				
		D4	ADUPR4			0				
		D3	ADUPR3			0				
		D2	ADUPR2			0				
		D1	ADUPR1			0				
D0	ADUPR0		0							
A/D lower limit value register (pAD_LOWER)	0030055A (HW)	D15–10	–	reserved	–	–	–	0 when being read.		
		D9	ADLWR9	A/D conversion lower limit value	0x0 to 0x3FF	0	R/W	Can be used when ADCADV = "1".		
		D8	ADLWR8	ADLWR9 = MSB		0				
		D7	ADLWR7	ADLWR0 = LSB		0				
		D6	ADLWR6			0				
		D5	ADLWR5			0				
		D4	ADLWR4			0				
		D3	ADLWR3			0				
		D2	ADLWR2			0				
		D1	ADLWR1			0				
D0	ADLWR0		0							
A/D conversion complete interrupt mask register (pAD_CH04_INTMASK)	0030055C (HW)	D15–5	–	reserved	–	–	–	0 when being read.		
		D4	INTMASK4	Ch.4 conversion-complete int. mask	1 Interrupt enabled	0 Interrupt mask	1	R/W	Can be used when ADCADV = "1".	
		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/ internal status register (pAD_ADVMODE)	0030055E (HW)	D15–9	–	reserved	–	–	–	Do not write 1.		
		D8	ADCADV	Standard/advanced mode selection	1 Advanced	0 Standard	0	R/W		
		D7–6	–	reserved	–	–	–	–	0 when being read.	
		D5	ISTATE1	Internal status	ISTATE[1:0]		Status	0	R	
		D4	ISTATE0		11	Converting	0			
					10	reserved				
				01	Sampling					
		00	Idle							
D3	ICOUNTER3	Internal counter value	0 to 15		0	R				
D2	ICOUNTER2		0							
D1	ICOUNTER1		0							
D0	ICOUNTER0		0							

0x300660–0x300666

Watchdog Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Watchdog timer write-protect register (pWD_WP)	00300660 (HW)	D15	WDPTC15	Watchdog timer register write-protect	Writing 0x96 removes the write protection of the watchdog timer enable and comparison data registers (0x300662–0x300666). Writing another value set the write protection.	X	W	0 when being read.	
		D14	WDPTC14						
		D13	WDPTC13						
		D12	WDPTC12						
		D11	WDPTC11						
		D10	WDPTC10						
		D9	WDPTC9						
		D8	WDPTC8						
		D7	WDPTC7						
		D6	WDPTC6						
		D5	WDPTC5						
		D4	WDPTC4						
		D3	WDPTC3						
		D2	WDPTC2						
		D1	WDPTC1						
		D0	WDPTC0						
Watchdog timer enable register (pWD_EN)	00300662 (HW)	D15–7	–	reserved	–	–	–	0 when being read.	
		D6	CLKSEL	Watchdog timer input clock select	1 External clock	0 Internal clock	0		R/W
		D5	CLKEN	Watchdog timer clock output control	1 On	0 Off	0	R/W	
		D4	RUNSTP	Watchdog timer Run/Stop control	1 Run	0 Stop	0	R/W	
		D3–2	–	reserved	–	–	–	–	0 when being read.
		D0	RESEN	Watchdog timer RESET enable	1 Enabled	0 Disabled	0	R/W	
Watchdog timer comparison data setup register 0 (pWD_COMP_LOW)	00300664 (HW)	D15	CMPDT15	Watchdog timer comparison data CMPDT0 = LSB	0x0 to 0x3FFFFFFF (low-order 16 bits)	0	R/W		
		D14	CMPDT14						
		D13	CMPDT13						
		D12	CMPDT12						
		D11	CMPDT11						
		D10	CMPDT10						
		D9	CMPDT9						
		D8	CMPDT8						
		D7	CMPDT7						
		D6	CMPDT6						
		D5	CMPDT5						
		D4	CMPDT4						
		D3	CMPDT3						
		D2	CMPDT2						
D1	CMPDT1								
D0	CMPDT0								
Watchdog timer comparison data setup register 1 (pWD_COMP_HIGH)	00300666 (HW)	D15–14	–	reserved	–	–	–	0 when being read.	
		D13	CMPDT29	Watchdog timer comparison data CMPDT29 = MSB	0x0 to 0x3FFFFFFF (high-order 14 bits)	0	R/W		
		D12	CMPDT28						
		D11	CMPDT27						
		D10	CMPDT26						
		D9	CMPDT25						
		D8	CMPDT24						
		D7	CMPDT23						
		D6	CMPDT22						
		D5	CMPDT21						
		D4	CMPDT20						
		D3	CMPDT19						
		D2	CMPDT18						
		D1	CMPDT17						
		D0	CMPDT16						

0x300668–0x30066C **Watchdog Timer**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog timer count register 0 (pWD_CNT_LOW)	00300668 (HW)	D15	CTRDT15	Watchdog timer counter data CTRDT0 = LSB	0x0 to 0x3FFFFFFF (low-order 16 bits)	X	R	
		D14	CTRDT14					
		D13	CTRDT13					
		D12	CTRDT12					
		D11	CTRDT11					
		D10	CTRDT10					
		D9	CTRDT9					
		D8	CTRDT8					
		D7	CTRDT7					
		D6	CTRDT6					
		D5	CTRDT5					
		D4	CTRDT4					
		D3	CTRDT3					
		D2	CTRDT2					
		D1	CTRDT1					
		D0	CTRDT0					
		Watchdog timer count register 1 (pWD_CNT_HIGH)	0030066A (HW)			D15-14		
D13	CTRDT29			Watchdog timer counter data CTRDT29 = MSB	0x0 to 0x3FFFFFFF (high-order 14 bits)	X	R	
D12	CTRDT28							
D11	CTRDT27							
D10	CTRDT26							
D9	CTRDT25							
D8	CTRDT24							
D7	CTRDT23							
D6	CTRDT22							
D5	CTRDT21							
D4	CTRDT20							
D3	CTRDT19							
D2	CTRDT18							
D1	CTRDT17							
D0	CTRDT16							
Watchdog timer control register (pWD_CNTL)	0030066C (HW)	D15-1	–	reserved	–	–	–	0 when being read.
		D0	WDRESEN	Watchdog timer reset	1 Reset	0 Invalid	0	W

0x300780–0x300786

16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 0 comparison data A setup register (pT16_CR0A)	00300780 (HW)	D15	CR0A15	16-bit timer 0 comparison data A CR0A15 = MSB CR0A0 = LSB	0 to 65535	X	R/W		
		D14	CR0A14						
		D13	CR0A13						
		D12	CR0A12						
		D11	CR0A11						
		D10	CR0A10						
		D9	CR0A9						
		D8	CR0A8						
		D7	CR0A7						
		D6	CR0A6						
		D5	CR0A5						
		D4	CR0A4						
		D3	CR0A3						
		D2	CR0A2						
		D1	CR0A1						
		D0	CR0A0						
16-bit timer 0 comparison data B setup register (pT16_CR0B)	00300782 (HW)	D15	CR0B15	16-bit timer 0 comparison data B CR0B15 = MSB CR0B0 = LSB	0 to 65535	X	R/W		
		D14	CR0B14						
		D13	CR0B13						
		D12	CR0B12						
		D11	CR0B11						
		D10	CR0B10						
		D9	CR0B9						
		D8	CR0B8						
		D7	CR0B7						
		D6	CR0B6						
		D5	CR0B5						
		D4	CR0B4						
		D3	CR0B3						
		D2	CR0B2						
		D1	CR0B1						
		D0	CR0B0						
16-bit timer 0 counter data register (pT16_TC0)	00300784 (HW)	D15	TC015	16-bit timer 0 counter data TC015 = MSB TC00 = LSB	0 to 65535	X	R/W	Data can be written only in advanced mode.	
		D14	TC014						
		D13	TC013						
		D12	TC012						
		D11	TC011						
		D10	TC010						
		D9	TC09						
		D8	TC08						
		D7	TC07						
		D6	TC06						
		D5	TC05						
		D4	TC04						
		D3	TC03						
		D2	TC02						
		D1	TC01						
		D0	TC00						
16-bit timer 0 control register (pT16_CTL0)	00300786 (HW)	D15–9	--	reserved	--	--	--	0 when being read.	
		D8	INITOLO	16-bit timer 0 initial output level	1 High	0 Low	0	R/W	Advanced mode
		D7	(TMODE0)	(reserved for 16-bit timer 0 test)	1 Test mode	0 Normal	0	R	Do not write 1.
		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	PTMO	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	

0x300788–0x30078E

16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 1 comparison data A setup register (pT16_CR1A)	00300788 (HW)	D15	CR1A15	16-bit timer 1 comparison data A CR1A15 = MSB CR1A0 = LSB	0 to 65535	X	R/W		
		D14	CR1A14						
		D13	CR1A13						
		D12	CR1A12						
		D11	CR1A11						
		D10	CR1A10						
		D9	CR1A9						
		D8	CR1A8						
		D7	CR1A7						
		D6	CR1A6						
		D5	CR1A5						
		D4	CR1A4						
		D3	CR1A3						
		D2	CR1A2						
		D1	CR1A1						
		D0	CR1A0						
16-bit timer 1 comparison data B setup register (pT16_CR1B)	0030078A (HW)	D15	CR1B15	16-bit timer 1 comparison data B CR1B15 = MSB CR1B0 = LSB	0 to 65535	X	R/W		
		D14	CR1B14						
		D13	CR1B13						
		D12	CR1B12						
		D11	CR1B11						
		D10	CR1B10						
		D9	CR1B9						
		D8	CR1B8						
		D7	CR1B7						
		D6	CR1B6						
		D5	CR1B5						
		D4	CR1B4						
		D3	CR1B3						
		D2	CR1B2						
		D1	CR1B1						
		D0	CR1B0						
16-bit timer 1 counter data register (pT16_TC1)	0030078C (HW)	D15	TC115	16-bit timer 1 counter data TC115 = MSB TC10 = LSB	0 to 65535	X	R/W	Data can be written only in advanced mode.	
		D14	TC114						
		D13	TC113						
		D12	TC112						
		D11	TC111						
		D10	TC110						
		D9	TC19						
		D8	TC18						
		D7	TC17						
		D6	TC16						
		D5	TC15						
		D4	TC14						
		D3	TC13						
		D2	TC12						
		D1	TC11						
		D0	TC10						
16-bit timer 1 control register (pT16_CTL1)	0030078E (HW)	D15–9	–	reserved	–	–	–	0 when being read.	
		D8	INITOL1	16-bit timer 1 initial output level	1 High	0 Low	0	R/W	Advanced mode
		D7	(TMODE1)	(reserved for 16-bit timer 1 test)	1 Test mode	0 Normal	0	R	Do not write 1.
		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	

0x300790–0x300796

16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 2 comparison data A setup register (pT16_CR2A)	00300790 (HW)	D15	CR2A15	16-bit timer 2 comparison data A CR2A15 = MSB CR2A0 = LSB	0 to 65535	X	R/W		
		D14	CR2A14						
		D13	CR2A13						
		D12	CR2A12						
		D11	CR2A11						
		D10	CR2A10						
		D9	CR2A9						
		D8	CR2A8						
		D7	CR2A7						
		D6	CR2A6						
		D5	CR2A5						
		D4	CR2A4						
		D3	CR2A3						
		D2	CR2A2						
		D1	CR2A1						
		D0	CR2A0						
		16-bit timer 2 comparison data B setup register (pT16_CR2B)	00300792 (HW)			D15	CR2B15		16-bit timer 2 comparison data B CR2B15 = MSB CR2B0 = LSB
D14	CR2B14								
D13	CR2B13								
D12	CR2B12								
D11	CR2B11								
D10	CR2B10								
D9	CR2B9								
D8	CR2B8								
D7	CR2B7								
D6	CR2B6								
D5	CR2B5								
D4	CR2B4								
D3	CR2B3								
D2	CR2B2								
D1	CR2B1								
D0	CR2B0								
16-bit timer 2 counter data register (pT16_TC2)	00300794 (HW)			D15	TC215	16-bit timer 2 counter data TC215 = MSB TC20 = LSB	0 to 65535	X	
		D14	TC214						
		D13	TC213						
		D12	TC212						
		D11	TC211						
		D10	TC210						
		D9	TC29						
		D8	TC28						
		D7	TC27						
		D6	TC26						
		D5	TC25						
		D4	TC24						
		D3	TC23						
		D2	TC22						
		D1	TC21						
		D0	TC20						
		16-bit timer 2 control register (pT16_CTL2)	00300796 (HW)	D15–9	--			reserved	--
D8	INITOL2			16-bit timer 2 initial output level	1 High	0 Low	0	R/W	Advanced mode
D7	(TMODE2)			(reserved for 16-bit timer 2 test)	1 Test mode	0 Normal	0	R	Do not write 1.
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	

0x300798–0x30079E **16-bit Timer**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 3 comparison data A setup register (pT16_CR3A)	00300798 (HW)	D15	CR3A15	16-bit timer 3 comparison data A CR3A15 = MSB CR3A0 = LSB	0 to 65535	X	R/W		
		D14	CR3A14						
		D13	CR3A13						
		D12	CR3A12						
		D11	CR3A11						
		D10	CR3A10						
		D9	CR3A9						
		D8	CR3A8						
		D7	CR3A7						
		D6	CR3A6						
		D5	CR3A5						
		D4	CR3A4						
		D3	CR3A3						
		D2	CR3A2						
		D1	CR3A1						
		D0	CR3A0						
		16-bit timer 3 comparison data B setup register (pT16_CR3B)	0030079A (HW)			D15			CR3B15
D14	CR3B14								
D13	CR3B13								
D12	CR3B12								
D11	CR3B11								
D10	CR3B10								
D9	CR3B9								
D8	CR3B8								
D7	CR3B7								
D6	CR3B6								
D5	CR3B5								
D4	CR3B4								
D3	CR3B3								
D2	CR3B2								
D1	CR3B1								
D0	CR3B0								
16-bit timer 3 counter data register (pT16_TC3)	0030079C (HW)			D15	TC315	16-bit timer 3 counter data TC315 = MSB TC30 = LSB	0 to 65535	X	R/W
		D14	TC314						
		D13	TC313						
		D12	TC312						
		D11	TC311						
		D10	TC310						
		D9	TC39						
		D8	TC38						
		D7	TC37						
		D6	TC36						
		D5	TC35						
		D4	TC34						
		D3	TC33						
		D2	TC32						
		D1	TC31						
		D0	TC30						
		16-bit timer 3 control register (pT16_CTL3)	0030079E (HW)	D15–9	–			reserved	
D8	INITOL3			16-bit timer 3 initial output level	1 High	0 Low	0	R/W	Advanced mode
D7	(TMODE3)			(reserved for 16-bit timer 3 test)	1 Test mode	0 Normal	0	R	Do not write 1.
D6	SELF3			16-bit timer 3 fine mode selection	1 Fine mode	0 Normal	0	R/W	
D5	SELCRB3			16-bit timer 3 comparison buffer	1 Enabled	0 Disabled	0	R/W	
D4	OUTINV3			16-bit timer 3 output inversion	1 Invert	0 Normal	0	R/W	
D3	CKSL3			16-bit timer 3 input clock selection	1 External clock	0 Internal clock	0	R/W	
D2	PTM3			16-bit timer 3 clock output control	1 On	0 Off	0	R/W	
D1	PRESET3			16-bit timer 3 reset	1 Reset	0 Invalid	0	W	0 when being read.
D0	PRUN3			16-bit timer 3 Run/Stop control	1 Run	0 Stop	0	R/W	

0x3007A0–0x3007A6

16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
16-bit timer 4 comparison data A setup register (pT16_CR4A)	003007A0 (HW)	D15	CR4A15	16-bit timer 4 comparison data A CR4A15 = MSB CR4A0 = LSB	0 to 65535	X	R/W				
		D14	CR4A14								
		D13	CR4A13								
		D12	CR4A12								
		D11	CR4A11								
		D10	CR4A10								
		D9	CR4A9								
		D8	CR4A8								
		D7	CR4A7								
		D6	CR4A6								
		D5	CR4A5								
		D4	CR4A4								
		D3	CR4A3								
		D2	CR4A2								
		D1	CR4A1								
		D0	CR4A0								
16-bit timer 4 comparison data B setup register (pT16_CR4B)	003007A2 (HW)	D15	CR4B15	16-bit timer 4 comparison data B CR4B15 = MSB CR4B0 = LSB	0 to 65535	X	R/W				
		D14	CR4B14								
		D13	CR4B13								
		D12	CR4B12								
		D11	CR4B11								
		D10	CR4B10								
		D9	CR4B9								
		D8	CR4B8								
		D7	CR4B7								
		D6	CR4B6								
		D5	CR4B5								
		D4	CR4B4								
		D3	CR4B3								
		D2	CR4B2								
		D1	CR4B1								
		D0	CR4B0								
16-bit timer 4 counter data register (pT16_TC4)	003007A4 (HW)	D15	TC415	16-bit timer 4 counter data TC415 = MSB TC40 = LSB	0 to 65535	X	R/W	Data can be written only in advanced mode.			
		D14	TC414								
		D13	TC413								
		D12	TC412								
		D11	TC411								
		D10	TC410								
		D9	TC49								
		D8	TC48								
		D7	TC47								
		D6	TC46								
		D5	TC45								
		D4	TC44								
		D3	TC43								
		D2	TC42								
		D1	TC41								
		D0	TC40								
16-bit timer 4 control register (pT16_CTL4)	003007A6 (HW)	D15–9	--	reserved	--	--	--	0 when being read.			
		D8	INITOL4	16-bit timer 4 initial output level	1	High	0	Low	0	R/W	Advanced mode
		D7	(TMODE4)	(reserved for 16-bit timer 4 test)	1	Test mode	0	Normal	0	R	Do not write 1.
		D6	SELF4	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	

0x3007A8–0x3007AE **16-bit Timer**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 5 comparison data A setup register (pT16_CR5A)	003007A8 (HW)	D15	CR5A15	16-bit timer 5 comparison data A CR5A15 = MSB CR5A0 = LSB	0 to 65535	X	R/W		
		D14	CR5A14						
		D13	CR5A13						
		D12	CR5A12						
		D11	CR5A11						
		D10	CR5A10						
		D9	CR5A9						
		D8	CR5A8						
		D7	CR5A7						
		D6	CR5A6						
		D5	CR5A5						
		D4	CR5A4						
		D3	CR5A3						
		D2	CR5A2						
		D1	CR5A1						
		D0	CR5A0						
16-bit timer 5 comparison data B setup register (pT16_CR5B)	003007AA (HW)	D15	CR5B15	16-bit timer 5 comparison data B CR5B15 = MSB CR5B0 = LSB	0 to 65535	X	R/W		
		D14	CR5B14						
		D13	CR5B13						
		D12	CR5B12						
		D11	CR5B11						
		D10	CR5B10						
		D9	CR5B9						
		D8	CR5B8						
		D7	CR5B7						
		D6	CR5B6						
		D5	CR5B5						
		D4	CR5B4						
		D3	CR5B3						
		D2	CR5B2						
		D1	CR5B1						
		D0	CR5B0						
16-bit timer 5 counter data register (pT16_TC5)	003007AC (HW)	D15	TC515	16-bit timer 5 counter data TC515 = MSB TC50 = LSB	0 to 65535	X	R/W	Data can be written only in advanced mode.	
		D14	TC514						
		D13	TC513						
		D12	TC512						
		D11	TC511						
		D10	TC510						
		D9	TC59						
		D8	TC58						
		D7	TC57						
		D6	TC56						
		D5	TC55						
		D4	TC54						
		D3	TC53						
		D2	TC52						
		D1	TC51						
		D0	TC50						
16-bit timer 5 control register (pT16_CTL5)	003007AE (HW)	D15–9	–	reserved	–	–	–	0 when being read.	
		D8	INITOL5	16-bit timer 5 initial output level	1 High	0 Low	0	R/W	Advanced mode
		D7	(TMODE5)	(reserved for 16-bit timer 5 test)	1 Test mode	0 Normal	0	R	Do not write 1.
		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	

0x3007D0–0x3007DE

16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
DA16 Ch.0 register (pDA16_CR0A)	003007D0 (HW)	D15	DA0A15	DA16 Ch.0 comparison data A DA0A15 = MSB DA0A0 = LSB	0 to 65535	X	R/W	Advanced mode			
		D14	DA0A14								
		D13	DA0A13								
		D12	DA0A12								
		D11	DA0A11								
		D10	DA0A10								
		D9	DA0A9								
		D8	DA0A8								
		D7	DA0A7								
		D6	DA0A6								
		D5	DA0A5								
		D4	DA0A4								
		D3	DA0A3								
		D2	DA0A2								
		D1	DA0A1								
		D0	DA0A0								
DA16 Ch.1 register (pDA16_CR1A)	003007D2 (HW)	D15	DA1A15	DA16 Ch.1 comparison data A DA1A15 = MSB DA1A0 = LSB	0 to 65535	X	R/W	Advanced mode			
		D14	DA1A14								
		D13	DA1A13								
		D12	DA1A12								
		D11	DA1A11								
		D10	DA1A10								
		D9	DA1A9								
		D8	DA1A8								
		D7	DA1A7								
		D6	DA1A6								
		D5	DA1A5								
		D4	DA1A4								
		D3	DA1A3								
		D2	DA1A2								
		D1	DA1A1								
		D0	DA1A0								
DA16 Ch.2 register (pDA16_CR2A)	003007D4 (HW)	D15	DA2A15	DA16 Ch.2 comparison data A DA2A15 = MSB DA2A0 = LSB	0 to 65535	X	R/W	Advanced mode			
		D14	DA2A14								
		D13	DA2A13								
		D12	DA2A12								
		D11	DA2A11								
		D10	DA2A10								
		D9	DA2A9								
		D8	DA2A8								
		D7	DA2A7								
		D6	DA2A6								
		D5	DA2A5								
		D4	DA2A4								
		D3	DA2A3								
		D2	DA2A2								
		D1	DA2A1								
		D0	DA2A0								
Count pause register (pT16_CNT_PAUSE)	003007DC (HW)	D15–6	–	reserved	–	–	–	0 when being read.			
		D5	PAUSE5	16-bit timer 5 count pause	1	Pause	0	Count	0	R/W	Advanced mode
		D4	PAUSE4	16-bit timer 4 count pause					0	R/W	
		D3	PAUSE3	16-bit timer 3 count pause					0	R/W	
		D2	PAUSE2	16-bit timer 2 count pause					0	R/W	
		D1	PAUSE1	16-bit timer 1 count pause					0	R/W	
		D0	PAUSE0	16-bit timer 0 count pause					0	R/W	
16-bit timer STD/ADV mode select register (pT16_ADVMODE)	003007DE (HW)	D15–1	–	reserved					–	–	
		D0	T16ADV	Standard mode/advanced mode select	1	Advanced mode	0	Standard mode	0	R/W	

0x3007E0–0x3007EA

16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
16-bit timer 0 clock control register (pT16_CLKCTL_0)	003007E0 (HW)	D15–4	–	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 select	P16TS0[2:0]	Division ratio	0	R/W	
		D1	P16TS01		111	MCLK/4096	0		
		D0	P16TS00		110	MCLK/1024	0		
					101	MCLK/256			
					100	MCLK/64			
					011	MCLK/16			
		010	MCLK/4						
		001	MCLK/2						
		000	MCLK/1						
16-bit timer 1 clock control register (pT16_CLKCTL_1)	003007E2 (HW)	D15–4	–	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 select	P16TS1[2:0]	Division ratio	0	R/W	
		D1	P16TS11		111	MCLK/4096	0		
		D0	P16TS10		110	MCLK/1024	0		
					101	MCLK/256			
					100	MCLK/64			
					011	MCLK/16			
		010	MCLK/4						
		001	MCLK/2						
		000	MCLK/1						
16-bit timer 2 clock control register (pT16_CLKCTL_2)	003007E4 (HW)	D15–4	–	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 select	P16TS2[2:0]	Division ratio	0	R/W	
		D1	P16TS21		111	MCLK/4096	0		
		D0	P16TS20		110	MCLK/1024	0		
					101	MCLK/256			
					100	MCLK/64			
					011	MCLK/16			
		010	MCLK/4						
		001	MCLK/2						
		000	MCLK/1						
16-bit timer 3 clock control register (pT16_CLKCTL_3)	003007E6 (HW)	D15–4	–	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 select	P16TS3[2:0]	Division ratio	0	R/W	
		D1	P16TS31		111	MCLK/4096	0		
		D0	P16TS30		110	MCLK/1024	0		
					101	MCLK/256			
					100	MCLK/64			
					011	MCLK/16			
		010	MCLK/4						
		001	MCLK/2						
		000	MCLK/1						
16-bit timer 4 clock control register (pT16_CLKCTL_4)	003007E8 (HW)	D15–4	–	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 select	P16TS4[2:0]	Division ratio	0	R/W	
		D1	P16TS41		111	MCLK/4096	0		
		D0	P16TS40		110	MCLK/1024	0		
					101	MCLK/256			
					100	MCLK/64			
					011	MCLK/16			
		010	MCLK/4						
		001	MCLK/2						
		000	MCLK/1						
16-bit timer 5 clock control register (pT16_CLKCTL_5)	003007EA (HW)	D15–4	–	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 select	P16TS5[2:0]	Division ratio	0	R/W	
		D1	P16TS51		111	MCLK/4096	0		
		D0	P16TS50		110	MCLK/1024	0		
					101	MCLK/256			
					100	MCLK/64			
					011	MCLK/16			
		010	MCLK/4						
		001	MCLK/2						
		000	MCLK/1						

0x300900–0x30090B

USB Function Controller

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks			
MainIntStat (Main interrupt status)	00300900 (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	RcvEPOSETUP	1	Receive EP0 SETUP	0	None	0	R(W)		
SIE_IntStat (SIE interrupt status)	00300901 (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)		
EPrintStat (EPr interrupt status)	00300902 (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		
DMA_IntStat (DMA interrupt status)	00300903 (B)	D7–2	–			–		–	–	0 when being read.	
		D1	DMA_CountUp	1	DMA counter overflow	0	None	0	R(W)		
		D0	DMA_Cmp	1	DMA complete	0	None	0	R(W)		
FIFO_IntStat (FIFO interrupt status)	00300904 (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)		
EP0IntStat (EP0 interrupt status)	00300907 (B)	D7–6	–			–		–	–	0 when being read.	
		D5	IN_TransACK	1	In transaction ACK	0	None	0	R(W)		
		D4	OUT_TransACK	1	Out transaction ACK	0	None	0	R(W)		
		D3	IN_TransNAK	1	In transaction NAK	0	None	0	R(W)		
		D2	OUT_TransNAK	1	Out transaction NAK	0	None	0	R(W)		
		D1	IN_TransErr	1	In transaction error	0	None	0	R(W)		
		D0	OUT_TransErr	1	Out transaction error	0	None	0	R(W)		
EPaIntStat (EPa interrupt status)	00300908 (B)	D7	–			–		–	–		0 when being read.
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)		
		D5	IN_TransACK	1	In transaction ACK	0	None	0	R(W)		
		D4	OUT_TransACK	1	Out transaction ACK	0	None	0	R(W)		
		D3	IN_TransNAK	1	In transaction NAK	0	None	0	R(W)		
		D2	OUT_TransNAK	1	Out transaction NAK	0	None	0	R(W)		
		D1	IN_TransErr	1	In transaction error	0	None	0	R(W)		
		D0	OUT_TransErr	1	Out transaction error	0	None	0	R(W)		
EPbIntStat (EPb interrupt status)	00300909 (B)	D7	–			–		–	–		0 when being read.
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)		
		D5	IN_TransACK	1	In transaction ACK	0	None	0	R(W)		
		D4	OUT_TransACK	1	Out transaction ACK	0	None	0	R(W)		
		D3	IN_TransNAK	1	In transaction NAK	0	None	0	R(W)		
		D2	OUT_TransNAK	1	Out transaction NAK	0	None	0	R(W)		
		D1	IN_TransErr	1	In transaction error	0	None	0	R(W)		
		D0	OUT_TransErr	1	Out transaction error	0	None	0	R(W)		
				D0	OUT_TransErr	1	Out transaction error	0	None		0
EPcIntStat (EPc interrupt status)	0030090A (B)	D7	–			–		–	–		0 when being read.
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)		
		D5	IN_TransACK	1	In transaction ACK	0	None	0	R(W)		
		D4	OUT_TransACK	1	Out transaction ACK	0	None	0	R(W)		
		D3	IN_TransNAK	1	In transaction NAK	0	None	0	R(W)		
		D2	OUT_TransNAK	1	Out transaction NAK	0	None	0	R(W)		
		D1	IN_TransErr	1	In transaction error	0	None	0	R(W)		
		D0	OUT_TransErr	1	Out transaction error	0	None	0	R(W)		
				D0	OUT_TransErr	1	Out transaction error	0	None		0
EPdIntStat (EPd interrupt status)	0030090B (B)	D7	–			–		–	–		0 when being read.
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)		
		D5	IN_TransACK	1	In transaction ACK	0	None	0	R(W)		
		D4	OUT_TransACK	1	Out transaction ACK	0	None	0	R(W)		
		D3	IN_TransNAK	1	In transaction NAK	0	None	0	R(W)		
		D2	OUT_TransNAK	1	Out transaction NAK	0	None	0	R(W)		
		D1	IN_TransErr	1	In transaction error	0	None	0	R(W)		
D0	OUT_TransErr	1	Out transaction error	0	None	0	R(W)				

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks									
MainIntEnb (Main interrupt enable)	00300910 (B)	D7	EnSIE_IntStat	1	Enabled	0	Disabled	0	R/W								
		D6	EnEPIntStat														
		D5	EnDMA_IntStat														
		D4	EnFIFO_IntStat														
		D3–2	–								–	–	–	0 when being read.			
		D1	EnEP0IntStat								1	Enabled	0	Disabled	0	R/W	
D0	EnRcvEP0SETUP																
SIE_IntEnb (SIE interrupt enable)	00300911 (B)	D7	EnVBUS_Changed	1	Enabled	0	Disabled	0	R/W								
		D6	EnNonJ														
		D5	EnDetectReset														
		D4	EnDetectSuspend														
		D3	EnRcvSOF														
		D2	EnDetectJ														
		D1	–								–	–	–	0 when being read.			
D0	EnSetAddressCmp	1	Enabled	0	Disabled	0	R/W										
EPIntEnb (EPi interrupt enable)	00300912 (B)	D7–4	–	–	–	–	–	–	–	0 when being read.							
		D3	EnEPdIntStat	1	Enabled	0	Disabled	0	R/W								
		D2	EnEPcIntStat														
		D1	EnEPbIntStat														
		D0	EnEPaIntStat														
D0	EnEPaIntStat																
DMA_IntEnb (DMA interrupt enable)	00300913 (B)	D7–2	–	–	–	–	–	–	–	0 when being read.							
		D1	EnDMA_CountUp	1	Enabled	0	Disabled	0	R/W								
		D0	EnDMA_Cmp														
FIFO_IntEnb (FIFO interrupt enable)	00300914 (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														
EP0IntEnb (EP0 interrupt enable)	00300917 (B)	D7–6	–	–	–	–	–	–	–	0 when being read.							
		D5	EnIN_TranACK	1	Enabled	0	Disabled	0	R/W								
		D4	EnOUT_TranACK														
		D3	EnIN_TranNAK														
		D2	EnOUT_TranNAK														
		D1	EnIN_TranErr														
		D0	EnOUT_TranErr														
D0	EnOUT_TranErr																
EPaIntEnb (EPa interrupt enable)	00300918 (B)	D7	–	–	–	–	–	–	–	0 when being read.							
		D6	EnOUT_ShortACK	1	Enabled	0	Disabled	0	R/W								
		D5	EnIN_TranACK														
		D4	EnOUT_TranACK														
		D3	EnIN_TranNAK														
		D2	EnOUT_TranNAK														
		D1	EnIN_TranErr														
		D0	EnOUT_TranErr														
D0	EnOUT_TranErr																
EPbIntEnb (EPb interrupt enable)	00300919 (B)	D7	–	–	–	–	–	–	–	0 when being read.							
		D6	EnOUT_ShortACK	1	Enabled	0	Disabled	0	R/W								
		D5	EnIN_TranACK														
		D4	EnOUT_TranACK														
		D3	EnIN_TranNAK														
		D2	EnOUT_TranNAK														
		D1	EnIN_TranErr														
		D0	EnOUT_TranErr														
		D0	EnOUT_TranErr														
D0	EnOUT_TranErr																
EPcIntEnb (EPc interrupt enable)	0030091A (B)	D7	–	–	–	–	–	–	–	0 when being read.							
		D6	EnOUT_ShortACK	1	Enabled	0	Disabled	0	R/W								
		D5	EnIN_TranACK														
		D4	EnOUT_TranACK														
		D3	EnIN_TranNAK														
		D2	EnOUT_TranNAK														
		D1	EnIN_TranErr														
		D0	EnOUT_TranErr														
D0	EnOUT_TranErr																
EPdIntEnb (EPd interrupt enable)	0030091B (B)	D7	–	–	–	–	–	–	–	0 when being read.							
		D6	EnOUT_ShortACK	1	Enabled	0	Disabled	0	R/W								
		D5	EnIN_TranACK														
		D4	EnOUT_TranACK														
		D3	EnIN_TranNAK														
		D2	EnOUT_TranNAK														
		D1	EnIN_TranErr														
		D0	EnOUT_TranErr														
D0	EnOUT_TranErr																

0x300920–0x30092F

USB Function Controller

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks			
RevisionNum (Revision number)	00300920 (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)	00300921 (B)			D7	DisBusDetect		1	Disable bus detect	0
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	Deactivate USB	0	R/W		
USB_Status (USB status register)	00300922 (B)	D7	VBUS	1	VBUS=High	0	VBUS=Low	X	R		
		D6	FS	1	FS mode (fixed)	0	–	1	R		
		D5–2	–	–	–	–	–	–	–		0 when being read.
		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						
XcvrControl (Xcvr control register)	00300923 (B)	D7	RpuEnb	1	Enable pull-up	0	Disable pull-up	0	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						
USB_Test (USB test)	00300924 (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)	00300925 (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)	00300926 (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)	0030092E (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)	0030092F (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			

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EP0Setup_0 (EP0 set-up 0)	00300930 (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		
EP0Setup_1 (EP0 set-up 1)	00300931 (B)	D7	EP0Setup_1[7]	Endpoint 0 set-up data 1	0	R	
		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 (EP0 set-up 2)	00300932 (B)	D7	EP0Setup_2[7]	Endpoint 0 set-up data 2	0	R	
		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 (EP0 set-up 3)	00300933 (B)	D7	EP0Setup_3[7]	Endpoint 0 set-up data 3	0	R	
		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 (EP0 set-up 4)	00300934 (B)	D7	EP0Setup_4[7]	Endpoint 0 set-up data 4	0	R	
		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 (EP0 set-up 5)	00300935 (B)	D7	EP0Setup_5[7]	Endpoint 0 set-up data 5	0	R	
		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 (EP0 set-up 6)	00300936 (B)	D7	EP0Setup_6[7]	Endpoint 0 set-up data 6	0	R	
		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 (EP0 set-up 7)	00300937 (B)	D7	EP0Setup_7[7]	Endpoint 0 set-up data 7	0	R	
		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		

0x300938–0x300943

USB Function Controller

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks					
USB_Address (USB address)	00300938 (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				
		EP0Control (EP0 control)	00300939 (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				
EP0ControlIN (EP0 control IN)	0030093A (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				
EP0ControlOUT (EP0 control OUT)	0030093B (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				
EP0MaxSize (EP0 max packet size)	0030093F (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.		
EPaControl (EPa control)	00300940 (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				
		EPbControl (EPb control)	00300941 (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				
EPcControl (EPc control)	00300942 (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				
		EPdControl (EPd control)	00300943 (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				

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks
EPaMaxSize_H (EPa max packet size high)	00300950 (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)	00300951 (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		
		EPaConfig_0 (EPa configuration 0)	00300952 (B)	D7	INxOUT	1 In	0 Out	0
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		
EPaConfig_1 (EPa configuration 1)	00300953 (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.
		D4	–	–		–	–	
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)		0	R/W	
EPbMaxSize_H (EPb max packet size high)	00300954 (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)	00300955 (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)	00300956 (B)	D7	INxOUT	1 In	0 Out	0
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		
EPbConfig_1 (EPb configuration 1)	00300957 (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.
		D4	–	–		–	–	
		D3	EndPointNumber[3]	Endpoint number (0x1 to 0xF)		0	R/W	
EPcMaxSize_H (EPc max packet size high)	00300958 (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)	00300959 (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)	0030095A (B)	D7	INxOUT	1 In	0 Out	0
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				

0x30095B–0x300976

USB Function Controller

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EPcConfig_1 (EPc configuration 1)	0030095B (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)	0030095C (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)	0030095D (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				
EPdConfig_0 (EPd configuration 0)	0030095E (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		0	R/W			
		D2	EndPointNumber[2]	(0x1 to 0xF)		0				
EPdConfig_1 (EPd configuration 1)	0030095F (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.
		D4	–	–	–	–	–	–	–	0 when being read.
		D3	EndPointNumber[3]	Endpoint number		0	R/W			
		D2	EndPointNumber[2]	(0x1 to 0xF)		0				
EPaStartAdrs_H (EPa FIFO start address high)	00300970 (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)	00300971 (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				
EPbStartAdrs_H (EPb FIFO start address high)	00300972 (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)	00300973 (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				
EPcStartAdrs_H (EPc FIFO start address high)	00300974 (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)	00300975 (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				
EPdStartAdrs_H (EPd FIFO start address high)	00300976 (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				

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPdStartAdrs_L (EPd FIFO start address low)	00300977 (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.	
CPU_JoinRd (CPU join FIFO read)	00300980 (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	
CPU_JoinWr (CPU join FIFO write)	00300981 (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)	00300982 (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)	00300983 (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)	00300984 (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)	00300985 (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)	00300986 (B)	D7-4	–	–	–	
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)	00300987 (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)	00300988 (B)	D7-4	–	–	–	
D3	DescAdrs[11]			Descriptor address		0	R/W	
D2	DescAdrs[10]					0		
D1	DescAdrs[9]					0		
D0	DescAdrs[8]					0		
DescAdrs_L (Descriptor address low)	00300989 (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		

0x30098A–0x300999

USB Function Controller

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
DescSize_H (Descriptor size high)	0030098A (B)	D7–2	–	–	–	–	0 when being read.	
		D1	DescSize[9]	Descriptor size	0	R/W		
		D0	DescSize[8]					
DescSize_L (Descriptor size low)	0030098B (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			
DescDoor (Descriptor door)	0030098F (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)	00300990 (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)	00300991 (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)	00300992 (B)	D7	DMA_Running	1 DMA is running	0 DMA is not running	0	R	
		D6	PDREQ	PDREQ signal logic		0	R	
		D5	PDACK	PDACK 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)	00300994 (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)	00300995 (B)	D7	RcvLimitMode	1 Receive 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)	00300997 (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)	00300998 (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)	00300999 (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		

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Count_HH (DMA transfer byte counter high/high)	0030099C (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)	0030099D (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)	0030099E (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)	0030099F (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		

0x300B00–0x300B06

Serial Interface

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Serial I/F Ch.0 transmit data register (pEFSIF0_TXD)	00300B00 (B)	D7	TXD07	Serial I/F Ch.0 transmit data TXD07(06) = MSB TXD00 = LSB	0x0 to 0xFF(0x7F)		X	R/W	7-bit asynchronous mode does not use TXD07.		
		D6	TXD06								
		D5	TXD05								
		D4	TXD04								
		D3	TXD03								
		D2	TXD02								
		D1	TXD01								
		D0	TXD00								
		Serial I/F Ch.0 receive data register (pEFSIF0_RXD)	00300B01 (B)				D7			RXD07	Serial I/F Ch.0 receive data RXD07(06) = MSB RXD00 = LSB
D6	RXD06										
D5	RXD05										
D4	RXD04										
D3	RXD03										
D2	RXD02										
D1	RXD01										
D0	RXD00										
Serial I/F Ch.0 status register (pEFSIF0_STATUS)	00300B02 (B)			D7	RXD0NUM1	Number of Ch.0 receive data in FIFO	RXD0NUM[1:0]	Number of data	0	R	
		D6	RXD0NUM0	11	4		0				
				10	3						
				01	2						
				00	1 or 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 empty	1	Empty	0	Not empty	1	R	
D0	RDBF0	Ch.0 receive data buffer full	1	Full	0	Not full	0	R			
Serial I/F Ch.0 control register (pEFSIF0_CTL)	00300B03 (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 select	1	Odd	0	Even	X	R/W	
		D3	STPB0	Ch.0 stop bit select	1	2 bits	0	1 bit	X	R/W	
		D2	SCK0	Ch.0 input clock select	1	#SCLK0	0	Internal clock	X	R/W	
		D1	SMD01	Ch.0 transfer mode select	SMD0[1:0]		Transfer mode		X	R/W	
		D0	SMD00		11	8-bit asynchronous	X				
			10	7-bit asynchronous							
			01	Clock sync. Slave							
			00	Clock sync. Master							
Serial I/F Ch.0 IrDA register (pEFSIF0_IRDA)	00300B04 (B)	D7	SRDYCTL0	Ch.0 #SRDY control	1	High mask	0	Normal	0	R/W	Writing is disabled when SIOADV (D0/0x300B4F) = "0".
		D6	FIFOINT01	Ch.0 receive buffer full interrupt timing	FIFOINT0[1:0]		Receive level		0	R/W	
		D5	FIFOINT00		11	4	0				
					10	3					
					01	2					
		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 select	IRMD0[1:0]		I/F mode		X	R/W			
D0	IRMD00		11	reserved	X						
			10	IrDA 1.0							
			01	reserved							
			00	General I/F							
Serial I/F Ch.0 baud-rate timer control register (pEFSIF0_BRTRUN)	00300B05 (B)	D7–1	–	reserved	–		–	–	–	0 when being read.	
		D0	BRTRUN0	Baud-rate timer Run/Stop control	1	Run	0	Stop	0	R/W	
Serial I/F Ch.0 baud-rate timer reload data register (LSB) (pEFSIF0_BRTRDL)	00300B06 (B)	D7	BRTRD07	Serial I/F Ch.0 baud-rate timer reload data [7:0]	0x0 to 0xFF (BRTRD0[11:0] = 0x0 to 0xFFFF)		0	R/W			
		D6	BRTRD06								
		D5	BRTRD05								
		D4	BRTRD04								
		D3	BRTRD03								
		D2	BRTRD02								
		D1	BRTRD01								
		D0	BRTRD00								

0x300B07–0x300B13

Serial Interface

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
Serial I/F Ch.0 baud-rate timer reload data register (MSB) (pEFSIF0_BRTDM)	00300B07 (B)	D7–4	–	reserved	–	–	–	0 when being read.				
		D3	BTRD011	Serial I/F Ch.0	0x0 to 0xF	0	R/W					
		D2	BTRD010	baud-rate timer reload data [11:8]	(BTRD0[11:0] = 0x0 to 0xFFF)	0						
		D1	BTRD009			0						
D0	BTRD008	0										
Serial I/F Ch.0 baud-rate timer count data register (LSB) (pEFSIF0_BRTCDL)	00300B08 (B)	D7	BRTCD07	Serial I/F Ch.0		0x0 to 0xFF		0	R			
		D6	BRTCD06	baud-rate timer count data [7:0]	(BRTCD0[11:0] = 0x0 to 0xFFF)	0						
		D5	BRTCD05			0						
		D4	BRTCD04			0						
		D3	BRTCD03			0						
		D2	BRTCD02			0						
		D1	BRTCD01			0						
D0	BRTCD00	0										
Serial I/F Ch.0 baud-rate timer count data register (MSB) (pEFSIF0_BRTCDM)	00300B09 (B)	D7–4	–	reserved	–	–	–	0 when being read.				
		D3	BRTCD011	Serial I/F Ch.0	0x0 to 0xF	0	R					
		D2	BRTCD010	baud-rate timer count data [11:8]	(BRTCD0[11:0] = 0x0 to 0xFFF)	0						
		D1	BRTCD009			0						
D0	BRTCD008	0										
Serial I/F Ch.1 transmit data register (pEFSIF1_TXD)	00300B10 (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 (pEFSIF1_RXD)	00300B11 (B)	D7		RXD17		Serial I/F Ch.1 receive data		0x0 to 0xFF(0x7F)	X	R
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 (pEFSIF1_STATUS)	00300B12 (B)			D7	RXD1NUM1		Number of Ch.1 receive data in FIFO	RXD1NUM[1:0]		Number of data	0	R
		D6	RXD1NUM0	11	4			0				
				10	3							
				01	2							
				00	1 or 0							
		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		
		D2	OER1	Ch.1 overrun error flag	1	Error	0	Normal	0	R/W		
		D1	TDBE1	Ch.1 transmit data buffer empty	1	Empty	0	Not empty	1	R		
D0	RDBF1	Ch.1 receive data buffer full	1	Full	0	Not full	0	R				
Serial I/F Ch.1 control register (pEFSIF1_CTL)	00300B13 (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 select	1	Odd	0	Even	X	R/W		
		D3	STPB1	Ch.1 stop bit select	1	2 bits	0	1 bit	X	R/W		
		D2	SCLK1	Ch.1 input clock select	1	#SCLK1	0	Internal clock	X	R/W		
		D1	SMD11	Ch.1 transfer mode select	SMD1[1:0]	Transfer mode	X	X	R/W			
		D0	SMD10				11			8-bit asynchronous		
							10			7-bit asynchronous		
		01	Clock sync. Slave									
				00	Clock sync. Master							

0x300B14–0x300B1C

Serial Interface

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks			
Serial I/F Ch.1 IrDA register (pEFSIF1_IRDA)	00300B14 (B)	D7	SRDYCTL1	Ch.1 #SRDY control	1	High mask	0	Normal	0	R/W	Writing is disabled when SIOADV (D0/0x300B4F) = "0".	
		D6	FIFOINT11	Ch.1 receive buffer full interrupt timing	FIFOINT1[1:0]		Receive level		0	R/W		
		D5	FIFOINT10		11	4	0					
					10	3						
					01	2						
				00	1							
		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 select	IRMD1[1:0]		I/F mode		X	R/W				
D0	IRMD10		11	reserved								
			10	IrDA 1.0								
			01	reserved								
		00	General I/F									
Serial I/F Ch.1 baud-rate timer control register (pEFSIF1_BRTRUN)	00300B15 (B)	D7–1	–	reserved	–		–	–	0 when being read.			
		D0	BRTRUN1	Baud-rate timer Run/Stop control	1	Run	0	Stop	0	R/W		
Serial I/F Ch.1 baud-rate timer reload data register (LSB) (pEFSIF1_BRTRDL)	00300B16 (B)	D7	BRTRD17	Serial I/F Ch.1 baud-rate timer reload data [7:0]	0x0 to 0xFF		0	R/W				
		D6	BRTRD16		(BRTRD1[11:0] = 0x0 to 0xFFFF)		0					
		D5	BRTRD15		0							
		D4	BRTRD14		0							
		D3	BRTRD13		0							
		D2	BRTRD12		0							
		D1	BRTRD11		0							
		D0	BRTRD10		0							
Serial I/F Ch.1 baud-rate timer reload data register (MSB) (pEFSIF1_BRTRDM)	00300B17 (B)	D7–4	–	reserved	–		–	–	0 when being read.			
		D3	BRTRD111	Serial I/F Ch.1	0x0 to 0xF		0	R/W				
		D2	BRTRD110	baud-rate timer reload data [11:8]	(BRTRD1[11:0] = 0x0 to 0xFFFF)		0					
		D1	BRTRD19		0							
		D0	BRTRD18		0							
Serial I/F Ch.1 baud-rate timer count data register (LSB) (pEFSIF1_BRTCDL)	00300B18 (B)	D7	BRTCD17	Serial I/F Ch.1 baud-rate timer count data [7:0]	0x0 to 0xFF		0	R				
		D6	BRTCD16		(BRTCD1[11:0] = 0x0 to 0xFFFF)		0					
		D5	BRTCD15		0							
		D4	BRTCD14		0							
		D3	BRTCD13		0							
		D2	BRTCD12		0							
		D1	BRTCD11		0							
		D0	BRTCD10		0							
Serial I/F Ch.1 baud-rate timer count data register (MSB) (pEFSIF1_BRTCDM)	00300B19 (B)	D7–4	–	reserved	–		–	–	0 when being read.			
		D3	BRTCD111	Serial I/F Ch.1	0x0 to 0xF		0	R				
		D2	BRTCD110	baud-rate timer count data [11:8]	(BRTCD1[11:0] = 0x0 to 0xFFFF)		0					
		D1	BRTCD19		0							
		D0	BRTCD18		0							
Serial I/F Ch.1 ISO7816 mode control register (pEFSIF1_7816CTL)	00300B1A (B)	D7	RPNUM12	Serial I/F Ch.1 number of transmit repetition	0x0 to 0x7		0	R/W				
		D6	RPNUM11		0							
		D5	RPNUM10		0							
		D4	CLKOEN1	Ch.1 clock output enable	1	Enabled	0	Disabled	0	R/W		
		D3	CLKOL1	Ch.1 clock output forced low	1	Normal	0	Forced low	0	R/W		
		D2	MSBSEL1	Ch.1 MSB first selection	1	MSB first	0	LSB first	0	R/W		
		D1	7816MD11	Serial I/F Ch.1 ISO7816 mode selection	7816MD1[1:0]		Mode		0	R/W		
		D0	7816MD10		11	reserved	0					
		10	ISO7816, T = 1									
		01	ISO7816, T = 0									
		00	Normal I/F									
Serial I/F Ch.1 ISO7816 mode status register (pEFSIF1_7816STA)	00300B1B (B)	D7–1	–	reserved	–		–	–	0 when being read.			
		D0	TER1	Ch.1 ISO7816 transmit error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.	
Serial I/F Ch.1 ISO7816 mode FI/DI ratio register (LSB) (pEFSIF1_FIDIL)	00300B1C (B)	D7	FIDI17	Serial I/F Ch.1 ISO7816 mode FI/DI ratio [7:0]	0x0 to 0xFF		0	R/W	Valid only in ISO7816 mode.			
		D6	FIDI16		(FIDI1[13:0] = 0x0 to 0x3FFF)		0					
		D5	FIDI15		0							
		D4	FIDI14		0							
		D3	FIDI13		0							
		D2	FIDI12		0							
		D1	FIDI11		0							
		D0	FIDI10		0							

0x300B1D–0x300B23

Serial Interface

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Serial I/F Ch.1 ISO7816 mode FI/DI ratio register (MSB) (pEFSIF1_FIDIM)	00300B1D (B)	D7–6	–	reserved	–	–	–	0 when being read.	
		D5	FIDI113	Serial I/F Ch.1	0x0 to 0x3F	0	R/W	Valid only in ISO7816 mode.	
		D4	FIDI112	ISO7816 mode FI/DI ratio [13:8]	(FIDI1[13:0] = 0x0 to 0x3FFF)	0			
		D3	FIDI111			0			
		D2	FIDI110			0			
		D1	FIDI19			0			
		D0	FIDI18			0			
Serial I/F Ch.1 transmit time guard register (pEFSIF1_TTGR)	00300B1E (B)	D7	TTGR17	Serial I/F Ch.1 transmit time guard	0x0 to 0xFF	0	R/W		
		D6	TTGR16			0			
		D5	TTGR15			0			
		D4	TTGR14			0			
		D3	TTGR13			0			
		D2	TTGR12			0			
		D1	TTGR11			0			
		D0	TTGR10			0			
Serial I/F Ch.1 ISO7816 mode output clock setup register (pEFSIF1_CLKNUM)	00300B1F (B)	D7	CLKNEN1	Ch.1 CLKN enable	1 Enabled 0 Disabled	0	R/W		
		D6	CLKN16	Serial I/F Ch.1	0x0 to 0x7F	0	R/W		
		D5	CLKN15	number of output clocks		0			
		D4	CLKN14			0			
		D3	CLKN13			0			
		D2	CLKN12			0			
		D0	CLKN10			0			
Serial I/F Ch.2 transmit data register (pEFSIF2_TXD)	00300B20 (B)	D7	TXD27	Serial I/F Ch.2 transmit data	0x0 to 0xFF(0x7F)	X	R/W	7-bit asynchronous mode does not use TXD27.	
		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 (pEFSIF2_RXD)	00300B21 (B)	D7	RXD27	Serial I/F Ch.2 receive data	0x0 to 0xFF(0x7F)	X	R	7-bit asynchronous mode does not use RXD27 (fixed at 0).	
		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 (pEFSIF2_STATUS)	00300B22 (B)	D7	RXD2NUM1	Number of Ch.2 receive data in FIFO	RXD2NUM[1:0]	Number of data	0	R	
		D6	RXD2NUM0		11	4	0		
					10	3			
					01	2			
				00	1 or 0				
		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	
		D2	OER2	Ch.2 overrun error flag	1 Error	0 Normal	0	R/W	
		D1	TDBE2	Ch.2 transmit data buffer empty	1 Empty	0 Not empty	1	R	
	D0	RDBF2	Ch.2 receive data buffer full	1 Full	0 Not full	0	R		
Serial I/F Ch.2 control register (pEFSIF2_CTL)	00300B23 (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	
		D4	PMD2	Ch.2 parity mode select	1 Odd	0 Even	X	R/W	
		D3	STPB2	Ch.2 stop bit select	1 2 bits	0 1 bit	X	R/W	
		D2	SSCK2	Ch.2 input clock select	1 #SCLK2	0 Internal clock	X	R/W	
		D1	SMD21	Ch.2 transfer mode select	SMD2[1:0]	Transfer mode	X	R/W	
		D0	SMD20		11	8-bit asynchronous	X		
			10	7-bit asynchronous					
			01	Clock sync. Slave					
			00	Clock sync. Master					

0x300B24–0x300B4F

Serial Interface

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks		
Serial I/F Ch.2 IrDA register (pEFSIF2_IRDA)	00300B24 (B)	D7	SRDYCTL2	Ch.2 #SRDY control	1	High mask	0	Normal	0	R/W	Writing is disabled when SIOADV (D0/0x300B4F) = "0".	
		D6	FIFOINT21	Ch.2 receive buffer full interrupt timing	FIFOINT2[1:0]		Receive level		0	R/W		
		D5	FIFOINT20		11	4	0					
					10	3						
					01	2						
				00	1							
		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 select	IRMD2[1:0]		I/F mode		X	R/W				
D0	IRMD20		11	reserved								
			10	IrDA 1.0								
			01	reserved								
		00	General I/F									
Serial I/F Ch.2 baud-rate timer control register (pEFSIF2_BRTRUN)	00300B25 (B)	D7–1	–	reserved	–			–	–	0 when being read.		
		D0	BRTRUN2	Baud-rate timer Run/Stop control	1	Run	0	Stop	0	R/W		
Serial I/F Ch.2 baud-rate timer reload data register (LSB) (pEFSIF2_BRTRDL)	00300B26 (B)	D7	BRTRD27	Serial I/F Ch.2 baud-rate timer reload data [7:0]	0x0 to 0xFF			0	R/W			
		D6	BRTRD26		(BRTRD2[11:0] = 0x0 to 0xFFFF)			0				
		D5	BRTRD25		0							
		D4	BRTRD24		0							
		D3	BRTRD23		0							
		D2	BRTRD22		0							
		D1	BRTRD21		0							
		D0	BRTRD20		0							
Serial I/F Ch.2 baud-rate timer reload data register (MSB) (pEFSIF2_BRTRDM)	00300B27 (B)	D7–4	–	reserved	–			–	–	0 when being read.		
		D3	BRTRD211	Serial I/F Ch.2 baud-rate timer reload data [11:8]	0x0 to 0xF			0	R/W			
		D2	BRTRD210		(BRTRD2[11:0] = 0x0 to 0xFFFF)			0				
		D1	BRTRD29		0							
		D0	BRTRD28		0							
Serial I/F Ch.2 baud-rate timer count data register (LSB) (pEFSIF2_BRTCDL)	00300B28 (B)	D7	BRTCD27	Serial I/F Ch.2 baud-rate timer count data [7:0]	0x0 to 0xFF			0	R			
		D6	BRTCD26		(BRTCD2[11:0] = 0x0 to 0xFFFF)			0				
		D5	BRTCD25		0							
		D4	BRTCD24		0							
		D3	BRTCD23		0							
		D2	BRTCD22		0							
		D1	BRTCD21		0							
		D0	BRTCD20		0							
Serial I/F Ch.2 baud-rate timer count data register (MSB) (pEFSIF2_BRTCDM)	00300B29 (B)	D7–4	–	reserved	–			–	–	0 when being read.		
		D3	BRTCD211	Serial I/F Ch.2 baud-rate timer count data [11:8]	0x0 to 0xF			0	R			
		D2	BRTCD210		(BRTCD2[11:0] = 0x0 to 0xFFFF)			0				
		D1	BRTCD29		0							
		D0	BRTCD28		0							
Serial I/F STD/ADV mode select register (pEFSIF_ADV)	00300B4F (B)	D7–1	–	reserved	–			–	–	Writing 1 not allowed.		
		D0	SIOADV	Standard mode/advanced mode select	1	Advanced mode	0	Standard mode	0	R/W		

0x300C00–0x300C21

Extended Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PA I/O control register (pPA_IOC)	00300C00 (B)	D7–5	–	reserved	–	–	–	0 when being read.	
		D4	IOCA4	PA4 I/O control	1 Output	0 Input	0	R/W	
		D3	IOCA3	PA3 I/O control			0	R/W	
		D2	IOCA2	PA2 I/O control			0	R/W	
		D1	IOCA1	PA1 I/O control			0	R/W	
		D0	IOCA0	PA0 I/O control			0	R/W	
PA I/O port data register (pPA_DATA)	00300C01 (B)	D7–5	–	reserved	–	–	–	0 when being read.	
		D4	PA4D	PA4 I/O port data	1 High	0 Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D3	PA3D	PA3 I/O port data			Ext.	R/W	
		D2	PA2D	PA2 I/O port data			Ext.	R/W	
		D1	PA1D	PA1 I/O port data			Ext.	R/W	
		D0	PA0D	PA0 I/O port data			Ext.	R/W	
PB I/O control register (pPB_IOC)	00300C02 (B)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3	IOCB3	PB3 I/O control	1 Output	0 Input	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			
PB I/O port data register (pPB_DATA)	00300C03 (B)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3	PB3D	PB3 I/O port data	1 High	0 Low	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D2	PB2D	PB2 I/O port data			Ext.	R/W	
		D1	PB1D	PB1 I/O port data			Ext.	R/W	
		D0	PB0D	PB0 I/O port data			Ext.	R/W	
PC I/O control register (pPC_IOC)	00300C04 (B)	D7	IOCC7	PC7 I/O control	1 Output	0 Input	0	R/W	
		D6	IOCC6	PC6 I/O control			0	R/W	
		D5	IOCC5	PC5 I/O control			0	R/W	
		D4	IOCC4	PC4 I/O control			0	R/W	
		D3	IOCC3	PC3 I/O control			0	R/W	
		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	
		PC I/O port data register (pPC_DATA)	00300C05 (B)	D7	PC7D	PC7 I/O port data	1 High	0 Low	
D6	PC6D			PC6 I/O port data			Ext.	R/W	
D5	PC5D			PC5 I/O port data			Ext.	R/W	
D4	PC4D			PC4 I/O port data			Ext.	R/W	
D3	PC3D			PC3 I/O port data			Ext.	R/W	
D2	PC2D			PC2 I/O port data			Ext.	R/W	
D1	PC1D			PC1 I/O port data			Ext.	R/W	
D0	PC0D			PC0 I/O port data			Ext.	R/W	
PA0–PA3 port function select register (pPA_03_CFP)	00300C20 (B)			D7 D6	CFPA31 CFPA30	PA3 port extended function	CFPA3[1:0]	Function	0
		11	CARD0						
		10	TFT_CTL2						
		01	FPDAT10						
		00	PA3						
		D5 D4	CFPA21 CFPA20	PA2 port extended function	CFPA2[1:0]	Function	0	R/W	
					11	reserved			
					10	TFT_CTL1			
					01	FPDAT9			
		00	PA2						
		D3 D2	CFPA11 CFPA10	PA1 port extended function	CFPA1[1:0]	Function	0	R/W	
					11	reserved			
10	TFT_CTL0								
01	FPDAT8								
00	PA1								
D1 D0	CFPA01 CFPA00	PA0 port extended function	CFPA0[1:0]	Function	0	R/W			
			1*	reserved					
			01	TFT_CTL0					
			00	PA0					
PA4 port function select register (pPA_4_CFP)	00300C21 (B)	D7–2	–	reserved	–	–	–	0 when being read.	
		D1 D0	CFPA41 CFPA40	PA4 port extended function	CFPA4[1:0]	Function	0	R/W	
					11	CARD1			
					10	TFT_CTL3			
					01	FPDAT11			
					00	PA4			

0x300C22–0x300C25

Extended Ports

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PB0–PB3 port function select register (pPB_03_CFP)	00300C22 (B)	D7 D6	CFPB31	PB3 port extended function	CFPB3[1:0]	Function	0	R/W	
			CFPB30		11	CARD5			
					10	I2S_MCLK			
					01	FPDAT11			
		D5 D4	CFPB21	PB2 port extended function	CFPB2[1:0]	Function	0	R/W	
			CFPB20		11	CARD4			
					10	I2S_SCK			
					01	FPDAT10			
		D3 D2	CFPB11	PB1 port extended function	CFPB1[1:0]	Function	0	R/W	
			CFPB10		11	CARD3			
					10	I2S_WS			
					01	FPDAT9			
D1 D0	CFPB01	PB0 port extended function	CFPB0[1:0]	Function	0	R/W			
	CFPB00		11	CARD2					
			10	I2S_SDO					
			01	FPDAT8					
PC0–PC3 port function select register (pPC_03_CFP)	00300C24 (B)	D7 D6	CFPC31	PC3 port extended function	CFPC3[1:0]	Function	0	R/W	
			CFPC30		1*	reserved			
					01	PC3			
					00	D11			
		D5 D4	CFPC21	PC2 port extended function	CFPC2[1:0]	Function	0	R/W	
			CFPC20		1*	reserved			
					01	PC2			
					00	D10			
		D3 D2	CFPC11	PC1 port extended function	CFPC1[1:0]	Function	0	R/W	
			CFPC10		1*	reserved			
					01	PC1			
					00	D9			
D1 D0	CFPC01	PC0 port extended function	CFPC0[1:0]	Function	0	R/W			
	CFPC00		1*	reserved					
			01	PC0					
			00	D8					
PC4–PC7 port function select register (pPC_47_CFP)	00300C25 (B)	D7 D6	CFPC71	PC7 port extended function	CFPC7[1:0]	Function	0	R/W	
			CFPC70		1*	reserved			
					01	PC7			
					00	D15			
		D5 D4	CFPC61	PC6 port extended function	CFPC6[1:0]	Function	0	R/W	
			CFPC60		1*	reserved			
					01	PC6			
					00	D14			
		D3 D2	CFPC51	PC5 port extended function	CFPC5[1:0]	Function	0	R/W	
			CFPC50		1*	reserved			
					01	PC5			
					00	D13			
D1 D0	CFPC41	PC4 port extended function	CFPC4[1:0]	Function	0	R/W			
	CFPC40		1*	reserved					
			01	PC4					
			00	D12					

0x300C40–0x300C48 **Misc Register**

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Bus signal pull-up control register (pMISC_BUSPUP)	00300C40 (B)	D7–0	–	reserved	–		–	–	0 when being read.		
Bus signal low drive control register (pMISC_BUSLOW)	00300C41 (B)	D7–4	–	reserved	–		–	–	0 when being read.		
		D3	LDRVDB	D15–D0 low drive	1	Low drive	0	Normal output		0	R/W
		D2	LDRVCE	#CE11–#CE4 low drive						0	R/W
		D1	LDRVAD	A24–A0 low drive						0	R/W
D0	LDRVVRW	#RD,#WRL,#WRH,#BSL low drive	0	R/W							
P0 pull-up control register (pMISC_PUP0)	00300C42 (B)	D7	PUP07	P07 pull-up	1	Pulled up	0	No pull-up	0	R/W	
		D6	PUP06	P06 pull-up					0	R/W	
		D5	PUP05	P05 pull-up					0	R/W	
		D4	PUP04	P04 pull-up					0	R/W	
		D3	PUP03	P03 pull-up					0	R/W	
		D2	PUP02	P02 pull-up					0	R/W	
		D1	PUP01	P01 pull-up					0	R/W	
		D0	PUP00	P00 pull-up					0	R/W	
P1 pull-up control register (pMISC_PUP1)	00300C43 (B)	D7	PUP17	P17 pull-up	1	Pulled up	0	No pull-up	1	R/W	
		D6	PUP16	P16 pull-up					1	R/W	
		D5	PUP15	P15 pull-up					1	R/W	
		D4	PUP14	P14 pull-up					0	R/W	
		D3	PUP13	P13 pull-up					0	R/W	
		D2	PUP12	P12 pull-up					0	R/W	
		D1	PUP11	P11 pull-up					0	R/W	
		D0	PUP10	P10 pull-up					0	R/W	
P2 pull-up control register (pMISC_PUP2)	00300C44 (B)	D7	PUP27	P27 pull-up	1	Pulled up	0	No pull-up	1	R/W	
		D6	PUP26	P26 pull-up					1	R/W	
		D5	PUP25	P25 pull-up					1	R/W	
		D4	PUP24	P24 pull-up					1	R/W	
		D3	PUP23	P23 pull-up					1	R/W	
		D2	PUP22	P22 pull-up					1	R/W	
		D1	PUP21	P21 pull-up					1	R/W	
		D0	PUP20	P20 pull-up					1	R/W	
P3 pull-up control register (pMISC_PUP3)	00300C45 (B)	D7	–	reserved	–		–	–	1 when being read.		
		D6	PUP36	P36 pull-up	1	Pulled up	0	No pull-up		1	R/W
		D5	PUP35	P35 pull-up						1	R/W
		D4	PUP34	P34 pull-up						1	R/W
		D3	PUP33	P33 pull-up						0	R/W
		D2	PUP32	P32 pull-up						0	R/W
		D1	PUP31	P31 pull-up						0	R/W
		D0	PUP30	P30 pull-up						0	R/W
P4 pull-up control register (pMISC_PUP4)	00300C46 (B)	D7	PUP47	P47 pull-up					1	Pulled up	0
		D6	PUP46	P46 pull-up	1	R/W					
		D5	PUP45	P45 pull-up	1	R/W					
		D4	PUP44	P44 pull-up	1	R/W					
		D3	PUP43	P43 pull-up	1	R/W					
		D2	PUP42	P42 pull-up	1	R/W					
		D1	PUP41	P41 pull-up	1	R/W					
		D0	PUP40	P40 pull-up	1	R/W					
P5 pull-up control register (pMISC_PUP5)	00300C47 (B)	D7	PUP57	P57 pull-up	1	Pulled up	0	No pull-up	1	R/W	
		D6	PUP56	P56 pull-up					1	R/W	
		D5	PUP55	P55 pull-up					1	R/W	
		D4	PUP54	P54 pull-up					1	R/W	
		D3	PUP53	P53 pull-up					1	R/W	
		D2	PUP52	P52 pull-up					1	R/W	
		D1	PUP51	P51 pull-up					1	R/W	
		D0	PUP50	P50 pull-up					1	R/W	
P6 pull-up control register (pMISC_PUP6)	00300C48 (B)	D7	PUP67	P67 pull-up	1	Pulled up	0	No pull-up	0	R/W	
		D6	PUP66	P66 pull-up					0	R/W	
		D5	PUP65	P65 pull-up					0	R/W	
		D4	PUP64	P64 pull-up					0	R/W	
		D3	PUP63	P63 pull-up					0	R/W	
		D2	PUP62	P62 pull-up					0	R/W	
		D1	PUP61	P61 pull-up					0	R/W	
		D0	PUP60	P60 pull-up					0	R/W	

0x300C49–0x300C4D

Misc Register

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
P7 pull-up control register (pMISC_PUP7)	00300C49 (B)	D7–5	–	reserved	–		–	–	1 when being read.	
		D4	PUP74	P74 pull-up	1	Pulled up	0	No pull-up	0	R/W
		D3	PUP73	P73 pull-up					0	R/W
		D2	PUP72	P72 pull-up					0	R/W
		D1	PUP71	P71 pull-up					0	R/W
		D0	PUP70	P70 pull-up					0	R/W
P8 pull-up control register (pMISC_PUP8)	00300C4A (B)	D7–6	–	reserved	–		–	–	1 when being read.	
		D5	PUP85	P85 pull-up	1	Pulled up	0	No pull-up	0	R/W
		D4	PUP84	P84 pull-up					0	R/W
		D3	PUP83	P83 pull-up					0	R/W
		D2	PUP82	P82 pull-up					0	R/W
		D1	PUP81	P81 pull-up					0	R/W
		D0	PUP80	P80 pull-up					0	R/W
P9 pull-up control register (pMISC_PUP9)	00300C4B (B)	D7	PUP97	P97 pull-up	1	Pulled up	0	No pull-up	0	R/W
		D6	PUP96	P96 pull-up					0	R/W
		D5	PUP95	P95 pull-up					0	R/W
		D4	PUP94	P94 pull-up					0	R/W
		D3	PUP93	P93 pull-up					0	R/W
		D2	PUP92	P92 pull-up					0	R/W
		D1	PUP91	P91 pull-up					0	R/W
		D0	PUP90	P90 pull-up					0	R/W
PA pull-up control register (pMISC_PUPA)	00300C4C (B)	D7–5	–	reserved	–		–	–	1 when being read.	
		D4	PUPA4	PA4 pull-up	1	Pulled up	0	No pull-up	1	R/W
		D3	PUPA3	PA3 pull-up					1	R/W
		D2	PUPA2	PA2 pull-up					1	R/W
		D1	PUPA1	PA1 pull-up					1	R/W
		D0	PUPA0	PA0 pull-up					1	R/W
PB pull-up control register (pMISC_PUPB)	00300C4D (B)	D7–4	–	reserved	–		–	–	1 when being read.	
		D3	PUPB3	PB3 pull-up	1	Pulled up	0	No pull-up	1	R/W
		D2	PUPB2	PB2 pull-up					1	R/W
		D1	PUPB1	PB1 pull-up					1	R/W
		D0	PUPB0	PB0 pull-up					1	R/W

0x301100–0x301105

Intelligent DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
IDMA base address register 0 (piDMABASE)	00301100 (HW)	D15	DBASEL15	IDMA base address low-order 16 bits (Initial value: 0x200003A0)		0	R/W				
		D14	DBASEL14								
		D13	DBASEL13								
		D12	DBASEL12								
		D11	DBASEL11								
		D10	DBASEL10								
		D9	DBASEL9								
		D8	DBASEL8								
		D7	DBASEL7								
		D6	DBASEL6								
		D5	DBASEL5								
		D4	DBASEL4								
		D3	DBASEL3								
		D2	DBASEL2								
		D1	DBASEL1								
		D0	DBASEL0								
IDMA base address register 1	00301102 (HW)	D15	DBASEH15	IDMA base address high-order 16 bits (Initial value: 0x200003A0)		0	R/W				
		D14	DBASEH14								
		D13	DBASEH13								
		D12	DBASEH12								
		D11	DBASEH11								
		D10	DBASEH10								
		D9	DBASEH9								
		D8	DBASEH8								
		D7	DBASEH7								
		D6	DBASEH6								
		D5	DBASEH5								
		D4	DBASEH4								
		D3	DBASEH3								
		D2	DBASEH2								
		D1	DBASEH1								
		D0	DBASEH0								
IDMA start register (piDMA_START)	00301104 (B)	D7	DSTART	IDMA start	1	IDMA start	0	Stop	0	R/W	
		D6	DCHN6	IDMA channel number	0 to 127				0	R/W	
		D5	DCHN5								
		D4	DCHN4								
		D3	DCHN3								
		D2	DCHN2								
		D1	DCHN1								
D0	DCHN0										
IDMA enable register (piDMA_EN)	00301105 (B)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	IDMAEN	IDMA enable (for software trigger)	1	Enabled	0	Disabled	0	R/W	

0x301120–0x301126

High-Speed DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
HSDMA Ch.0 transfer counter register (pHS0_CNT)	00301120 (HW)	D15	TC0_L7	Ch.0 transfer counter[7:0] (block transfer mode)			0	R/W		
		D14	TC0_L6							
		D13	TC0_L5							
		D12	TC0_L4							
		D11	TC0_L3	Ch.0 transfer counter[15:8] (single/successive transfer mode)			0			
		D10	TC0_L2				0			
		D9	TC0_L1				0			
		D8	TC0_L0				0			
		D7	BLKLEN07	Ch.0 block length (block transfer mode)			0			R/W
		D6	BLKLEN06				0			
		D5	BLKLEN05				0			
		D4	BLKLEN04	Ch.0 transfer counter[7:0] (single/successive transfer mode)			0			
		D3	BLKLEN03				0			
		D2	BLKLEN02				0			
		D1	BLKLEN01				0			
		D0	BLKLEN00	0						
HSDMA Ch.0 control register	00301122 (HW)	D15	DUALM0	Ch.0 address mode selection	1 Dual addr 0 Single addr	0	R/W			
		D14	D0DIR	D) Invalid	–	–	–			
			S) Ch.0 transfer direction control	1 Memory WR 0 Memory RD	0	R/W				
		D13–8	–	reserved	–	–	–	0 when being read.		
		D7	TC0_H7	Ch.0 transfer counter[15:8] (block transfer mode)	0	R/W				
		D6	TC0_H6		0					
		D5	TC0_H5	Ch.0 transfer counter[23:16] (single/successive transfer mode)	0					
		D4	TC0_H4		0					
		D3	TC0_H3		0					
		D2	TC0_H2		0					
D1	TC0_H1	0								
D0	TC0_H0	0								
HSDMA Ch.0 low-order source address setup register (pHS0_SADR)	00301124 (HW)	D15	S0ADRL15	D) Ch.0 source address[15:0]			0	R/W		
		D14	S0ADRL14	S) Ch.0 memory address[15:0]			0			
		D13	S0ADRL13	0						
		D12	S0ADRL12	0						
		D11	S0ADRL11	0						
		D10	S0ADRL10	0						
		D9	S0ADRL9	0						
		D8	S0ADRL8	0						
		D7	S0ADRL7	0						
		D6	S0ADRL6	0						
		D5	S0ADRL5	0						
		D4	S0ADRL4	0						
		D3	S0ADRL3	0						
		D2	S0ADRL2	0						
D1	S0ADRL1	0								
D0	S0ADRL0	0								
HSDMA Ch.0 high-order source address setup register	00301126 (HW)	D15	–	reserved	–	–	–	0 when being read.		
		D14	DATSIZE0	Ch.0 transfer data size	1 Half word 0 Byte	0	R/W			
		D13	S0IN1	D) Ch.0 source address control	S0IN[1:0]	Inc/dec	0	R/W		
		D12	S0IN0	S) Ch.0 memory address control	11 Inc.(no init) 10 Inc.(init) 01 Dec.(no init) 00 Fixed	0				
		D11	S0ADRH11	D) Ch.0 source address[27:16]			0	R/W		
		D10	S0ADRH10	S) Ch.0 memory address[27:16]			0			
		D9	S0ADRH9	0						
		D8	S0ADRH8	0						
		D7	S0ADRH7	0						
		D6	S0ADRH6	0						
		D5	S0ADRH5	0						
		D4	S0ADRH4	0						
		D3	S0ADRH3	0						
		D2	S0ADRH2	0						
		D1	S0ADRH1	0						
		D0	S0ADRH0	0						

0x301128–0x301130 High-Speed DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
HSDMA Ch.0 low-order destination address setup register (pHS0_DADR) Note: D) Dual address mode S) Single address mode	00301128 (HW)	D15	D0ADRL15	D) Ch.0 destination address[15:0] S) Invalid		0	R/W				
		D14	D0ADRL14								
		D13	D0ADRL13								
		D12	D0ADRL12								
		D11	D0ADRL11								
		D10	D0ADRL10								
		D9	D0ADRL9								
		D8	D0ADRL8								
		D7	D0ADRL7								
		D6	D0ADRL6								
		D5	D0ADRL5								
		D4	D0ADRL4								
		D3	D0ADRL3								
		D2	D0ADRL2								
		D1	D0ADRL1								
		D0	D0ADRL0								
		HSDMA Ch.0 high-order destination address setup register Note: D) Dual address mode S) Single address mode	0030112A (HW)						D15	D0MOD1	Ch.0 transfer mode
D14	D0MOD0										
				11 Invalid							
				10 Block							
				01 Successive							
				00 Single							
				D13	D0IN1	D) Ch.0 destination address control S) Invalid	D0IN[1:0]	Inc/dec	0	R/W	
				D12	D0IN0						
											11 Inc.(no init)
											10 Inc.(init)
						01 Dec.(no init)					
						00 Fixed					
				D11	D0ADRH11	D) Ch.0 destination address[27:16] S) Invalid		0	R/W		
				D10	D0ADRH10						
				D9	D0ADRH9						
				D8	D0ADRH8						
				D7	D0ADRH7						
				D6	D0ADRH6						
D5	D0ADRH5										
D4	D0ADRH4										
D3	D0ADRH3										
D2	D0ADRH2										
D1	D0ADRH1										
D0	D0ADRH0										
HSDMA Ch.0 enable register (pHS0_EN)	0030112C (HW)	D15–1	–	reserved	–	–	–	0 when being read.			
		D0	HS0_EN	Ch.0 enable	1 Enable 0 Disable	0	R/W				
HSDMA Ch.0 trigger flag register (pHS0_TF)	0030112E (HW)	D15–1	–	reserved	–	–	–	0 when being read.			
		D0	HS0_TF	Ch.0 trigger flag clear (writing) Ch.0 trigger flag status (reading)	1 Clear 0 No operation 1 Set 0 Cleared	0	R/W				
HSDMA Ch.1 transfer counter register (pHS1_CNT)	00301130 (HW)	D15	TC1_L7	Ch.1 transfer counter[7:0] (block transfer mode)		0	R/W				
		D14	TC1_L6								
		D13	TC1_L5								
		D12	TC1_L4								
		Ch.1 transfer counter[15:8] (single/successive transfer mode)	D11	TC1_L3							
			D10	TC1_L2							
			D9	TC1_L1							
			D8	TC1_L0							
		Ch.1 block length (block transfer mode)	D7	BLKLEN17							
			D6	BLKLEN16							
			D5	BLKLEN15							
			Ch.1 transfer counter[7:0] (single/successive transfer mode)	D4					BLKLEN14		
				D3					BLKLEN13		
				D2					BLKLEN12		
				D1					BLKLEN11		
			D0	BLKLEN10							

0x301132–0x301138

High-Speed DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
HSDMA Ch.1 control register Note: D) Dual address mode S) Single address mode	00301132 (HW)	D15	DUALM1	Ch.1 address mode selection	1 Dual addr 0 Single addr	0	R/W		
		D14	D1DIR	D) Invalid S) Ch.1 transfer direction control	– 1 Memory WR 0 Memory RD	– 0	– R/W		
		D13–8	–	reserved	–	–	–	0 when being read.	
		D7	TC1_H7	Ch.1 transfer counter[15:8] (block transfer mode)			0	R/W	
		D6	TC1_H6				0		
		D5	TC1_H5				0		
		D4	TC1_H4	Ch.1 transfer counter[23:16] (single/successive transfer mode)			0		
		D3	TC1_H3				0		
		D2	TC1_H2				0		
		D1	TC1_H1				0		
		D0	TC1_H0				0		
HSDMA Ch.1 low-order source address setup register (pHS1_SADR) Note: D) Dual address mode S) Single address mode	00301134 (HW)	D15	S1ADRL15	D) Ch.1 source address[15:0]		0	R/W		
		D14	S1ADRL14	S) Ch.1 memory address[15:0]		0			
		D13	S1ADRL13			0			
		D12	S1ADRL12			0			
		D11	S1ADRL11			0			
		D10	S1ADRL10			0			
		D9	S1ADRL9			0			
		D8	S1ADRL8			0			
		D7	S1ADRL7			0			
		D6	S1ADRL6			0			
		D5	S1ADRL5			0			
D4	S1ADRL4			0					
D3	S1ADRL3			0					
D2	S1ADRL2			0					
D1	S1ADRL1			0					
D0	S1ADRL0			0					
HSDMA Ch.1 high-order source address setup register Note: D) Dual address mode S) Single address mode	00301136 (HW)	D15	–	reserved	–	–	–	0 when being read.	
		D14	DATSIZE1	Ch.1 transfer data size	1 Half word 0 Byte	0	R/W		
		D13	S1IN1	D) Ch.1 source address control	S1IN[1:0]	Inc/dec	0	R/W	
		D12	S1IN0	S) Ch.1 memory address control	11	Inc.(no init)	0		
					10	Inc.(init)			
					01	Dec.(no init)			
					00	Fixed			
		D11	S1ADRH11	D) Ch.1 source address[27:16]			0	R/W	
		D10	S1ADRH10	S) Ch.1 memory address[27:16]			0		
		D9	S1ADRH9			0			
		D8	S1ADRH8			0			
D7	S1ADRH7			0					
D6	S1ADRH6			0					
D5	S1ADRH5			0					
D4	S1ADRH4			0					
D3	S1ADRH3			0					
D2	S1ADRH2			0					
D1	S1ADRH1			0					
D0	S1ADRH0			0					
HSDMA Ch.1 low-order destination address setup register (pHS1_DADR) Note: D) Dual address mode S) Single address mode	00301138 (HW)	D15	D1ADRL15	D) Ch.1 destination address[15:0]		0	R/W		
		D14	D1ADRL14	S) Invalid		0			
		D13	D1ADRL13			0			
		D12	D1ADRL12			0			
		D11	D1ADRL11			0			
		D10	D1ADRL10			0			
		D9	D1ADRL9			0			
		D8	D1ADRL8			0			
		D7	D1ADRL7			0			
		D6	D1ADRL6			0			
		D5	D1ADRL5			0			
D4	D1ADRL4			0					
D3	D1ADRL3			0					
D2	D1ADRL2			0					
D1	D1ADRL1			0					
D0	D1ADRL0			0					

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
HSDMA Ch.1 high-order destination address setup register Note: D) Dual address mode S) Single address mode	0030113A (HW)	D15	D1MOD1	Ch.1 transfer mode	D1MOD[1:0]	Mode	0	R/W			
		D14	D1MOD0		11	Invalid	0				
					10	Block					
					01	Successive					
		D13	D1IN1	D) Ch.1 destination address control S) Invalid	D1IN[1:0]	Inc/dec	0	R/W			
		D12	D1IN0		11	Inc.(no init)	0				
					10	Inc.(init)					
					01	Dec.(no init)					
		D11	D1ADRH11	D) Ch.1 destination address[27:16] S) Invalid			0	R/W			
		D10	D1ADRH10				0				
		D9	D1ADRH9				0				
		D8	D1ADRH8				0				
		D7	D1ADRH7				0				
D6	D1ADRH6	0									
D5	D1ADRH5	0									
D4	D1ADRH4	0									
D3	D1ADRH3	0									
D2	D1ADRH2	0									
D1	D1ADRH1	0									
D0	D1ADRH0	0									
HSDMA Ch.1 enable register (pHS1_EN)	0030113C (HW)	D15–1	–	reserved	–	–	–	0 when being read.			
		D0	HS1_EN	Ch.1 enable	1 Enable	0 Disable	0	R/W			
HSDMA Ch.1 trigger flag register (pHS1_TF)	0030113E (HW)	D15–1	–	reserved	–	–	–	0 when being read.			
		D0	HS1_TF	Ch.1 trigger flag clear (writing) Ch.1 trigger flag status (reading)	1 Clear 1 Set	0 No operation 0 Cleared	0	R/W			
HSDMA Ch.2 transfer counter register (pHS2_CNT)	00301140 (HW)	D15	TC2_L7	Ch.2 transfer counter[7:0] (block transfer mode)			0	R/W			
		D14	TC2_L6				0				
		D13	TC2_L5				0				
		D12	TC2_L4	Ch.2 transfer counter[15:8] (single/successive transfer mode)			0				
		D11	TC2_L3	0							
		D10	TC2_L2	0							
		D9	TC2_L1	0							
		D8	TC2_L0	0							
		D7	BLKLEN27	Ch.2 block length (block transfer mode)						0	R/W
		D6	BLKLEN26							0	
		D5	BLKLEN25	0							
		D4	BLKLEN24	Ch.2 transfer counter[7:0] (single/successive transfer mode)						0	
		D3	BLKLEN23	0							
		D2	BLKLEN22	0							
D1	BLKLEN21	0									
D0	BLKLEN20	0									
HSDMA Ch.2 control register Note: D) Dual address mode S) Single address mode	00301142 (HW)	D15	DUALM2	Ch.2 address mode selection	1 Dual addr	0 Single addr		0		R/W	
		D14	D2DIR	D) Invalid S) Ch.2 transfer direction control	–	–		–		–	
		D13–8	–	reserved	–	–		–		–	0 when being read.
		D7	TC2_H7	Ch.2 transfer counter[15:8] (block transfer mode)				0		R/W	
		D6	TC2_H6					0			
		D5	TC2_H5	0							
		D4	TC2_H4	Ch.2 transfer counter[23:16] (single/successive transfer mode)			0				
		D3	TC2_H3	0							
D2	TC2_H2	0									
D1	TC2_H1	0									
D0	TC2_H0	0									

0x301144–0x301148

High-Speed DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks				
HSDMA Ch.2 low-order source address setup register (pHS2_SADR) Note: D) Dual address mode S) Single address mode	00301144 (HW)	D15	S2ADRL15	D) Ch.2 source address[15:0]			0	R/W				
		D14	S2ADRL14	S) Ch.2 memory address[15:0]								
		D13	S2ADRL13									
		D12	S2ADRL12									
		D11	S2ADRL11									
		D10	S2ADRL10									
		D9	S2ADRL9									
		D8	S2ADRL8									
		D7	S2ADRL7									
		D6	S2ADRL6									
		D5	S2ADRL5									
		D4	S2ADRL4									
		D3	S2ADRL3									
		D2	S2ADRL2									
		D1	S2ADRL1									
		D0	S2ADRL0									
HSDMA Ch.2 high-order source address setup register Note: D) Dual address mode S) Single address mode	00301146 (HW)	D15	–	reserved	–	–	–	–	0 when being read.			
		D14	DATSIZE2	Ch.2 transfer data size	1 Half word	0 Byte	0	R/W				
		D13	S2IN1	D) Ch.2 source address control	S2IN[1:0]		0	R/W				
					D12	S2IN0				S) Ch.2 memory address control	11	Inc.(no init)
											10	Inc.(init)
											01	Dec.(no init)
		00	Fixed									
		D11	S2ADRH11	D) Ch.2 source address[27:16]				0	R/W			
		D10	S2ADRH10	S) Ch.2 memory address[27:16]								
		D9	S2ADRH9									
D8	S2ADRH8											
D7	S2ADRH7											
D6	S2ADRH6											
D5	S2ADRH5											
D4	S2ADRH4											
D3	S2ADRH3											
D2	S2ADRH2											
D1	S2ADRH1											
D0	S2ADRH0											
HSDMA Ch.2 low-order destination address setup register (pHS2_DADR) Note: D) Dual address mode S) Single address mode	00301148 (HW)	D15	D2ADRL15	D) Ch.2 destination address[15:0]			0	R/W				
		D14	D2ADRL14	S) Invalid								
		D13	D2ADRL13									
		D12	D2ADRL12									
		D11	D2ADRL11									
		D10	D2ADRL10									
		D9	D2ADRL9									
		D8	D2ADRL8									
		D7	D2ADRL7									
		D6	D2ADRL6									
		D5	D2ADRL5									
		D4	D2ADRL4									
		D3	D2ADRL3									
		D2	D2ADRL2									
		D1	D2ADRL1									
		D0	D2ADRL0									

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
HSDMA Ch.2 high-order destination address setup register Note: D) Dual address mode S) Single address mode	0030114A (HW)	D15	D2MOD1	Ch.2 transfer mode	D2MOD[1:0]	Mode	0	R/W			
		D14	D2MOD0		11	Invalid	0				
					10	Block					
					01	Successive					
				00	Single						
		D13	D2IN1	D) Ch.2 destination address control S) Invalid	D2IN[1:0]	Inc/dec	0	R/W			
		D12	D2IN0		11	Inc.(no init)	0				
					10	Inc.(init)					
					01	Dec.(no init)					
				00	Fixed						
		D11	D2ADRH11	D) Ch.2 destination address[27:16] S) Invalid			0	R/W			
		D10	D2ADRH10				0				
		D9	D2ADRH9				0				
D8	D2ADRH8	0									
D7	D2ADRH7	0									
D6	D2ADRH6	0									
D5	D2ADRH5	0									
D4	D2ADRH4	0									
D3	D2ADRH3	0									
D2	D2ADRH2	0									
D1	D2ADRH1	0									
D0	D2ADRH0	0									
HSDMA Ch.2 enable register (pHS2_EN)	0030114C (HW)	D15–1	–	reserved	–	–	–	0 when being read.			
		D0	HS2_EN	Ch.2 enable	1 Enable	0 Disable	0	R/W			
HSDMA Ch.2 trigger flag register (pHS2_TF)	0030114E (HW)	D15–1	–	reserved	–	–	–	0 when being read.			
		D0	HS2_TF	Ch.2 trigger flag clear (writing) Ch.2 trigger flag status (reading)	1 Clear 1 Set	0 No operation 0 Cleared	0	R/W			
HSDMA Ch.3 transfer counter register (pHS3_CNT)	00301150 (HW)	D15	TC3_L7	Ch.3 transfer counter[7:0] (block transfer mode)			0	R/W			
		D14	TC3_L6				0				
		D13	TC3_L5				0				
		D12	TC3_L4	Ch.3 transfer counter[15:8] (single/successive transfer mode)			0				
		D11	TC3_L3				0				
		D10	TC3_L2				0				
		D9	TC3_L1				0				
		D8	TC3_L0	0							
		D7	BLKLEN37	Ch.3 block length (block transfer mode)						0	R/W
		D6	BLKLEN36							0	
		D5	BLKLEN35	0							
		D4	BLKLEN34	Ch.3 transfer counter[7:0] (single/successive transfer mode)						0	
		D3	BLKLEN33							0	
		D2	BLKLEN32							0	
D1	BLKLEN31	0									
D0	BLKLEN30	0									
HSDMA Ch.3 control register Note: D) Dual address mode S) Single address mode	00301152 (HW)	D15	DUALM3	Ch.3 address mode selection	1 Dual addr	0 Single addr		0		R/W	
		D14	D3DIR	D) Invalid	–	–		–			
				S) Ch.3 transfer direction control	1 Memory WR	0 Memory RD		0		R/W	
		D13–8	–	reserved	–	–		–		0 when being read.	
		D7	TC3_H7	Ch.3 transfer counter[15:8] (block transfer mode)				0		R/W	
		D6	TC3_H6					0			
		D5	TC3_H5	0							
		D4	TC3_H4	Ch.3 transfer counter[23:16] (single/successive transfer mode)			0				
		D3	TC3_H3				0				
		D2	TC3_H2				0				
D1	TC3_H1	0									
D0	TC3_H0	0									

0x301154–0x301158

High-Speed DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
HSDMA Ch.3 low-order source address setup register (pHS3_SADR) Note: D) Dual address mode S) Single address mode	00301154 (HW)	D15	S3ADRL15	D) Ch.3 source address[15:0]			0	R/W	
		D14	S3ADRL14	S) Ch.3 memory address[15:0]					
		D13	S3ADRL13						
		D12	S3ADRL12						
		D11	S3ADRL11						
		D10	S3ADRL10						
		D9	S3ADRL9						
		D8	S3ADRL8						
		D7	S3ADRL7						
		D6	S3ADRL6						
		D5	S3ADRL5						
		D4	S3ADRL4						
		D3	S3ADRL3						
		D2	S3ADRL2						
		D1	S3ADRL1						
		D0	S3ADRL0						
		HSDMA Ch.3 high-order source address setup register Note: D) Dual address mode S) Single address mode	00301156 (HW)	D15			–		
D14	DATSIZE3			Ch.3 transfer data size	1 Half word	0 Byte	0	R/W	
D13	S3IN1			D) Ch.3 source address control	S3IN[1:0]		0	R/W	
					Inc/dec				
					11	Inc.(no init)			
					10	Inc.(init)			
D12	S3IN0			S) Ch.3 memory address control	Inc/dec		0	R/W	
					Dec.(no init)				
					01	Dec.(no init)			
					00	Fixed			
D11	S3ADRH11			D) Ch.3 source address[27:16]				0	R/W
D10	S3ADRH10	S) Ch.3 memory address[27:16]							
D9	S3ADRH9								
D8	S3ADRH8								
D7	S3ADRH7								
D6	S3ADRH6								
D5	S3ADRH5								
D4	S3ADRH4								
D3	S3ADRH3								
D2	S3ADRH2								
D1	S3ADRH1								
D0	S3ADRH0								
HSDMA Ch.3 low-order destination address setup register (pHS3_DADR) Note: D) Dual address mode S) Single address mode	00301158 (HW)	D15	D3ADRL15	D) Ch.3 destination address[15:0]			0	R/W	
		D14	D3ADRL14	S) Invalid					
		D13	D3ADRL13						
		D12	D3ADRL12						
		D11	D3ADRL11						
		D10	D3ADRL10						
		D9	D3ADRL9						
		D8	D3ADRL8						
		D7	D3ADRL7						
		D6	D3ADRL6						
		D5	D3ADRL5						
		D4	D3ADRL4						
		D3	D3ADRL3						
		D2	D3ADRL2						
		D1	D3ADRL1						
		D0	D3ADRL0						

0x30115A–0x301164

High-Speed DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.3 high-order destination address setup register Note: D) Dual address mode S) Single address mode	0030115A (HW)	D15	D3MOD1	Ch.3 transfer mode	D3MOD[1:0]	Mode	0	R/W
		D14	D3MOD0		11	Invalid	0	
					10	Block		
					01	Successive		
				00	Single			
		D13	D3IN1	D) Ch.3 destination address control S) Invalid	D3IN[1:0]	Inc/dec	0	R/W
		D12	D3IN0		11	Inc.(no init)	0	
					10	Inc.(init)		
					01	Dec.(no init)		
				00	Fixed			
		D11	D3ADRH11	D) Ch.3 destination address[27:16] S) Invalid			0	R/W
		D10	D3ADRH10				0	
		D9	D3ADRH9				0	
D8	D3ADRH8	0						
D7	D3ADRH7	0						
D6	D3ADRH6	0						
D5	D3ADRH5	0						
D4	D3ADRH4	0						
D3	D3ADRH3	0						
D2	D3ADRH2	0						
D1	D3ADRH1	0						
D0	D3ADRH0	0						
HSDMA Ch.3 enable register (pHS3_EN)	0030115C (HW)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	HS3_EN	Ch.3 enable	1 Enable	0 Disable	0	R/W
HSDMA Ch.3 trigger flag register (pHS3_TF)	0030115E (HW)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	HS3_TF	Ch.3 trigger flag clear (writing) Ch.3 trigger flag status (reading)	1 Clear 1 Set	0 No operation 0 Cleared	0	R/W
HSDMA Ch.0 control register (pHS0_ADVMODE) for ADV mode Note: D) Dual mode S) Single mode	00301162 (HW)	D15–6	–	reserved	–	–	–	0 when being read.
		D5	DOID	D) Ch.0 destination address control S) Invalid	1 Decrement (with init.)	0 DOIN[1:0] setting	0	R/W
		D4	SOID	D) Ch.0 source address control S) Ch.0 memory address control	1 Decrement (with init.)	0 SOIN[1:0] setting	0	R/W
		D3–1	–	reserved	–	–	–	–
D0	WORDSIZE0	Ch.0 transfer data size	1 Word	0 DATSIZE0 setting	0	R/W		
HSDMA Ch.0 low-order source address setup register (pHS0_ADV_SADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301164 (HW)	D15	S0ADRL15	D) Ch.0 source address[15:0]			0	R/W
		D14	S0ADRL14	S) Ch.0 memory address[15:0]			0	
		D13	S0ADRL13	0				
		D12	S0ADRL12	0				
		D11	S0ADRL11	0				
		D10	S0ADRL10	0				
		D9	S0ADRL9	0				
		D8	S0ADRL8	0				
		D7	S0ADRL7	0				
		D6	S0ADRL6	0				
		D5	S0ADRL5	0				
		D4	S0ADRL4	0				
		D3	S0ADRL3	0				
D2	S0ADRL2	0						
D1	S0ADRL1	0						
D0	S0ADRL0	0						

0x301166–0x301172

High-Speed DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
HSDMA Ch.0 high-order source address setup register for ADV mode Note: D) Dual address mode S) Single address mode	00301166 (HW)	D15	S0ADRH15	D) Ch.0 source address[31:16]		0	R/W		
		D14	S0ADRH14	S) Ch.0 memory address[31:16]		0			
		D13	S0ADRH13			0			
		D12	S0ADRH12			0			
		D11	S0ADRH11			0			
		D10	S0ADRH10			0			
		D9	S0ADRH9			0			
		D8	S0ADRH8			0			
		D7	S0ADRH7			0			
		D6	S0ADRH6			0			
		D5	S0ADRH5			0			
		D4	S0ADRH4			0			
		D3	S0ADRH3			0			
		D2	S0ADRH2			0			
		D1	S0ADRH1			0			
		D0	S0ADRH0			0			
		HSDMA Ch.0 low-order destination address setup register (pHS0_ADV_DADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301168 (HW)	D15	D0ADRL15	D) Ch.0 destination address[15:0]		0	R/W
D14	D0ADRL14			S) Invalid		0			
D13	D0ADRL13					0			
D12	D0ADRL12					0			
D11	D0ADRL11					0			
D10	D0ADRL10					0			
D9	D0ADRL9					0			
D8	D0ADRL8					0			
D7	D0ADRL7					0			
D6	D0ADRL6					0			
D5	D0ADRL5					0			
D4	D0ADRL4					0			
D3	D0ADRL3					0			
D2	D0ADRL2					0			
D1	D0ADRL1					0			
D0	D0ADRL0					0			
HSDMA Ch.0 high-order destination address setup register for ADV mode Note: D) Dual address mode S) Single address mode	0030116A (HW)			D15	D0ADRH15	D) Ch.0 destination address[31:16]		0	R/W
		D14	D0ADRH14	S) Invalid		0			
		D13	D0ADRH13			0			
		D12	D0ADRH12			0			
		D11	D0ADRH11			0			
		D10	D0ADRH10			0			
		D9	D0ADRH9			0			
		D8	D0ADRH8			0			
		D7	D0ADRH7			0			
		D6	D0ADRH6			0			
		D5	D0ADRH5			0			
		D4	D0ADRH4			0			
		D3	D0ADRH3			0			
		D2	D0ADRH2			0			
		D1	D0ADRH1			0			
		D0	D0ADRH0			0			
		HSDMA Ch.1 control register (pHS1_ADVMODE) for ADV mode Note: D) Dual mode S) Single mode	00301172 (HW)	D15–6	–	reserved	–	–	–
D5	D11D			D) Ch.1 destination address control S) Invalid	1 Decrement (with init.)	0 D11N[1:0] setting	0	R/W	
D4	S11D			D) Ch.1 source address control S) Ch.1 memory address control	1 Decrement (with init.)	0 S11N[1:0] setting	0	R/W	
D3–1	–			reserved	–	–	–	–	0 when being read.
D0	WORDSIZE1			Ch.1 transfer data size	1 Word	0 DATSIZE1 setting	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.1 low-order source address setup register (pHS1_ADV_SADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301174 (HW)	D15	S1ADRL15	D) Ch.1 source address[15:0]		0	R/W	
		D14	S1ADRL14	S) Ch.1 memory address[15:0]				
		D13	S1ADRL13					
		D12	S1ADRL12					
		D11	S1ADRL11					
		D10	S1ADRL10					
		D9	S1ADRL9					
		D8	S1ADRL8					
		D7	S1ADRL7					
		D6	S1ADRL6					
		D5	S1ADRL5					
		D4	S1ADRL4					
		D3	S1ADRL3					
		D2	S1ADRL2					
		D1	S1ADRL1					
		D0	S1ADRL0					
HSDMA Ch.1 high-order source address setup register for ADV mode Note: D) Dual address mode S) Single address mode	00301176 (HW)	D15	S1ADRH15	D) Ch.1 source address[31:16]		0	R/W	
		D14	S1ADRH14	S) Ch.1 memory address[31:16]				
		D13	S1ADRH13					
		D12	S1ADRH12					
		D11	S1ADRH11					
		D10	S1ADRH10					
		D9	S1ADRH9					
		D8	S1ADRH8					
		D7	S1ADRH7					
		D6	S1ADRH6					
		D5	S1ADRH5					
		D4	S1ADRH4					
		D3	S1ADRH3					
		D2	S1ADRH2					
		D1	S1ADRH1					
		D0	S1ADRH0					
HSDMA Ch.1 low-order destination address setup register (pHS1_ADV_DADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301178 (HW)	D15	D1ADRL15	D) Ch.1 destination address[15:0]		0	R/W	
		D14	D1ADRL14	S) Invalid				
		D13	D1ADRL13					
		D12	D1ADRL12					
		D11	D1ADRL11					
		D10	D1ADRL10					
		D9	D1ADRL9					
		D8	D1ADRL8					
		D7	D1ADRL7					
		D6	D1ADRL6					
		D5	D1ADRL5					
		D4	D1ADRL4					
		D3	D1ADRL3					
		D2	D1ADRL2					
		D1	D1ADRL1					
		D0	D1ADRL0					
HSDMA Ch.1 high-order destination address setup register for ADV mode Note: D) Dual address mode S) Single address mode	0030117A (HW)	D15	D1ADRH15	D) Ch.1 destination address[31:16]		0	R/W	
		D14	D1ADRH14	S) Invalid				
		D13	D1ADRH13					
		D12	D1ADRH12					
		D11	D1ADRH11					
		D10	D1ADRH10					
		D9	D1ADRH9					
		D8	D1ADRH8					
		D7	D1ADRH7					
		D6	D1ADRH6					
		D5	D1ADRH5					
		D4	D1ADRH4					
		D3	D1ADRH3					
		D2	D1ADRH2					
		D1	D1ADRH1					
		D0	D1ADRH0					

0x301182–0x301188

High-Speed DMA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
HSDMA Ch.2 control register (pHS2_ADVMODE) for ADV mode Note: D) Dual mode S) Single mode	00301182 (HW)	D15–6	–	reserved	–	–	–	0 when being read.
		D5	D2ID	D) Ch.2 destination address control S) Invalid	1 Decrement (with init.)	0 D2IN[1:0] setting	0	R/W
		D4	S2ID	D) Ch.2 source address control S) Ch.2 memory address control	1 Decrement (with init.)	0 S2IN[1:0] setting	0	R/W
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	WORDSIZE2	Ch.2 transfer data size	1 Word	0 DATSIZE2 setting	0	R/W
HSDMA Ch.2 low-order source address setup register (pHS2_ADV_SADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301184 (HW)	D15	S2ADRL15	D) Ch.2 source address[15:0]			0	R/W
		D14	S2ADRL14	S) Ch.2 memory address[15:0]			0	
		D13	S2ADRL13				0	
		D12	S2ADRL12				0	
		D11	S2ADRL11				0	
		D10	S2ADRL10				0	
		D9	S2ADRL9				0	
		D8	S2ADRL8				0	
		D7	S2ADRL7				0	
		D6	S2ADRL6				0	
		D5	S2ADRL5				0	
		D4	S2ADRL4				0	
		D3	S2ADRL3				0	
		D2	S2ADRL2				0	
		D1	S2ADRL1				0	
		D0	S2ADRL0				0	
HSDMA Ch.2 high-order source address setup register for ADV mode Note: D) Dual address mode S) Single address mode	00301186 (HW)	D15	S2ADRH15	D) Ch.2 source address[31:16]			0	R/W
		D14	S2ADRH14	S) Ch.2 memory address[31:16]			0	
		D13	S2ADRH13				0	
		D12	S2ADRH12				0	
		D11	S2ADRH11				0	
		D10	S2ADRH10				0	
		D9	S2ADRH9				0	
		D8	S2ADRH8				0	
		D7	S2ADRH7				0	
		D6	S2ADRH6				0	
		D5	S2ADRH5				0	
		D4	S2ADRH4				0	
		D3	S2ADRH3				0	
		D2	S2ADRH2				0	
		D1	S2ADRH1				0	
		D0	S2ADRH0				0	
HSDMA Ch.2 low-order destination address setup register (pHS2_ADV_DADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301188 (HW)	D15	D2ADRL15	D) Ch.2 destination address[15:0]			0	R/W
		D14	D2ADRL14	S) Invalid			0	
		D13	D2ADRL13				0	
		D12	D2ADRL12				0	
		D11	D2ADRL11				0	
		D10	D2ADRL10				0	
		D9	D2ADRL9				0	
		D8	D2ADRL8				0	
		D7	D2ADRL7				0	
		D6	D2ADRL6				0	
		D5	D2ADRL5				0	
		D4	D2ADRL4				0	
		D3	D2ADRL3				0	
		D2	D2ADRL2				0	
		D1	D2ADRL1				0	
		D0	D2ADRL0				0	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
HSDMA Ch.2 high-order destination address setup register for ADV mode Note: D) Dual address mode S) Single address mode	0030118A (HW)	D15	D2ADRH15	D) Ch.2 destination address[31:16]		0	R/W		
		D14	D2ADRH14	S) Invalid		0			
		D13	D2ADRH13			0			
		D12	D2ADRH12			0			
		D11	D2ADRH11			0			
		D10	D2ADRH10			0			
		D9	D2ADRH9			0			
		D8	D2ADRH8			0			
		D7	D2ADRH7			0			
		D6	D2ADRH6			0			
		D5	D2ADRH5			0			
		D4	D2ADRH4			0			
		D3	D2ADRH3			0			
		D2	D2ADRH2			0			
		D1	D2ADRH1			0			
		D0	D2ADRH0			0			
		HSDMA Ch.3 control register (pHS3_ADV_MODE) for ADV mode Note: D) Dual mode S) Single mode	00301192 (HW)	D15–6	–	reserved	–	–	–
D5	D3ID			D) Ch.3 destination address control S) Invalid	1 Decrement (with init.)	0 D3IN[1:0] setting	0	R/W	
D4	S3ID			D) Ch.3 source address control S) Ch.3 memory address control	1 Decrement (with init.)	0 S3IN[1:0] setting	0	R/W	
D3–1	–			reserved	–	–	–	–	0 when being read.
D0	WORDSIZ3			Ch.3 transfer data size	1 Word	0 DATSIZE3 setting	0	R/W	
HSDMA Ch.3 low-order source address setup register (pHS3_ADV_SADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301194 (HW)	D15	S3ADRL15	D) Ch.3 source address[15:0]		0	R/W		
		D14	S3ADRL14	S) Ch.3 memory address[15:0]		0			
		D13	S3ADRL13			0			
		D12	S3ADRL12			0			
		D11	S3ADRL11			0			
		D10	S3ADRL10			0			
		D9	S3ADRL9			0			
		D8	S3ADRL8			0			
		D7	S3ADRL7			0			
		D6	S3ADRL6			0			
		D5	S3ADRL5			0			
		D4	S3ADRL4			0			
		D3	S3ADRL3			0			
D2	S3ADRL2			0					
D1	S3ADRL1			0					
D0	S3ADRL0			0					
HSDMA Ch.3 high-order source address setup register for ADV mode Note: D) Dual address mode S) Single address mode	00301196 (HW)	D15	S3ADRH15	D) Ch.3 source address[31:16]		0	R/W		
		D14	S3ADRH14	S) Ch.3 memory address[31:16]		0			
		D13	S3ADRH13			0			
		D12	S3ADRH12			0			
		D11	S3ADRH11			0			
		D10	S3ADRH10			0			
		D9	S3ADRH9			0			
		D8	S3ADRH8			0			
		D7	S3ADRH7			0			
		D6	S3ADRH6			0			
		D5	S3ADRH5			0			
		D4	S3ADRH4			0			
		D3	S3ADRH3			0			
D2	S3ADRH2			0					
D1	S3ADRH1			0					
D0	S3ADRH0			0					

0x301198–0x30119E

High-Speed DMA

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
HSDMA Ch.3 low-order destination address setup register (pHS3_ADV_DADR) for ADV mode Note: D) Dual address mode S) Single address mode	00301198 (HW)	D15	D3ADRL15	D) Ch.3 destination address[15:0] S) Invalid				0	R/W		
		D14	D3ADRL14								
		D13	D3ADRL13								
		D12	D3ADRL12								
		D11	D3ADRL11								
		D10	D3ADRL10								
		D9	D3ADRL9								
		D8	D3ADRL8								
		D7	D3ADRL7								
		D6	D3ADRL6								
		D5	D3ADRL5								
		D4	D3ADRL4								
		D3	D3ADRL3								
		D2	D3ADRL2								
		D1	D3ADRL1								
		D0	D3ADRL0								
		HSDMA Ch.3 high-order destination address setup register for ADV mode Note: D) Dual address mode S) Single address mode	0030119A (HW)					D15			D3ADRH15
D14	D3ADRH14										
D13	D3ADRH13										
D12	D3ADRH12										
D11	D3ADRH11										
D10	D3ADRH10										
D9	D3ADRH9										
D8	D3ADRH8										
D7	D3ADRH7										
D6	D3ADRH6										
D5	D3ADRH5										
D4	D3ADRH4										
D3	D3ADRH3										
D2	D3ADRH2										
D1	D3ADRH1										
D0	D3ADRH0										
HSDMA STD/ADV mode select register (pHS_CNTLMODE)	0030119C (HW)			D15–1	–	reserved	–			–	–
		D0	HSDMAADV	Standard mode/advanced mode select	1	Advanced mode	0	Standard mode	0	R/W	
DMA sequential access time register (pHS_ACCTIME)	0030119E (HW)	D15–4	–	reserved	–			–	–	0 when being read.	
		D3	DMAACCTIME3	IDMA and HSDMA sequential access time setup	0	Unlimited	8	512 cycles	0	R/W	
		D2	DMAACCTIME2								
		D1	DMAACCTIME1								
		D0	DMAACCTIME0								
					1	64 cycles	9	576 cycles	0		
					2	128 cycles	A	640 cycles	0		
					3	192 cycles	B	704 cycles	0		
		4	256 cycles		C	768 cycles	0				
		5	320 cycles	D	832 cycles	0					
		6	384 cycles	E	896 cycles	0					
		7	448 cycles	F	960 cycles	0					

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
BCLK and setup time control register (pSRAMC_BCLK_SETUP)	00301500 (W)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	CE9BCLK	#CE9 area BCLK divide control	1 SRAMC_CLK × 1/2 0 SRAMC_CLK × 1	1	R/W		
		D6	CE9HOLD2	#CE9 area output disable time	0 to 7	0	0	R/W	
		D5	CE9HOLD1					0	
		D4	CE9HOLD0					0	
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	CE11STUP	#CE11 setup time	1 No setup time 0 +1 BCLK	0	R/W		
D1	CE4STUP	#CE4 setup time	1 No setup time 0 +1 BCLK	0	R/W				
D0	BCLK	BCLK divide control	1 SRAMC_CLK × 1/2 0 SRAMC_CLK × 1	1	R/W				
Wait control register (pSRAMC_SWAIT)	00301504 (W)	D31	–	reserved	–	–	–	0 when being read.	
		D30	CE11WAIT2	Number of #CE11 static wait cycles	0 to 7	1	R/W		
		D29	CE11WAIT1				1		
		D28	CE11WAIT0				1		
		D27	CE10WAIT3	reserved	–	–	–	0 when being read.	
		D26	CE10WAIT2	Number of #CE10 static wait cycles	0 to 7	1	R/W		
		D25	CE10WAIT1				1		
		D24	CE10WAIT0				1		
		D23	CE9WAIT3	reserved	–	–	–	0 when being read.	
		D22	CE9WAIT2	Number of #CE9 static wait cycles	0 to 7	1	R/W		
		D21	CE9WAIT1				1		
		D20	CE9WAIT0				1		
		D19	CE8WAIT3	reserved	–	–	–	0 when being read.	
		D18	CE8WAIT2	Number of #CE8 static wait cycles	0 to 7	1	R/W		
		D17	CE8WAIT1				1		
		D16	CE8WAIT0				1		
		D15	CE7WAIT3	reserved	–	–	–	0 when being read.	
		D14	CE7WAIT2	Number of #CE7 static wait cycles	0 to 7	1	R/W		
		D13	CE7WAIT1				1		
		D12	CE7WAIT0				1		
		D11	CE6WAIT3	reserved	–	–	–	0 when being read.	
		D10	CE6WAIT2	Number of #CE6 static wait cycles	0 to 7	1	R/W		
		D9	CE6WAIT1				1		
		D8	CE6WAIT0				1		
D7	CE5WAIT3	reserved	–	–	–	0 when being read.			
D6	CE5WAIT2	Number of #CE5 static wait cycles	0 to 7	1	R/W				
D5	CE5WAIT1				1				
D4	CE5WAIT0				1				
D3	CE4WAIT3	reserved	–	–	–	0 when being read.			
D2	CE4WAIT2	Number of #CE4 static wait cycles	0 to 7	1	R/W				
D1	CE4WAIT1				1				
D0	CE4WAIT0				1				
Device size setup register (pSRAMC_SLV_SIZE)	00301508 (W)	D31–16	–	reserved	–	–	–	0 when being read.	
		D15	CE11SIZE1	#CE11 device size	(See below)	0	R/W		
		D14	CE11SIZE0				1		
		D13–12	–	reserved	–	–	–	0 when being read.	
		D11	CE9SIZE1	#CE9 device size	CE.xSIZE[1:0]	Size	0	R/W	
		D10	CE9SIZE0					11	reserved
		D9	CE8SIZE1	#CE8 device size	10	8 bits	0	R/W	
		D8	CE8SIZE0					1	
		D7	CE7SIZE1	#CE7 device size	01	16 bits	0	R/W	
		D6	CE7SIZE0					1	
		D5	CE6SIZE1	#CE6 device size	00	reserved	0	R/W	
		D4	CE6SIZE0					1	
		D3	CE5SIZE1	#CE5 device size	00	reserved	0	R/W	
D2	CE5SIZE0	1							
D1	CE4SIZE1	#CE4 device size	00	reserved	0	R/W			
D0	CE4SIZE0					1			
Device type setup register (pSRAMC_A0_BSL)	0030150C (W)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	CE11TYPE	#CE11 device type	1 BSL 0 A0	0	R/W		
		D6	CE10TYPE	#CE10 device type		0	R/W		
		D5	CE9TYPE	#CE9 device type		0	R/W		
		D4	CE8TYPE	#CE8 device type		0	R/W		
		D3	CE7TYPE	#CE7 device type		0	R/W		
		D2	CE6TYPE	#CE6 device type		0	R/W		
		D1	CE5TYPE	#CE5 device type		0	R/W		
		D0	CE4TYPE	#CE4 device type		0	R/W		
Area location setup register (pSRAMC_ALS)	00301510 (W)	D31–1	–	reserved	–	–	–	0 when being read.	
		D0	A6LOC	Area 6 location setup	1 External 0 Internal	0	R/W		

0x301600–0x301610

SDRAM Controller

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
SDRAM initial register (pSDRAMC_INI)	00301600 (W)	D31–5	–	reserved	–		–	–	0 when being read.	
		D4	SDON	SDRAM controller enable	1	Enabled	0 Disabled	0	R/W	
		D3	SDEN	SDRAM initialize flag	1	Initialized	0 Not initialized	0	R	
		D2	INIMRS	MRS command enable for init.	1	Enabled	0 Disabled	0	R/W	
		D1	INIPRE	PALL command enable for init.	1	Enabled	0 Disabled	0	R/W	
		D0	INIREF	REF command enable for init.	1	Enabled	0 Disabled	0	R/W	
SDRAM configuration register (pSDRAMC_CTL)	00301604 (W)	D31–14	–	reserved	–		–	–	0 when being read.	
		D13	T24NS1	Number of SDRAM tRP and tRCD cycles	T24NS[1:0] = 0 to 3 → 1 to 4 cycles		0	R/W		
		D12	T24NS0				0			
		D11	–	reserved	–		–	–		0 when being read.
		D10	T60NS2	Number of SDRAM tRAS cycles	T60NS[2:0] = 0 to 7 → 1 to 8 cycles		0	R/W		
		D9	T60NS1				0			
		D8	T60NS0				0			
		D7	T80NS3	Number of SDRAM tRC, tRFC and tXSR cycles	T80NS[3:0] = 0 to 15 → 1 to 16 cycles		1	R/W		
		D6	T80NS2		1					
		D5	T80NS1		1					
		D4	T80NS0		0					
D3	–	reserved	–		–	–	0 when being read.			
D2	ADDRC2	SDRAM address configuration	ADDRC[2:0]	Configuration	0	R/W				
D1	ADDRC1				111	32M x 16 bits x 1	0			
D0	ADDRC0				110	16M x 8 bits x 2	0			
					101	8M x 8 bits x 2				
			100	2M x 8 bits x 2						
			011	16M x 16 bits x 1						
			010	8M x 16 bits x 1						
			001	4M x 16 bits x 1						
			000	1M x 16 bits x 1						
SDRAM refresh register (pSDRAMC_REF)	00301608 (W)	D31–26	–	reserved	–		–	–	0 when being read.	
		D25	SELDO	SDRAM self-refresh status	1	Refresh mode	0 Done	0	R	
		D24	SCKON	SDRAM clock during self-refresh	1	Enabled	0 Disabled	0	R/W	
		D23	SELEN	SDRAM self-refresh enable	1	Enabled	0 Disabled	0	R/W	
		D22	SELCO6	SDRAM self-refresh counter	0x0 to 0x7F		1	R/W		
		D21	SELCO5		1					
		D20	SELCO4		1					
		D19	SELCO3		1					
		D18	SELCO2		1					
		D17	SELCO1		1					
		D16	SELCO0	1						
		D15–12	–	reserved	–		–	–	0 when being read.	
		D11	AURCO11	SDRAM auto-refresh counter	0x0 to 0xFFFF		0	R/W		
		D10	AURCO10		0					
		D9	AURCO9		0					
		D8	AURCO8		0					
		D7	AURCO7		1					
D6	AURCO6	0								
D5	AURCO5	0								
D4	AURCO4	0								
D3	AURCO3	1								
D2	AURCO2	1								
D1	AURCO1	0								
D0	AURCO0	0								
SDRAM application configuration register (pSDRAMC_APP)	00301610 (W)	D31	ARBON	Arbiter enable	1	Enabled	0 Disabled	0	R/W	
		D30–6	–	reserved	–		–	–	0 when being read.	
		D5	DBF	Double frequency mode enable	1	Enabled	0 Disabled	0	R/W	
		D4	INCR	INCR transfer enable	1	Enabled	0 Disabled	0	R/W	
		D3	CAS1	CAS latency setup	CAS[1:0]		1	R/W		
		D2	CAS0		11	3	0			
					10	2				
			01	1						
			00	reserved						
D1	APPON	SDAPP control	1	On	0 Off	0	R/W			
D0	IQB	Instruction queue buffer enable	1	Enabled	0 Disabled	0	R/W			

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
SPI receive data register (pSPI_RXD)	00301700 (W)	D31 D0	SPIRXD31 SPIRXD0	SPI receive data SPIRXD31 = MSB SPIRXD0 = LSB	0x0 to 0xFFFFFFFF	0x0	R				
SPI transmit data register (pSPI_TXD)	00301704 (W)	D31 D0	SPITXD31 SPITXD0	SPI transmit data SPITXD31 = MSB SPITXD0 = LSB	0x0 to 0xFFFFFFFF	0x0	R/W				
SPI control register 1 (pSPI_CTL1)	00301708 (W)	D31–15	–	reserved	–	–	–	0 when being read.			
		D14	BPT4	Number of data bits per transfer	Number of data bits per transfer = BPT + 1	0	R/W				
		D13	BPT3			0					
		D12	BPT2			0					
		D11	BPT1			0					
		D10	BPT0			0					
		D9	CPHA	SPI_CLK phase selection	1	Phase 1	0	Phase 0	0	R/W	
		D8	CPOL	SPI_CLK polarity selection	1	Active low	0	Active high	0	R/W	
		D7	MWEN	reserved	–	Fix at 0.	0	–	0	–	
		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	TXDE	Transmit DMA enable	1	Enabled	0	Disabled	0	R/W	
D2	RXDE	Receive DMA enable	1	Enabled	0	Disabled	0	R/W			
D1	MODE	SPI mode selection	1	Master	0	Slave	0	R/W			
D0	ENA	SPI enable	1	Enabled	0	Disabled	0	R/W			
SPI control register 2 (pSPI_CTL2)	0030170C (W)	D31–12	–	reserved	–	–	–	0 when being read.			
		D11	SSA	reserved	–	Fix at 0.	0	–			
		D10	SS	Slave select control	–		0	R/W	Master mode		
					1	SPI select	0	SPI deselect		0	Slave mode
		D9	SSP	reserved	–	Fix at 0.	0	–			
		D8	SSC	reserved	–	Fix at 0.	0	–			
		D7–3	–	reserved	–	–	–	–	0 when being read.		
		D2	RDYP	reserved	–	Fix at 0.	0	–			
		D1	RDYS	reserved	–	Fix at 0.	0	–			
		D0	RDYE	reserved	–	Fix at 0.	0	–			
SPI wait register (pSPI_WAIT)	00301710 (W)	D31 D0	SPIW31 SPIW0	Wait cycle control SPIW31 = MSB SPIW0 = LSB	Number of wait cycles = SPIW[31:0] + 1 (1 to 65536)	0x0	R/W				
SPI status register (pSPI_STAT)	00301714 (W)	D31–7	–	reserved	–	–	–	0 when being read.			
		D6	BSYF	Transfer busy flag	1	Busy	0	Idle	0	R	Master mode
		D5	MFEF	reserved	–	–	–	–	0 when being read.		
		D4	TDEF	Transmit data empty flag	1	Empty	0	Not empty	1	R	
		D3	RDOF	Receive data overflow flag	1	Occurred	0	Not occurred	0	R	
		D2	RDFF	Receive data full flag	1	Full	0	Not full	0	R	
D1–0	–	reserved	–	–	–	–	0 when being read.				
SPI interrupt control register (pSPI_INT)	00301718 (W)	D31–6	–	reserved	–	–	–	0 when being read.			
		D5	MFIE	reserved	–	Fix at 0.	0	–			
		D4	TEIE	Transmit data empty int. enable	1	Enabled	0	Disabled	0	R/W	
		D3	ROIE	Receive overflow interrupt enable	1	Enabled	0	Disabled	0	R/W	
		D2	RFIE	Receive data full 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	
SPI receive data mask register (pSPI_RXMK)	0030171C (W)	D31–15	–	reserved	–	–	–	0 when being read.			
		D14	RXMASK4	Bit mask for reading received data	0x0 to 0x1F	0	R/W				
		D13	RXMASK3			0					
		D12	RXMASK2			0					
		D11	RXMASK1			0					
		D10	RXMASK0			0					
		D9–2	–	reserved	–	–	–	0 when being read.			
		D1	RXME	Receive data mask enable	1	Enabled	0	Disabled	0	R/W	
D0	–	reserved	–	–	–	–	Do not write 1.				

0x301800–0x30181C

DCSIO

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
DCSIO control register (pDCSIO_CTL)	00301800 (W)	D31–16	–	reserved	–	–	–	0 when being read.	
		D15	DIVRT7	DCSIO system clock baud rate setup (DIVRT must be 2 or more.)	Baud rate = $f_{MCLK}/(DIVRT \times 2)$	0	R/W		
		D14	DIVRT6			0			
		D13	DIVRT5			0			
		D12	DIVRT4			0			
		D11	DIVRT3			0			
		D10	DIVRT2			0			
		D9	DIVRT1			0			
		D8	DIVRT0			0			
		D7–4	–	reserved	–	–	–	0 when being read.	
D3	ADVRATE	Non-base line transfer rate setup	1	Base \times 1/8	0 = Base line	0	R/W		
D2	BASESEL	Base line selection	1	A	0 B	0	R/W		
D1	LSBFIRST	LSB first selection	1	LSB first	0 MSB first	0	R/W		
D0	DCSIOEN	DCSIO enable	1	Enabled	0 Disabled	0	R/W		
DCSIO data load register (pDCSIO_LOAD)	00301804 (W)	D31–24	–	reserved	–	–	–	0 when being read.	
		D23	TXDA7	Line A transmit data	0x0 to 0xFF	1	R/W		
		D22	TXDA6			1			
		D21	TXDA5			1			
		D20	TXDA4			1			
		D19	TXDA3			1			
		D18	TXDA2			1			
		D17	TXDA1			1			
		D16	TXDA0			1			
		D15–8	–	reserved	–	–	–	0 when being read.	
		D7	TXDB7	Line B transmit data	0x0 to 0xFF	1	R/W		
		D6	TXDB6			1			
		D5	TXDB5			1			
		D4	TXDB4			1			
		D3	TXDB3			1			
		D2	TXDB2			1			
		D1	TXDB1			1			
D0	TXDB0	1							
DCSIO receive data register (pDCSIO_RCV)	00301808 (W)	D31–24	–	reserved	–	–	–	0 when being read.	
		D23	RXDA7	Line A receive data	0x0 to 0xFF	1	R		
		D22	RXDA6			1			
		D21	RXDA5			1			
		D20	RXDA4			1			
		D19	RXDA3			1			
		D18	RXDA2			1			
		D17	RXDA1			1			
		D16	RXDA0			1			
		D15–8	–	reserved	–	–	–	0 when being read.	
		D7	RXDB7	Line B receive data	0x0 to 0xFF	1	R		
		D6	RXDB6			1			
		D5	RXDB5			1			
		D4	RXDB4			1			
		D3	RXDB3			1			
		D2	RXDB2			1			
		D1	RXDB1			1			
D0	RXDB0	1							
DCSIO interrupt control register (pDCSIO_INT)	00301814 (W)	D31–3	–	reserved	–	–	–	0 when being read.	
		D2	RXBFFEN	Receive data full interrupt enable	1	Enabled	0 Disabled	0	R/W
		D1	TXBEEN	Transmit data empty int. enable	1	Enabled	0 Disabled	0	R/W
		D0	INTEN	DCSIO interrupt request enable	1	Enabled	0 Disabled	0	R/W
DCSIO status register (pDCSIO_STAT)	00301818 (W)	D31–3	–	reserved	–	–	–	0 when being read.	
		D2	RXBF	Receive data full flag	1	Full	0 Not full	0	R
		D1	TXBE	Transmit data empty flag	1	Empty	0 Not empty	0	R
		D0	BUSY	DCSIO busy flag	1	Busy	0 Idle	0	R
DCSIO port direction control register (pDCSIO_DIR)	0030181C (W)	D31–4	–	reserved	–	–	–	0 when being read.	
		D3	DIRA1	Line A direction select	DIRA[1:0]	I/O		0	R/W
						1*	Input only	0	
		D2	DIRA0			01	Input/push-pull output		
						00	Input/open-drain output		
		D1	DIRB1	Line B direction select	DIRB[1:0]	I/O		0	R/W
1*	Input only					0			
D0	DIRB0			01	Input/push-pull output				
				00	Input/open-drain output				

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC interrupt status register (pRTCINTSTAT)	00301900 (W)	D31–1	–	reserved	–	–	–	0 when being read.	
		D0	RTCI RQ	Interrupt status	1 Occurred 0 Not occurred	X	R/W	Reset by writing 1.	
RTC interrupt mode register (pRTCINTMODE)	00301904 (W)	D31–4	–	reserved	–	–	–	0 when being read.	
		D3	RTCT1	RTC interrupt cycle setup	RTCT[1:0]	Cycle	X	R/W	
		D2	RTCT0		11	1 hour	X		
					10	1 minute			
					01	1 second			
		00	1/64 second						
		D1	RTCI MD	RTC interrupt mode select	1 Level sense 0 Edge trigger	X	R/W		
		D0	RTCI EN	RTC interrupt enable	1 Enabled 0 Disabled	X	R/W		
RTC control register (pRTC_CNTL0)	00301908 (W)	D31–5	–	reserved	–	–	–	0 when being read.	
		D4	RTC24H	24H/12H mode select	1 24H 0 12H	X	R/W		
		D3	–	reserved	–	–	–		
		D2	RTCADJ	30-second adjustment	1 Adjust 0 –	X	R/W		
		D1	RTCSTP	Counter run/stop control	1 Stop 0 Run	X	R/W		
		D0	RTC RST	Software reset	1 Reset 0 –	X	R/W		
RTC access control register (pRTC_CNTL1)	0030190C (W)	D31–2	–	reserved	–	–	–	0 when being read.	
		D1	RTCBSY	Counter busy flag	1 Busy 0 R/W possible	X	R		
		D0	RTC HLD	Counter hold control	1 Hold 0 Running	X	R/W		
RTC second register (pRTCSEC)	00301910 (W)	D31–7	–	reserved	–	–	–	0 when being read.	
		D6	RTC SH2	RTC 10-second counter	0 to 5	X	R/W		
		D5	RTC SH1			X			
		D4	RTC SH0			X			
		D3	RTC SL3	RTC 1-second counter	0 to 9	X	R/W		
		D2	RTC SL2			X			
		D1	RTC SL1			X			
D0	RTC SL0	X							
RTC minute register (pRTCMIN)	00301914 (W)	D31–7	–	reserved	–	–	–	0 when being read.	
		D6	RTC MIH2	RTC 10-minute counter	0 to 5	X	R/W		
		D5	RTC MIH1			X			
		D4	RTC MIH0			X			
		D3	RTC MIL3	RTC 1-minute counter	0 to 9	X	R/W		
		D2	RTC MIL2			X			
		D1	RTC MIL1			X			
D0	RTC MIL0	X							
RTC hour register (pRTC HOUR)	00301918 (W)	D31–7	–	reserved	–	–	–	0 when being read.	
		D6	RTC AP	AM/PM indicator	1 PM 0 AM	X	R/W		
		D5	RTC HH1	RTC 10-hour counter	0 to 2 or 0 to 1	X	R/W		
		D4	RTC HH0			X			
		D3	RTC HL3	RTC 1-hour counter	0 to 9	X	R/W		
		D2	RTC HL2			X			
		D1	RTC HL1			X			
D0	RTC HL0	X							
RTC day register (pRTC DAY)	0030191C (W)	D31–6	–	reserved	–	–	–	0 when being read.	
		D5	RTC DH1	RTC 10-day counter	0 to 3	X	R/W		
		D4	RTC DH0			X			
		D3	RTC DL3	RTC 1-day counter	0 to 9	X	R/W		
		D2	RTC DL2			X			
D1	RTC DL1	X							
D0	RTC DL0	X							
RTC month register (pRTC MONTH)	00301920 (W)	D31–5	–	reserved	–	–	–	0 when being read.	
		D4	RTC MOH	RTC 10-month counter	0 or 1	X	R/W		
		D3	RTC MOL3	RTC 1-month counter	0 to 9	X	R/W		
		D2	RTC MOL2			X			
		D1	RTC MOL1			X			
		D0	RTC MOL0			X			
RTC year register (pRTC YEAR)	00301924 (W)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	RTC YH3	RTC 10-year counter	0 to 9	X	R/W		
		D6	RTC YH2			X			
		D5	RTC YH1			X			
		D4	RTC YH0			X			
		D3	RTC YL3	RTC 1-year counter	0 to 9	X	R/W		
		D2	RTC YL2			X			
D1	RTC YL1	X							
D0	RTC YL0	X							

0x301928**Real Time Clock**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC days of week register (pRTCDAWEEK)	00301928 (W)	D31-3	--	reserved		--	--	0 when being read.	
		D2	RTCWK2	RTC days of week counter	RTCWK[2:0]	Days of week	X	R/W	
		D1	RTCWK1		111	--	X		
		D0	RTCWK0		110	Saturday	X		
					101	Friday			
					100	Thursday			
					011	Wednesday			
					010	Tuesday			
					001	Monday			
					000	Sunday			

0x301A00–0x301A20

LCD Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Frame interrupt register (pLDCDC_INT)	00301A00 (W)	D31–1	–	reserved	–	–	–	0 when being read.	
		D0	INTEN	Frame interrupt enable	1 Enabled 0 Disabled	0	R/W		
Status and power save configuration register (pLDCDC_PS)	00301A04 (W)	D31	INTF	Frame interrupt flag	1 Generated 0 Not generated	0	R/W	Reset by writing 1.	
		D30–8	–	reserved	–	–	–	0 when being read.	
		D7	VNDPF	Vertical display status	1 VNDP 0 VDP	1	R		
		D6–2	–	reserved	–	–	–	–	0 when being read.
		D1	PSAVE1	Power save mode	PSAVE[1:0] Mode 1 1 Normal operation 1 0 Doze mode 0 1 reserved 0 0 Power save mode	0	R/W		
D0	PSAVE0	0							
Horizontal display register (pLDCDC_HD)	00301A10 (W)	D31–23	–	reserved	–	–	–	0 when being read.	
		D22	HTCNT6	Horizontal total period (HT) setup HT = HDP + HNDP	HT = (HTCNT + 1) × 8 [Ts]	0	R/W		
		D21	HTCNT5			0			
		D20	HTCNT4	HT > HDPS + HDP (for HR-TFT)	HNDP = (HTCNT - HDPCNT) × 8 [Ts]	0			
		D19	HTCNT3			0			
		D18	HTCNT2	0					
		D17	HTCNT1	0					
		D16	HTCNT0	0					
		D15–7	–	reserved	–	–	–	0 when being read.	
		D6	HDPCNT6	Horizontal display period (HDP) setup	HDP = (HDPCNT + 1) × 8 [Ts]	0	R/W		
		D5	HDPCNT5			0			
		D4	HDPCNT4	0					
		D3	HDPCNT3	0					
		D2	HDPCNT2	0					
		D1	HDPCNT1	0					
D0	HDPCNT0	0							
Vertical display register (pLDCDC_VD)	00301A14 (W)	D31–26	–	reserved	–	–	–	0 when being read.	
		D25	VTCNT9	Vertical total period (VT) setup VT = VDP + VNDP	VT = VTCNT + 1 [lines]	0	R/W		
		D24	VTCNT8			0			
		D23	VTCNT7	VT > VDPS + VDP (for HR-TFT)	VNDP = HTCNT - HDPCNT [lines]	0			
		D22	VTCNT6			0			
		D21	VTCNT5	0					
		D20	VTCNT4	0					
		D19	VTCNT3	0					
		D18	VTCNT2	0					
		D17	VTCNT1	0					
		D16	VTCNT0	0					
		D15–10	–	reserved	–	–	–	0 when being read.	
		D9	VDPCNT9	Vertical display period (VDP) setup	VDP = VDPCNT + 1 [lines]	0	R/W		
		D8	VDPCNT8			0			
		D7	VDPCNT7	0					
D6	VDPCNT6	0							
D5	VDPCNT5	0							
D4	VDPCNT4	0							
D3	VDPCNT3	0							
D2	VDPCNT2	0							
D1	VDPCNT1	0							
D0	VDPCNT0	0							
MOD rate register (pLDCDC_MR)	00301A18 (W)	D31–6	–	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			
Horizontal display start position register (pLDCDC_HDPS)	00301A20 (W)	D31–10	–	reserved	–	–	–	0 when being read.	
		D9	HDPSCNT9	Horizontal display period start position for HR-TFT HT > HDPS + HDP	HDPS = HDPSCNT + 1 [pixels]	0	R/W		
		D8	HDPSCNT8			0			
		D7	HDPSCNT7	0					
		D6	HDPSCNT6	0					
		D5	HDPSCNT5	0					
		D4	HDPSCNT4	0					
		D3	HDPSCNT3	0					
		D2	HDPSCNT2	0					
		D1	HDPSCNT1	0					
D0	HDPSCNT0	0							

0x301A24–0x301A2C

LCD Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vertical display start position register (pLCDC_VDPS)	00301A24 (W)	D31–10	–	reserved	–	–	–	0 when being read.
		D9	VDPSCNT9	Vertical display period start position for HR-TFT VT > VDPS + VDP	VDPS = VDPSCNT [lines]	0	R/W	0x0 must be set for STN panels.
		D8	VDPSCNT8			0		
		D7	VDPSCNT7			0		
		D6	VDPSCNT6			0		
		D5	VDPSCNT5			0		
		D4	VDPSCNT4			0		
		D3	VDPSCNT3			0		
		D2	VDPSCNT2			0		
		D1	VDPSCNT1			0		
		D0	VDPSCNT0			0		
FPLINE pulse setup register (pLCDC_L)	00301A28 (W)	D31–26	–	reserved	–	–	–	0 when being read.
		D25	FPLST9	FPLINE pulse start position	Start position = FPLST + 1 [pixels]	0	R/W	*1: For HR-TFT 0x0 must be set for STN panels.
		D24	FPLST8			0		
		D23	FPLST7			0		
		D22	FPLST6			0		
		D21	FPLST5			0		
		D20	FPLST4			0		
		D19	FPLST3			0		
		D18	FPLST2			0		
		D17	FPLST1			0		
		D16	FPLST0			0		
		D15–8	–	reserved	–	–	–	0 when being read.
		D7	FPLPOL	FPLINE pulse polarity	1 Active high 0 Active low	0	R/W	(*1)
		D6	FPLWD6	FPLINE pulse width	Pulse width = FPLWD + 1 [pixels]	0	R/W	
D5	FPLWD5	0						
D4	FPLWD4	0						
D3	FPLWD3	0						
D2	FPLWD2	0						
D1	FPLWD1	0						
D0	FPLWD0	0						
FPFRAME pulse setup register (pLCDC_F)	00301A2C (W)	D31–26	–	reserved	–	–	–	0 when being read.
		D25	FPFST9	FPFRAME pulse start position	Start position = FPFST × HT [pixels]	0	R/W	*1: For HR-TFT 0x0 must be set for STN panels.
		D24	FPFST8			0		
		D23	FPFST7			0		
		D22	FPFST6			0		
		D21	FPFST5			0		
		D20	FPFST4			0		
		D19	FPFST3			0		
		D18	FPFST2			0		
		D17	FPFST1			0		
		D16	FPFST0			0		
D15–8	–	reserved	–	–	–	0 when being read.		
D7	FPFPOL	FPFRAME pulse polarity	1 Active high 0 Active low	0	R/W	(*1)		
D6–3	–	reserved	–	–	–	0 when being read.		
D2	FPFWD2	FPFRAME pulse width	Pulse width = (FPFWD+1) × HT [pixels]	0	R/W	(*1)		
D1	FPFWD1			0				
D0	FPFWD0			0				

0x301A30–0x301A48

LCD Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPFRAME pulse offset register (pLDCD_F0)	00301A30 (W)	D31–26	–	reserved	–	–	–	0 when being read.	
		D25	FPFSTPO9	FPFRAME pulse stop offset	Stop offset = FPFSTPO [pixels]	0	R/W	*1: For HR-TFT 0x0 must be set for STN panels.	
		D24	FPFSTPO8			0			
		D23	FPFSTPO7			0			
		D22	FPFSTPO6			0			
		D21	FPFSTPO5			0			
		D20	FPFSTPO4			0			
		D19	FPFSTPO3			0			
		D18	FPFSTPO2			0			
		D17	FPFSTPO1			0			
		D16	FPFSTPO0			0			
		D15–10	–	reserved	–	–	–	–	0 when being read.
		D9	FPFSTO9	FPFRAME pulse start offset	Start offset = FPFSTO [pixels]	0	R/W	(*1)	
		D8	FPFSTO8			0			
		D7	FPFSTO7			0			
		D6	FPFSTO6			0			
		D5	FPFSTO5			0			
		D4	FPFSTO4			0			
		D3	FPFSTO3			0			
D2	FPFSTO2			0					
D1	FPFSTO1			0					
D0	FPFSTO0			0					
HR-TFT special output register (pLDCD_TSO)	00301A40 (W)	D31–4	–	reserved	–	–	–	0 when being read.	
		D3	CTL1CTL	TFT_CTL1 control	1 Program 0 Toggle/line	0	R/W	For HR-TFT 0x0 must be set for STN panels.	
		D2	PRESET	TFT_CTL0–2 preset enable	1 Program 0 Preset	0	R/W		
		D1	FPSPOL	FPSHIFT polarity	1 Falling 0 Rising	0	R/W		
		D0	CTLSWAP	TFT_CTL0/TFT_CTL1 swap	1 Swap 0 Not swap	0	R/W		
TFT_CTL1 pulse register (pLDCD_TC1)	00301A44 (W)	D31–26	–	reserved	–	–	–	0 when being read.	
		D25	CTL1STP9	TFT_CTL1 pulse stop offset	Stop offset = CTL1STP + 1 [pixels]	0	R/W	*2: For HR-TFT This register is enabled when PRESET = 1.	
		D24	CTL1STP8			0			
		D23	CTL1STP7	TFT_CTL1 pulse width = (CTL1STP - CTL1ST + 1) Ts		0			
		D22	CTL1STP6			0			
		D21	CTL1STP5			0			
		D20	CTL1STP4			0			
		D19	CTL1STP3			0			
		D18	CTL1STP2			0			
		D17	CTL1STP1			0			
		D16	CTL1STP0			0			
		D15–10	–	reserved	–	–	–	–	0 when being read.
		D9	CTL1ST9	TFT_CTL1 pulse start offset	Start offset = CTL1ST [pixels]	0	R/W	(*2)	
		D8	CTL1ST8			0			
		D7	CTL1ST7			0			
		D6	CTL1ST6			0			
		D5	CTL1ST5			0			
		D4	CTL1ST4			0			
		D3	CTL1ST3			0			
D2	CTL1ST2			0					
D1	CTL1ST1			0					
D0	CTL1ST0			0					
TFT_CTL0 pulse register (pLDCD_TC0)	00301A48 (W)	D31–26	–	reserved	–	–	–	0 when being read.	
		D25	CTL0STP9	TFT_CTL0 pulse stop offset	Stop offset = CTL0STP + 1 [pixels]	0	R/W	*2: For HR-TFT This register is enabled when PRESET = 1.	
		D24	CTL0STP8			0			
		D23	CTL0STP7	TFT_CTL0 pulse width = (CTL0STP - CTL0ST + 1) Ts		0			
		D22	CTL0STP6			0			
		D21	CTL0STP5			0			
		D20	CTL0STP4			0			
		D19	CTL0STP3			0			
		D18	CTL0STP2			0			
		D17	CTL0STP1			0			
		D16	CTL0STP0			0			
		D15–10	–	reserved	–	–	–	–	0 when being read.
		D9	CTL0ST9	TFT_CTL0 pulse start offset	Start offset = CTL0ST [pixels]	0	R/W	(*2)	
		D8	CTL0ST8			0			
		D7	CTL0ST7			0			
		D6	CTL0ST6			0			
		D5	CTL0ST5			0			
		D4	CTL0ST4			0			
		D3	CTL0ST3			0			
D2	CTL0ST2			0					
D1	CTL0ST1			0					
D0	CTL0ST0			0					

0x301A4C–0x301A80

LCD Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
TFT_CTL2 register (pLCDC_TC2)	00301A4C (W)	D31–10	–	reserved	–	–	–	0 when being read.	
		D9	CTL2DLY9	TFT_CTL2 delay	Delay = CTL2DLY [pixels]	0	R/W	For HR-TFT This register is enabled when PRESET = 1.	
		D8	CTL2DLY8			0			
		D7	CTL2DLY7			0			
		D6	CTL2DLY6			0			
		D5	CTL2DLY5			0			
		D4	CTL2DLY4			0			
		D3	CTL2DLY3			0			
		D2	CTL2DLY2			0			
		D1	CTL2DLY1			0			
D0	CTL2DLY0			0					
LCDC display mode register (pLCDC_DMD)	00301A60 (W)	D31	TFTSEL	HR-TFT panel selection	1 HR-TFT 0 STN	0	R/W		
		D30	COLOR	Color/mono selection	1 Color 0 Mono	0	R/W		
		D29	FPSMASK	FPSHIFT mask enable	1 Enabled 0 Disabled	0	R/W		
		D28	–	reserved	–	–	–		0 when being read.
		D27	DWD1	LCD panel data width	DWD[1:0]	Data format	0		R/W
		D26	DWD0		11	8-bit (format2)	0		
					10	reserved			
					01	8-bit (format1)			
				00	4-bit				
		D25	SWINV	Software video invert	1 Inverted 0 Normal	0	R/W		
		D24	BLANK	Display blank enable	1 Blank 0 Normal	0	R/W		
		D23–8	–	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.			
D2–0	BPP2 BPP1 BPP0	Bit-per-pixel select	BPP[2:0]	bpp (color/gray)	0	R/W			
			101	16 bpp (64Kc)	0				
			100	12 bpp (4Kc)	0				
			011	8 bpp (256c)					
			010	4 bpp (16c/16gr)					
			001	2 bpp (4c/4gr)					
			000	1 bpp (2c/2gr)					
			Other	reserved					
IRAM select register (pLCDC_IRAM)	00301A64 (W)	D31–1	–	reserved	–	–	0 when being read.		
		D0	IRAM	IRAM assignment	1 A0RAM 0 IVRAM	0	R/W		
Main window display start address register (pLCDC_MADD)	00301A70 (W)	D31	MWADR31	Main window start address MWADR31 = MSB MWADR0 = LSB	0x0 to 0xFFFFFFFFC	0x0	R/W		
		D30	MWADR30						
		D1	MWADR1						
		D0	MWADR0						
Main window line address offset register (pLCDC_MLADD)	00301A74 (W)	D31–10	–	reserved	–	–	0 when being read.		
		D9	MWLADR9	Main window line address offset Main window width (pixels) × BPP/32	0	R/W			
		D8	MWLADR8						
		D7	MWLADR7						
		D6	MWLADR6						
		D5	MWLADR5						
		D4	MWLADR4						
		D3	MWLADR3						
		D2	MWLADR2						
		D1	MWLADR1						
D0	MWLADR0								
Sub-window display start address register (pLCDC_SADD)	00301A80 (W)	D31	SWADR31	Sub-window start address SWADR31 = MSB SWADR0 = LSB	0x0 to 0xFFFFFFFFC	0x0	R/W		
		D30	SWADR30						
		D1	SWADR1						
		D0	SWADR0						

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Sub-window start position register (pLCDC_SSP)	00301A88 (W)	D31	PIPEN	PIP sub-window enable	1 Enabled 0 Disabled	0	R/W	
		D30–26	–	reserved	–	–	–	0 when being read.
		D25	PIPYST9	Sub-window vertical (Y) start position	Y start position = PIPYST (lines) from the origin	0	R/W	*3: This register is enabled when PIPEN = 1.
		D24	PIPYST8			0		
		D23	PIPYST7			0		
		D22	PIPYST6			0		
		D21	PIPYST5			0		
		D20	PIPYST4			0		
		D19	PIPYST3			0		
		D18	PIPYST2			0		
		D17	PIPYST1			0		
		D16	PIPYST0	0				
		D15–10	–	reserved	–	–	–	0 when being read.
		D9	PIPXST9	Sub-window horizontal (X) start position	X start position = PIPXST (pixels) from the origin (word units)	0	R/W	(*3)
		D8	PIPXST8			0		
		D7	PIPXST7			0		
		D6	PIPXST6			0		
		D5	PIPXST5			0		
		D4	PIPXST4			0		
		D3	PIPXST3			0		
D2	PIPXST2	0						
D1	PIPXST1	0						
D0	PIPXST0	0						
Sub-window end position register (pLCDC_SEP)	00301A8C (W)	D31–26	–	reserved	–	–	–	0 when being read.
		D25	PIPYEND9	Sub-window vertical (Y) end position	Y end position = PIPYEND (lines) from the origin	0	R/W	*3: This register is enabled when PIPEN = 1.
		D24	PIPYEND8			0		
		D23	PIPYEND7			0		
		D22	PIPYEND6			0		
		D21	PIPYEND5			0		
		D20	PIPYEND4			0		
		D19	PIPYEND3			0		
		D18	PIPYEND2			0		
		D17	PIPYEND1			0		
		D16	PIPYEND0	0				
		D15–10	–	reserved	–	–	–	0 when being read.
		D9	PIPXEND9	Sub-window horizontal (X) end position	X end position = PIPXEND (pixels) from the origin (word units)	0	R/W	(*3)
		D8	PIPXEND8			0		
		D7	PIPXEND7			0		
		D6	PIPXEND6			0		
		D5	PIPXEND5			0		
		D4	PIPXEND4			0		
		D3	PIPXEND3			0		
		D2	PIPXEND2			0		
D1	PIPXEND1	0						
D0	PIPXEND0	0						

0x301AA0–0x301AA4

LCD Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Look-up table data register 0 (pLCDC_LUT_03)	00301AA0 (W)	D31	LUT35	Look-up table entry 3 data	0x0 to 0x3F	0	R/W	
		D30	LUT34					
		D29	LUT33					
		D28	LUT32					
		D27	LUT31					
		D26	LUT30					
		D25–24	–			reserved		
		D23	LUT25	Look-up table entry 2 data	0x0 to 0x3F	0	R/W	
		D22	LUT24					
		D21	LUT23					
		D20	LUT22					
		D19	LUT21					
		D18	LUT20					
		D17–16	–	reserved	–	–	–	0 when being read.
		D15	LUT15	Look-up table entry 1 data	0x0 to 0x3F	0	R/W	
		D14	LUT14					
		D13	LUT13					
D12	LUT12							
D11	LUT11							
D10	LUT10							
D9–8	–	reserved	–	–	–	0 when being read.		
D7	LUT05	Look-up table entry 0 data	0x0 to 0x3F	0	R/W			
D6	LUT04							
D5	LUT03							
D4	LUT02							
D3	LUT01							
D2	LUT00							
D1–0	–	reserved	–	–	–	0 when being read.		
Look-up table data register 1 (pLCDC_LUT_47)	00301AA4 (W)	D31	LUT75	Look-up table entry 7 data	0x0 to 0x3F	0	R/W	
		D30	LUT74					
		D29	LUT73					
		D28	LUT72					
		D27	LUT71					
		D26	LUT70					
		D25–24	–			reserved		
		D23	LUT65	Look-up table entry 6 data	0x0 to 0x3F	0	R/W	
		D22	LUT64					
		D21	LUT63					
		D20	LUT62					
		D19	LUT61					
		D18	LUT60					
		D17–16	–	reserved	–	–	–	0 when being read.
		D15	LUT55	Look-up table entry 5 data	0x0 to 0x3F	0	R/W	
		D14	LUT54					
		D13	LUT53					
D12	LUT52							
D11	LUT51							
D10	LUT50							
D9–8	–	reserved	–	–	–	0 when being read.		
D7	LUT45	Look-up table entry 4 data	0x0 to 0x3F	0	R/W			
D6	LUT44							
D5	LUT43							
D4	LUT42							
D3	LUT41							
D2	LUT40							
D1–0	–	reserved	–	–	–	0 when being read.		

0x301AA8–0x301AAC **LCD Controller**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Look-up table data register 2 (pLCDC_LUT_8B)	00301AA8 (W)	D31	LUTB5	Look-up table entry 11 data	0x0 to 0x3F	0	R/W	
		D30	LUTB4			0		
		D29	LUTB3			0		
		D28	LUTB2			0		
		D27	LUTB1			0		
		D26	LUTB0			0		
		D25–24	–	reserved	–	–	–	0 when being read.
		D23	LUTA5	Look-up table entry 10 data	0x0 to 0x3F	0	R/W	
		D22	LUTA4			0		
		D21	LUTA3			0		
		D20	LUTA2			0		
		D19	LUTA1			0		
		D18	LUTA0			0		
		D17–16	–	reserved	–	–	–	0 when being read.
		D15	LUT95	Look-up table entry 9 data	0x0 to 0x3F	0	R/W	
D14	LUT94	0						
D13	LUT93	0						
D12	LUT92	0						
D11	LUT91	0						
D10	LUT90	0						
D9–8	–	reserved	–	–	–	0 when being read.		
D7	LUT85	Look-up table entry 8 data	0x0 to 0x3F	0	R/W			
D6	LUT84			0				
D5	LUT83			0				
D4	LUT82			0				
D3	LUT81			0				
D2	LUT80			0				
D1–0	–	reserved	–	–	–	0 when being read.		
Look-up table data register 3 (pLCDC_LUT_CF)	00301AAC (W)	D31	LUTF5	Look-up table entry 15 data	0x0 to 0x3F	0	R/W	
		D30	LUTF4			0		
		D29	LUTF3			0		
		D28	LUTF2			0		
		D27	LUTF1			0		
		D26	LUTF0			0		
		D25–24	–	reserved	–	–	–	0 when being read.
		D23	LUTE5	Look-up table entry 14 data	0x0 to 0x3F	0	R/W	
		D22	LUTE4			0		
		D21	LUTE3			0		
		D20	LUTE2			0		
		D19	LUTE1			0		
		D18	LUTE0			0		
		D17–16	–	reserved	–	–	–	0 when being read.
		D15	LUTD5	Look-up table entry 13 data	0x0 to 0x3F	0	R/W	
D14	LUTD4	0						
D13	LUTD3	0						
D12	LUTD2	0						
D11	LUTD1	0						
D10	LUTD0	0						
D9–8	–	reserved	–	–	–	0 when being read.		
D7	LUTC5	Look-up table entry 12 data	0x0 to 0x3F	0	R/W			
D6	LUTC4			0				
D5	LUTC3			0				
D4	LUTC2			0				
D3	LUTC1			0				
D2	LUTC0			0				
D1–0	–	reserved	–	–	–	0 when being read.		

0x301B00–0x301B04

Clock Management Unit

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
Gated clock control register 0 (pCMU _GATEDCLK0)	00301B00 (W)	D31–10	–	reserved	–	–	–	–	0 when being read.	
		D9	USBSAPB_CKE	USB SAPB I/F clock control	1	On	0	Off	0	R/W
		D8	USB_CKE	USB IP 48 MHz clock control					0	R/W
		D7	SDAPCPU_HCKE	SDRAMC CPU APP clock control (HALT)					0	R/W
		D6	SDAPCPU_CKE	SDRAMC CPU APP clock control					0	R/W
		D5	SDAPLDC_CKE	SDRAMC LCDC APP clock control					0	R/W
		D4	SDSAPB_CKE	SDRAMC SAPB I/F clock control					0	R/W
		D3	DSTRAM_CKE	DST RAM clock control					1	R/W
		D2	LCDCAHBIF_CKE	LCDC AHB I/F clock control					0	R/W
		D1	LCDCSAPB_CKE	LCDC SAPB I/F clock control					0	R/W
		D0	LCDC_CKE	LCDC main clock control					0	R/W
Gated clock control register 1 (pCMU _GATEDCLK1)	00301B04 (W)	D31–30	–	reserved	–	–	–	–	0 when being read.	
		D29	CPUAHB_HCKE	CPU_AHB bus clk control (HALT)	1	On	0	Off	1	R/W
		D28	LCDCAHB_HCKE	LCDC_AHB bus clk control (HALT)					1	R/W
		D27	GPIONSTP_HCKE	GPIO no stop clock control (HALT)					1	R/W
		D26	SRAMC_HCKE	SRAMC clock control (HALT)					1	R/W
		D25	EFSIOBR_HCKE	EFSIO baud rate clk control (HALT)					1	R/W
		D24	MISC_HCKE	Misc clock control (HALT)					1	R/W
		D23–20	–	reserved	–	–	–	–	–	0 when being read.
		D19	IVRAMARB_CKE	IVRAM arbiter clock control	1	On	0	Off	1	R/W
		D18	TM5_CKE	16-bit timer 5 clock control					1	R/W
		D17	TM4_CKE	16-bit timer 4 clock control					1	R/W
		D16	TM3_CKE	16-bit timer 3 clock control					1	R/W
		D15	TM2_CKE	16-bit timer 2 clock control					1	R/W
		D14	TM1_CKE	16-bit timer 1 clock control					1	R/W
		D13	TM0_CKE	16-bit timer 0 clock control					1	R/W
		D12	EGPIO_CKE	EGPIO clock control					1	R/W
		D11	I2S_CKE	I ² S clock control					1	R/W
		D10	DCSIO_CKE	DCSIO clock control					1	R/W
		D9	WDT_CKE	Watchdog timer clock control					1	R/W
		D8	GPIO_CKE	GPIO normal clock control					1	R/W
		D7	SRAMSAPB_CKE	SRAMC SAPB I/F clock control					1	R/W
		D6	SPI_CKE	SPI clock control					1	R/W
		D5	EFSIOSAPB_CKE	EFSIO SAPB I/F clock control					1	R/W
		D4	CARD_CKE	CARD I/F clock control					1	R/W
		D3	ADC_CKE	ADC clock control					1	R/W
		D2	ITC_CKE	ITC clock control					1	R/W
		D1	DMA_CKE	DMAC clock control					1	R/W
D0	RTCSAPB_CKE	RTC SAPB I/F clock control					1	R/W		

0x301B08

Clock Management Unit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
System clock control register (pCMU_CLKCNTL) Protected	00301B08 (W)	D31-29	–	reserved	–	–	–	0 when being read.			
		D28	CMU_CLKSEL4	CMU_CLK output clock source selection	CMU_CLKSEL[4:0]	Clock source	0	R/W			
		D27	CMU_CLKSEL3		Other	reserved	0				
		D26	CMU_CLKSEL2		01010	OSC3_DIV*1/32	0				
		D25	CMU_CLKSEL1		01001	OSC3_DIV*1/16	0				
		D24	CMU_CLKSEL0		01000	OSC3_DIV*1/8	0				
					00111	OSC3_DIV*1/4					
					00110	OSC3_DIV*1/2					
					00101	OSC3_DIV*1/1					
					00100	LCDC_CLK					
					00011	MCLK					
					00010	PLL					
					00001	OSC1					
					00000	OSC3					
		D23	PLLINDIV3		PLL input clock source divider selection	PLLINDIV[3:0]	Divider			0	R/W
		D22	PLLINDIV2			Other	OSC3*1/8			1	
		D21	PLLINDIV1			1001	OSC3*1/10			1	
		D20	PLLINDIV0			1000	OSC3*1/9			1	
				0111		OSC3*1/8					
				0110		OSC3*1/7					
				0101		OSC3*1/6					
				0100		OSC3*1/5					
				0011		OSC3*1/4					
				0010		OSC3*1/3					
				0001		OSC3*1/2					
				0000	OSC3*1/1						
		D19	LCDCDIV3	LCDC clock divider selection	LCDCDIV[3:0]	Divider	0	R/W			
		D18	LCDCDIV2		1111	OSC3*1/16	1				
		D17	LCDCDIV1		1110	OSC3*1/15	1				
		D16	LCDCDIV0		1101	OSC3*1/14	1				
					1100	OSC3*1/13					
					1011	OSC3*1/12					
					1010	OSC3*1/11					
					1001	OSC3*1/10					
					1000	OSC3*1/9					
					0111	OSC3*1/8					
					0110	OSC3*1/7					
				0101	OSC3*1/6						
				0100	OSC3*1/5						
				0011	OSC3*1/4						
				0010	OSC3*1/3						
				0001	OSC3*1/2						
				0000	OSC3*1/1						
		D15-13	–	reserved	–	–	–	–	0 when being read.		
		D12	MCLKDIV	MCLK clock divider selection	1 1/2	0 1/1	0	R/W			
		D11	–	reserved	–	–	–	–	0 when being read.		
		D10	OSC3DIV2	OSC3 clock divider selection	OSC3DIV[2:0]	Divider	0	R/W			
D9	OSC3DIV1	111	OSC3*1/1		0						
D8	OSC3DIV0	110	OSC3*1/1		0						
		101	OSC3*1/32								
		100	OSC3*1/16								
		011	OSC3*1/8								
		010	OSC3*1/4								
		001	OSC3*1/2								
		000	OSC3*1/1								
D7-4	–	reserved	–	–	–	–	0 when being read.				
D3	OSCSEL1	OSC clock selection	OSCSEL[1:0]	Clock source	0	R/W					
D2	OSCSEL0		11	PLL	0						
			10	OSC3							
			01	OSC1							
			00	OSC3							
D1	SOSC3	High-speed oscillation (OSC3) On/Off	1 On	0 Off	1	R/W					
D0	SOSC1	Low-speed oscillation (OSC1) On/Off	1 On	0 Off	1	R/W					

0x301B0C–0x301B14

Clock Management Unit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
PLL control register (pCMU_PLL) Protected	00301B0C (W)	D31–24	–	reserved	–	–	–	0 when being read.		
		D23	PLLCS1	PLL LPF capacitance setup	Fixed at "00" (default)	0	R/W			
		D22	PLLCS0				0			
		D21	PLLBYP	PLL bypass mode setup	Fixed at "0" (default)	0	R/W			
		D20	PLLCP4	PLL charge pump current setup	Fixed at "10000" (default)		1	R/W		
		D19	PLLCP3				0			
		D18	PLLCP2				0			
		D17	PLLCP1				0			
		D16	PLLCP0				0			
		D15	PLLVC3	PLL VCO Kv setup		PLLVC[3:0] f _{vco} [MHz]	0	R/W		
		D14	PLLVC2				1000	360 < f _{vco} ≤ 400	0	
		D13	PLLVC1				0111	320 < f _{vco} ≤ 360	0	
		D12	PLLVC0				0110	280 < f _{vco} ≤ 320	1	
							0101	240 < f _{vco} ≤ 280		
					0100	200 < f _{vco} ≤ 240				
					0011	160 < f _{vco} ≤ 200				
					0010	120 < f _{vco} ≤ 160				
					0001	100 ≤ f _{vco} ≤ 120				
					Other	Not allowed				
		D11	PLLRS3	PLL LPF resistance setup		PLLRS[3:0] f _{REFCK} [MHz]	1	R/W		
D10	PLLRS2	1010	5 ≤ f _{REFCK} < 20				0			
D9	PLLRS1	1000	20 ≤ f _{REFCK} ≤ 150				0			
D8	PLLRS0	Other	Not allowed				0			
D7	PLLN3	PLL multiplication rate setup		PLLN[3:0] Multiplication rate	0	R/W				
D6	PLLN2				1111	x16	0			
D5	PLLN1				1110	x15	0			
D4	PLLN0				:	:	0			
					0001	x2				
		0000	x1							
D3	PLLV1	PLL V-divider setup		PLLV[1:0] W	0	R/W				
D2	PLLV0				11	8	1			
					10	4				
					01	2				
					00	Not allowed				
D1	–	reserved	–	–	–	0 when being read.				
D0	PLLPOWR	PLL on/off control	1 On	0 Off	0	R/W				
SSCG macro control register (pCMU_SSCG) Protected	00301B10 (W)	D31–16	–	reserved	–	–	–	0 when being read.		
		D15	SSMCITM3	SSCG macro interval timer (ITM) setting	0 to 0xF	1	R/W			
		D14	SSMCITM2					1		
		D13	SSMCITM1					1		
		D12	SSMCITM0					1		
		D11	SSMCIDT3	SSCG macro maximum frequency change width setting	0 to 0xF	0	R/W			
		D10	SSMCIDT2					0		
		D9	SSMCIDT1					0		
		D8	SSMCIDT0					0		
		D7–1	–	reserved	–	–	–	0 when being read.		
D0	SSMCON	SSCG macro On/Off	1 On	0 Off	0	R/W				
Clock option register (pCMU_OPT) Protected	00301B14 (W)	D31–16	–	reserved	–	–	–	0 when being read.		
		D15	OSCTM7	OSC oscillation stabilization-wait timer	0 to 255	0	R/W			
		D14	OSCTM6					0		
		D13	OSCTM5					0		
		D12	OSCTM4					0		
		D11	OSCTM3					0		
		D10	OSCTM2					0		
		D9	OSCTM1					0		
		D8	OSCTM0					0		
		D7–4	–	reserved	–	–	–	0 when being read.		
		D3	OSC3OFF	OSC3 disable during SLEEP	1 Stop	0 Run	0	R/W		
		D2	TMHSP	Wait-timer high-speed mode	1 High speed	0 Normal	0	R/W		
D1	–	reserved	–	–	–	0 when being read.				
D0	WAKEUPWT	Wakeup-wait function enable	1 Wait interrupt	0 No wait	0	R/W				

0x301B24**Clock Management Unit**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock control protect register (pCMU_PROTECT)	00301B24 (W)	D31-8	–	reserved	–	–	–	0 when being read.
		D7	CLGP7	Clock control register protect flag	Writing 10010110 (0x96) removes the write protection of the clock control registers (0x301B00–0x301B14). Writing another value set the write protection.	0	R/W	
		D6	CLGP6					
		D5	CLGP5					
		D4	CLGP4					
		D3	CLGP3					
		D2	CLGP2					
		D1	CLGP1					
		D0	CLGP0					

0x301C00–0x301C20

I²S Interface

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S control register (pl2S_CONTRL)	00301C00 (W)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	I2SEN	I ² S module enable	1 Enabled 0 Disabled	0	R/W		
		D6	I2SOUTEN	I ² S output enable	1 Enabled 0 Disabled	0	R/W		
		D5	CHMD1 CHMD0	I ² S output channel mode	CHMD[1:0] Mode		0	R/W	
		11 Mute			0				
		10 Mono left							
		01 Mono right							
		D4			00 Stereo				
D3	WCLKMD	I ² S output word clock mode	1 L: high R: low	0 L: low R: high	0	R/W			
D2	BCLKPOL	I ² S output bit clock polarity	1 Negative	0 Positive	0	R/W			
D1	DTTMG	I ² S output data timing	1 Not delayed	0 Delayed	0	R/W			
D0	DTFORM	I ² S output data format	1 LSB first	0 MSB first	0	R/W			
I ² S MCLK divide ratio register (pl2S_DV_MCLK)	00301C04 (W)	D31–6	–	reserved	–	–	–	0 when being read.	
		D5	I2SMCLK5	I ² S_MCLK divide ratio selection	I2SMCLK[5:0] I2S_MCLK	0	R/W		
		D4	I2SMCLK4		111111 MCLK _(CMU) /64	0			
		D3	I2SMCLK3		111110 MCLK _(CMU) /63	0			
		D2	I2SMCLK2			0			
		D1	I2SMCLK1		000001 MCLK _(CMU) /2	0			
		D0	I2SMCLK0		000000 MCLK _(CMU) /1	0			
I ² S audio clock divide ratio register (pl2S_DV_LRCLK)	00301C08 (W)	D31–3	–	reserved	–	–	–	0 when being read.	
		D2	LRCLK2	LRCLK divide ratio selection	LRCLK[2:0] LRCLK	0	R/W		
		D1	LRCLK1		111 I2S_MCLK/512	0			
		D0	LRCLK0		110 I2S_MCLK/448	0			
					101 I2S_MCLK/384				
					100 I2S_MCLK/320				
					011 I2S_MCLK/256				
					010 I2S_MCLK/192				
		001 I2S_MCLK/128							
		000 I2S_MCLK/64							
I ² S start register (pl2S_START)	00301C0C (W)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	I2SBUSY	I ² S module busy flag	1 Busy 0 Idle	0	R		
		D6–1	–	reserved	–	–	–	0 when being read.	
		D0	I2SSTART	I ² S start/stop	1 Start 0 Stop	0	R/W		
I ² S HSDMA mode select register (pl2S_HSDMAMD)	00301C10 (W)	D31–1	–	reserved	–	–	–	0 when being read.	
		D0	I2SHSDMA	I ² S HSDMA mode selection	1 16 bits × 2 ch. 0 32 bits × 1 ch.	0	R/W		
I ² S FIFO status register (pl2S_FIFO_EMPTY)	00301C14 (W)	D31–1	–	reserved	–	–	–	0 when being read.	
		D0	I2SFIFOEF	I ² S FIFO empty flag	1 Empty 0 Not empty	0	R		
I ² S FIFO register (pl2S_FIFO)	00301C20 (W)	D31–0	I2SFIFO[31:0]	PCM data for writing to FIFO		0	W	0 when being read.	

THIS PAGE IS BLANK.

Appendix B Differences Between C33 PE Core and Other C33 Core

The functions below have been added to or changed for the C33 PE Core, based on functions of the C33 STD Core CPU (S1C33000). For details, see the “S1C33 Family C33 PE Core Manual.”

B.1 Instructions

The C33 PE Core instruction set is compatible with the C33 STD Core CPU, note, however, that some existing instructions have been function extended or removed and new instructions have been added for high-performance operations and cost reduction.

Function-extended instructions

The C33 PE Core has the following function-extended instructions.

- The number of bits shifted by shift/rotate instructions has been increased from 8 to 32.

<i>shift</i>	<i>%rd, imm5</i>	0–8 bits shift → 0–32 bits shift, <i>shift</i> = <i>srl, sll, sra, sla, rr, rl</i>
<i>shift</i>	<i>%rd, %rs</i>	0–8 bits shift → 0–32 bits shift, <i>shift</i> = <i>srl, sll, sra, sla, rr, rl</i>
- The data transfer instructions between a general-purpose register and a special register have been modified to support newly added special registers.

<i>ld.w</i>	<i>%sd, %rs</i>	Special register specifiable in <i>%sd</i> added
<i>ld.w</i>	<i>%rd, %ss</i>	Special register specifiable in <i>%ss</i> added

Added instructions

The instructions added to the C33 PE Core are listed below.

- Instructions specifically designed to save and restore single or special registers have been added.

<i>push</i>	<i>%rs</i>	Pushes single register
<i>pop</i>	<i>%rd</i>	Pops single register
<i>pushs</i>	<i>%ss</i>	Pushes special registers successively
<i>pop</i>	<i>%sd</i>	Pops special registers successively
- Instructions specifically designed for use with the coprocessor interface have been added.

<i>ld.c</i>	<i>%rd, imm4</i>	Coprocessor data transfer
<i>ld.c</i>	<i>imm4, %rs</i>	Coprocessor data transfer
<i>do.c</i>	<i>imm6</i>	Coprocessor execution
<i>ld.cf</i>		Coprocessor flag transfer
- Other special instructions have been added.

<i>swaph</i>	<i>%rd, %rs</i>	Switches between big and little endians
<i>psrset</i>	<i>imm5</i>	Sets the PSR bit
<i>psrclr</i>	<i>imm5</i>	Clears the PSR bit
<i>jpr</i>	<i>%rb</i>	Register indirect unconditional relative branch

Instructions removed

In the C33 PE Core, the instructions listed below have been removed from the instruction set of the C33 STD Core CPU.

<i>div0s</i>	Preprocessing for signed step division
<i>div0u</i>	Preprocessing for unsigned step division
<i>div1</i>	Step division
<i>div2s</i>	Correction of the result of signed step division, 1
<i>div3s</i>	Correction of the result of signed step division, 2
<i>mac</i>	Multiply-accumulate operation
<i>scan0</i>	Scan bits for 0
<i>scan1</i>	Scan bits for 1
<i>mirror</i>	Mirroring

These functions can be realized using the software library provided or by other means.

B.2 Registers

The general-purpose registers (R0 to R15) are basically the same as in the C33 STD Core CPU. The special registers have been functionally extended as described below.

PC

All 32 bits can now be used.

Moreover, the PC can now be read out to enable high-speed leaf calls.

Trap table base register

A trap table base register (TTBR) has been added.

TTBR, which was mapped at address 0x48134 in the C33 STD Core CPU, is incorporated in the C33 PE Core as a special register. The initial value (boot address) has not changed from 0xC00000.

Processor identification register

A processor identification register (IDIR) has been added for identifying the core type and version.

Debug base register

A debug base register (DBBR) has been added. This register indicates the start address of the debug area. It normally is fixed to 0x60000.

Processor status register

The following flags in PSR have been removed as have the related instructions:

MO flag (bit 7)	Mac overflow flag
DS flag (bit 6)	Divide sign

B.3 Address Space and Other

Address space

The C33 PE Core supports a 4G-byte space based on a 32-bit address bus.

Other

1. Interrupt/exception processing

The Trap Table Base Register (TTBR) now serves as an internal special register of the processor.

Furthermore, this processor has come to generate an exception when an undefined instruction (an object code not defined in the instruction set) is executed or more than two `ext` instructions are described.

2. Pipeline

The 3-stage pipeline in the C33 STD Core CPU has been modified to a 2-stage pipeline in the C33 PE Core (consisting of fetch/decode and execute/access/write back).

Appendix C Development Tools

C.1 Major Development Tools

The following shows the major development tools that support the S1C33E08:

1. S1C33 Family C/C++ Compiler Package (S5U1C33001C)

The S1C33 Family C/C++ Compiler Package includes software development tools for compiling C/C++ source files, assembling assembly source files, linking object files, debugging executable files, making mask data and other utilities. Also included in the package is the GNU33 IDE workbench that provides an integrated development environment from creating source files to debugging.

These tools run on Windows 2000 or Windows XP.

2. In-Circuit Debugger (S5U1C3300xH)

The In-Circuit Debugger (ICD) is a hardware tool used for debugging. By using an ICD with a target board, various debug functions can be executed from the debugger on a personal computer.

3. S1C33 Family Middleware Packages

Various kinds of middleware packages, such as real-time OS, file system middleware, audio and JPEG codec middleware, music play middleware, are provided.

C.2 Precautions on Use of S5U1C33001C

This section describes the precautions to develop an S1C33E08 application using the S1C33 Family C/C++ Compiler Package (S5U1C33001C). For common precautions and details of tools, see the S5U1C33001C manual.

Version of S5U1C33001C

Use the S5U1C33001C (Ver. 3) or later.

Precautions on use of C/C++ compiler and assembler

Compiler and assembler options

- When executing the compiler or assembler (or when creating a makefile), specify the `-mc33pe` option to generate the codes for the C33 PE Core.
- In GNU33 IDE, be sure to select C33 PE Core for the target CPU (select “PE” from [Properties] dialog for the project > [GNU33 General] > [Target CPU Macro type]).

CPU registers

Table C.2.1 General-Purpose Register Usage

Register	Method of use
%r0	Register used as a frame pointer Register that need has to their values saved when calling a function
%r1 %r2 %r3	Registers that need have to their values saved when calling a function
%r4 %r5	Register for storing returned values (8/16/32-bit data, 32 low-order bits of double-type data) Register for storing returned values (32 high-order bits of double-type data)
%r6 %r7 %r8 %r9	Register for passing argument (1st word) Register for passing argument (2nd word) Register for passing argument (3rd word) Register for passing argument (4th word)
%r10 %r11 %r12 %r13 %r14	Scratch register/unused
%r15	Default data area pointer register*

* When the `-medda32` option (default data area is not used) is not specified

- The C33 PE Core does not include the `%dp` register. Do not describe the `%dp` register in sources and do not specify functions that use the `%dp` register.
- Before the `%r0` to `%r3` registers can be used, the contents must be saved to the stack using the `pushn` instruction. Also, the saved contents must be restored from the stack using the `popn` instruction.
- The `%r4` and `%r5` registers can be used without saving/restoring the contents until a returned value is set in the register before returning.
- The `%r6` to `%r9` registers can be used after the stored arguments are used. It is not necessary to restore the contents before returning.
- The `%r10` to `%r15` registers are reserved by the as assembler and ld linker for referencing symbols. Try to use these registers as little as possible.
- Passing arguments to the function that returns a structure data
If the length of the structure data that is returned from a function is 8 bytes or less, the structure data is stored in the `%r4` and `%r5` registers used for storing returned values. In this case, the pointer to the structure that is normally sent as the 1st argument is not passed to the function.

Libraries

The S5U1C33001C package contains different library files for each C33 Core. Specify the C33 PE Core library files for the libraries to be linked.

Precautions on use of debugger

- Use the S5U1C33001H1100 (ICD33 Ver. 3.0) or later when debugging an S1C33E08 application using an ICD. Earlier versions of ICD do not support the S1C33E08. Furthermore, it is necessary to update the ICD firmware when using the S5U1C33001H1100 (ICD33 Ver. 3.0), since its standard specification does not support the S1C33E08.

The following S1C33E08 pins are used for debugging.

DSIO(P34)
DCLK(P35)
DST2(P36)

These pins must be configured for debugging. Fix the Debug Port MUX Register (0x300014) at 0x1 or the P34–P36 Port Function Select Register (0x3003A7) at 0x0 and do not switch the pin function.

When PC tracing is not performed, disable the PC trace function in the ICD by setting the DIP switch (SW5: on).

- The PC trace function of the ICD uses the S1C33E08 pins listed below.

DST0(P15/TM5/SOUT1/TFT_CTL0)
DST1(P16/DCSIO0/#SCLK1/TFT_CTL3)
DPCO(P17/DCSIO1/#SRDY1/TFT_CTL2)
DSIO(P34)
DCLK(P35)
DST2(P36)

These pins must be configured for debugging. Fix the Debug Port MUX Register (0x300014) at 0x1 and do not switch the pin function.

- When debugging a C33 PE Core application with the ICD, execute the `c33 das` command as shown below before executing the `target` command.

```
(gdb) c33 das 0x60000 0x84780 1
(gdb) target icd usb
```

(0x60000 = debug ROM address, 0x84780 = debug RAM address, 1 = C33 PE Core)

- The areas listed below are reserved for debugging or system use. Do not access these areas during debugging.
 - Area 0, addresses 0x0 to 0xF (when the debug monitor is used)
 - Area 1 (reserved for system)
 - Area 2 (reserved for debugging)
 - Area 3, addresses 0x84700 to 0x847FF and 0x90000 to 0xFFFFF (reserved for debugging/system)
- Setting **DSTRAM_CKE** (D3/0x301B00)

When the debug tools (ICD33 and GNU33) are used, addresses 0x846FF to 0x847FF in the DST RAM are used as the debug area. Therefore, set **DSTRAM_CKE** (D3/0x301B00) to 1 to supply the operating clock to the DST RAM before debugging can be started.

DSTRAM_CKE (D3/0x301B00) can be set to 0 to stop the DST RAM operation when the debug tool and the DST RAM are not used.

* **DSTRAM_CKE**: **DSTRAM** Clock Control Bit in the Gated Clock Control Register 0 (D3/0x301B00)

APPENDIX C DEVELOPMENT TOOLS

- Writing boot program using the debugger

The following shows the procedure to write a program to the boot device using the debugger and to start up the system from the boot device.

1. Make sure that the target board (S1C33E08) and ICD33 are off.
2. Configure the S1C33E08 BOOT[1: 0] pins, as you want.
3. Turn the target board (S1C33E08) and ICD33 on and start up the debugger on the PC.
4. Execute a command file to disable the boot sequence and to write a boot program to the boot device (NAND flash, NOR flash, SPI flash, etc.) from the debugger.
5. Terminate the debugger and turn the target board (S1C33E08) and ICD33 off.
6. Configure the S1C33E08 BOOT[1:0] pins according to the boot device.
7. Disconnect the ICD33 and turn the target board (S1C33E08) on.

The S1C33E08 boots up by the boot program written in the device specified.

The following shows an example command file to write the boot program in Step 4. Set the Boot Register (0x300018) to 0x01 to disable the boot sequence before erasing the flash and writing the program.

Example: To write the boot program test.elf to the NOR flash located in the #CE10 area

```
# gwb33.cmd command file made by GWB33
c33 das 0x60000 0x84780

# set the map in the debugger
# c33 rpf c33e08.par

# load symbol information
file am29f800.elf

# decide debugger mode and port
target icd com1
# target icd usb

# load to memory
load am29f800.elf

# boot disable, CE10 = 16 bits
set {char} 0x300020 = 0x96
set {char} 0x300018 = 0x01

# flash set command
c33 fls 0xc00000 0xcfffffff FLASH_ERASE FLASH_LOAD
# 0xc00000 = flash area start address
# 0xcfffffff = flash area end address
# FLASH_ERASE = flash erase routine top address
# FLASH_LOAD = flash load routine top address

#flash erase command
c33 fle 0xc00000 0x0 0x0
# 0xc00000 = flash control register
# 0x0 = erase start block (0: erase all areas 1-19)
# 0x0 = erase end block (this parameter is ignored as the start block is 0)

file test.elf

target icd com1
#target icd usb

# boot disable, CE10 = 16 bits
set {char} 0x300020 = 0x96
set {char} 0x300018 = 0x01

load test.elf
```

Appendix D Boot

D.1 Boot Mode

The S1C33E08 supports the five boot modes listed below.

- Large-page NAND Flash boot (Large page: 2048 + 64, 4096 + 128 bytes/page)
- Small-page NAND Flash boot (Small page: 512 + 16, 1024 + 32 bytes/page)
- NOR Flash/external ROM boot (Either 8 bits or 16 bits)
- SPI-EEPROM boot
- PC RS232C boot

The S1C33E08 boots up in the boot mode that can be selected with the BOOT and #CE10 pin configuration at initial reset.

Table D.1.1 Setting Boot Mode (PFBGA-180pin or die model)

BOOT1 pin	BOOT0 pin	#CE10	Boot mode	Boot code start address	MBR execution address
1	1	1 (Input)	SPI-EEPROM	0x20010 in the internal ROM (area 1)	0x0 in AORAM
		0 (Input)	PC RS232C		
1	0	Output	NOR Flash/external ROM	0x2000C in the internal ROM (area 1)	Depending on the contents in 0xC00000
0	1	–	Reserved	–	–
0	0	1 (Input)	Large-page NAND Flash (> 1024 + 32 bytes/page)	0x20004 in the internal ROM (area 1)	0x0 in AORAM
		0 (Input)	Small-page NAND Flash (≤ 1024 + 32 bytes/page)		

Table D.1.2 Setting Boot Mode (QFP-144pin model)

BOOT1 pin	#CE10	Boot mode	Boot code start address	MBR execution address
1	Output	NOR Flash/external ROM	0x2000C in the internal ROM (area 1)	Depending on the contents in 0xC00000
0	1 (Input)	Large-page NAND Flash (> 1024 + 32 bytes/page)	0x20004 in the internal ROM (area 1)	0x0 in AORAM
	0 (Input)	Small-page NAND Flash (≤ 1024 + 32 bytes/page)		

* The QFP-144pin model provides the BOOT1 pin only due to the limitation of number of pins (the BOOT0 pin is pulled down to V_{SS} inside the IC).

When the power is turned on or the chip is reset, the boot address is set to 0xC00000 in area 10 (TTBR initial value) due to the C33 PE Core specification. In the S1C33E08, a 4-word Gate ROM is located at area 10 (address 0xC00000 in the internal area). It contains vectors to jump to the boot sequence (boot code start address) for the boot mode specified by the pins above.

The boot sequence is programmed in the Specific ROM in area 1.

D.2 NAND Flash Boot

D.2.1 Configuration of NAND Flash Boot System

When the S1C33E08 is turned on or reset with both the BOOT1 and BOOT0 pins set to 0 (Vss), the S1C33E08 boots up by executing the MBR after loading it from the NAND Flash in the #CE11 area to A0RAM.

The NAND Flash conditions supported by the S1C33E08 are as follows:

- Data size: 8 bits or 16 bits
- Memory size: 1G bits max. (small page)
128G bits max. (large page)
- Page size: 512 + 16 bytes/page
1024 + 32 bytes/page
2048 + 64 bytes/page
4096 + 128 bytes/page

Figure D.2.1.1 shows a NAND Flash boot system connection diagram.

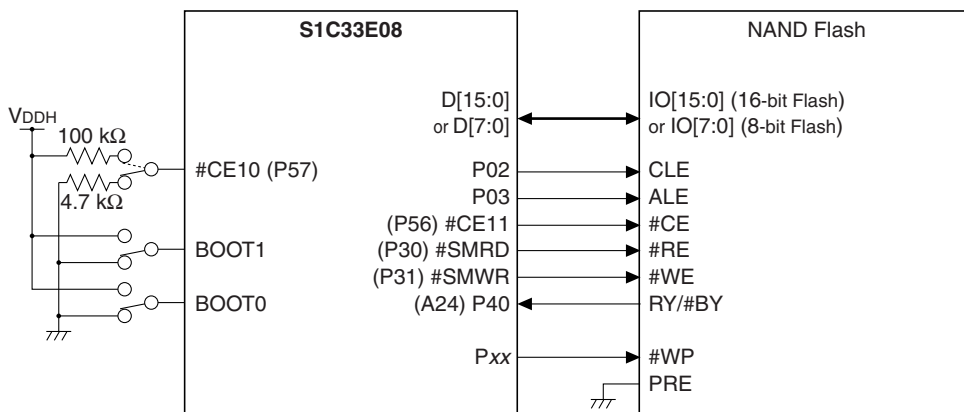


Figure D.2.1.1 NAND Flash Boot System

Table D.2.1.1 Pins Used for NAND Flash Boot

S1C33E08 pins	NAND Flash pins	Pin status before booting	Pin status after booting
#CE11/P56	#CE	High output	High output
P02 /#SCLK0/#DMAEND2	CLE	Input (Hi-Z)	Low output
P03 /#SRDY0/#DMAEND3	ALE	Input (Hi-Z)	Low output
P30 / CARD2 *1/#DMAREQ0	#RE	Input (Hi-Z)	High output
P31 / CARD3 *1/#DMAREQ1	#WE	Input (Hi-Z)	High output
A24 / P40 /#SDCAS/EXCL4	RY/#BY	Low output	Input (pulled up internally)
D[7:0]	IO[7:0]	Input (with bus latch)	Input (with bus latch)
D[15:8] /PC[7:0]	IO[15:8]	Input (with bus latch)	Input (with bus latch)
Pxx *2	#WP	(Input)	(Input)
#CE10/ P57 *3	–	Input (pulled up internally)	Input

*1: CARD2 is set to #SMRD and CARD3 is set to #SMWR.

*2: It should be controlled according to the application by the user program after booting.

*3: Used to specify the page size during boot mode configuration with the BOOT pins.

The pins listed in the table are configured for NAND Flash devices (pin names in **boldface**) in the boot sequence. Therefore, these pins cannot be used for general-purpose I/O or other peripheral functions.

D.2.2 NAND Flash Boot Sequence

Figure D.2.2.1 shows the NAND Flash boot flowchart.

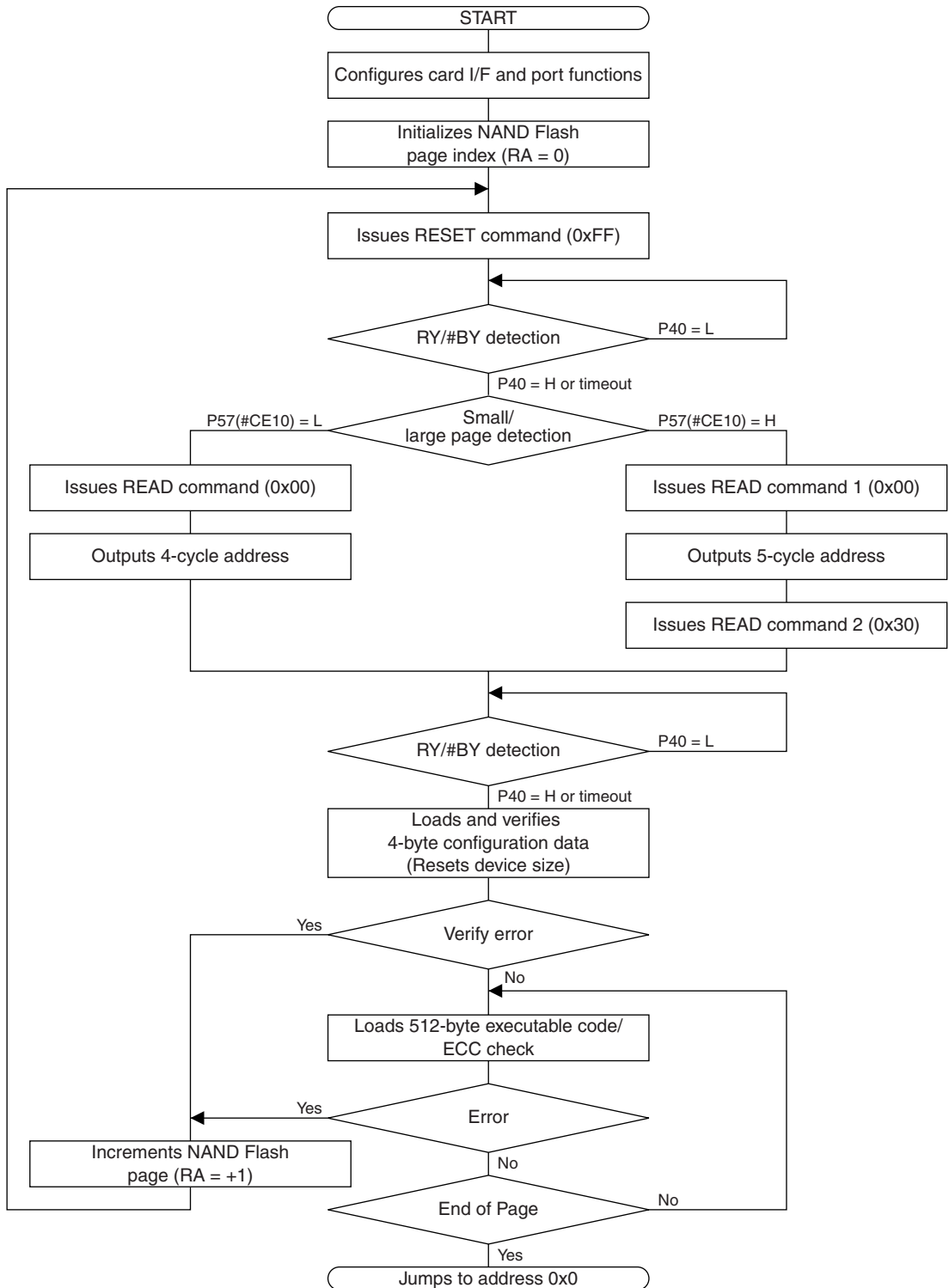


Figure D.2.2.1 NAND Flash Boot Flowchart

APPENDIX D BOOT

- (1) When the BOOT1 and BOOT0 pins are set to 0 at power-on or reset, the NAND Flash boot sequence programmed in the Specific ROM (area 1) is executed.
- (2) The boot sequence configures the port and the card interface.
- (3) Issues the reset command to reset the NAND Flash.
- (4) Reads the P57 (#CE10) pin status and issues a read command according to the page size determined by the read pin level.
 When P57 (#CE10) pin = 0, a read command for small page (1-cycle command with a 4-cycle address) is issued.
 When P57 (#CE10) pin = 1, a read command for large page (2-cycle command with a 5-cycle address) is issued.
- (5) Reads 512 + 16 bytes of data from the first page.
 The first 4 bytes are configuration data that indicates the NAND Flash bus size and page size. The subsequent contents are executable codes and ECC. (See Section D.2.3 for details.)
 According to the read configuration data, read operation continues until the end of the page.
 The executable codes are loaded from the beginning of A0RAM (0x0–).

 The read process performs an error check every 512 bytes using an 8-byte ECC that follows. If an error is detected, read operation from the current page is aborted. In this case, the sequence returns to Step (3) and reads MBR again from the next page.
- (6) When all the data in a page has been read, the sequence jumps to address 0x0 to execute the loaded codes.

Figures D.2.2.2 and D.2.2.3 show MBR reading start sequences for a small-page and a large-page NAND Flash.

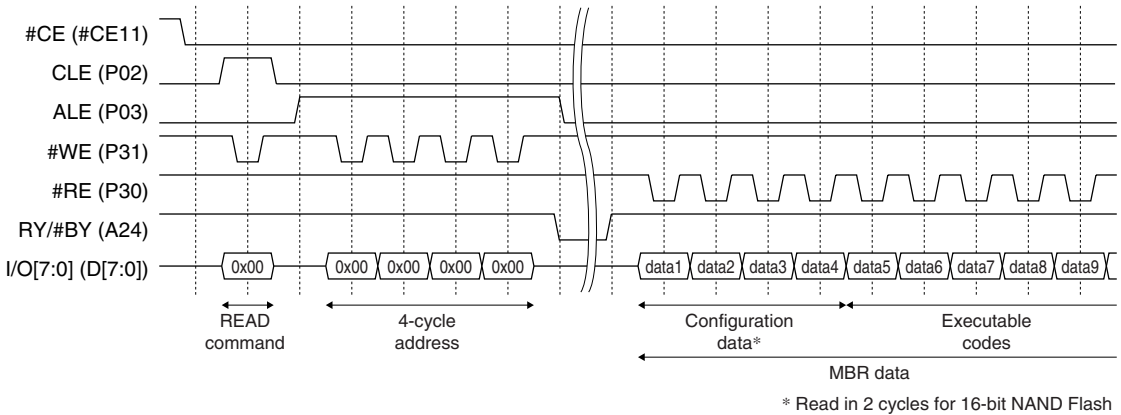


Figure D.2.2.2 Small-page NAND Flash Read (8-bit device)

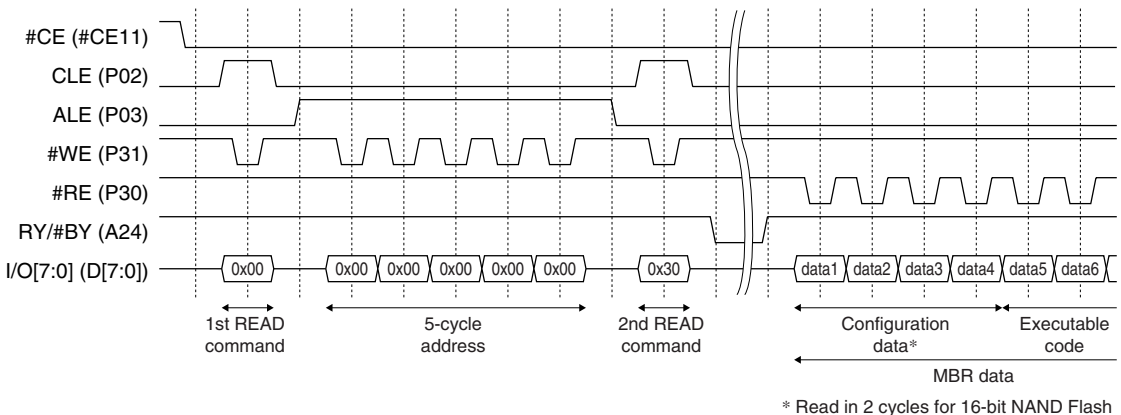


Figure D.2.2.3 Large-page NAND Flash Read (8-bit device)

D.2.3 NAND Flash Data

MBR code layout

Figure D.2.3.1 shows the MBR code layout according to the page size.

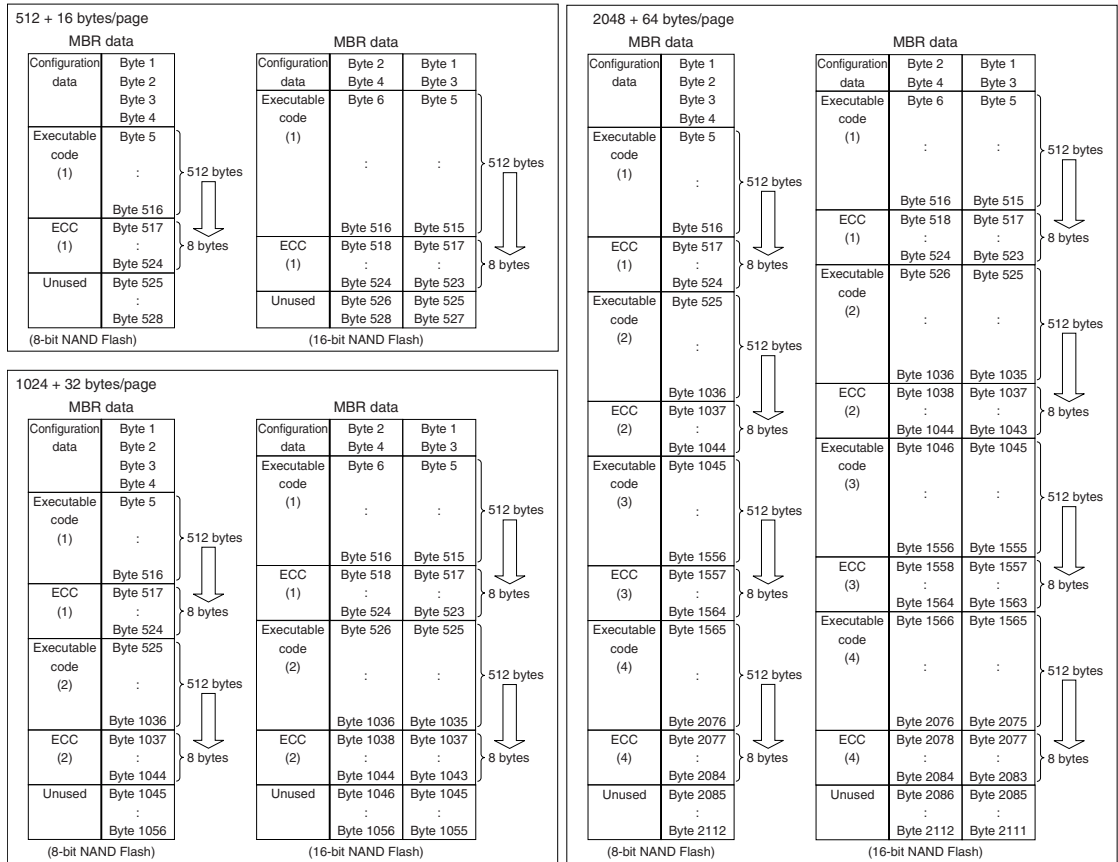


Figure D.2.3.1 MBR Code Layout

Configuration data

A 4-byte configuration data, which represents the NAND Flash bus size and page size, must be written at the beginning of the MBR code.

Figure D.2.3.2 below shows the bit configuration of the first byte.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
User ID		Bus size		Page size			

Figure D.2.3.2 First Byte of Configuration Data

Bits 7–6: User ID, or any user defined contents

This is a free-field allowing the user to define any ID. These 2 bits do not affect the boot up operation.

Bits 5–4: NAND Flash bus size

01: 16 bits 10: 8 bits Other: invalid

Bits 3–0: NAND Flash page size

0001: 512 + 16 bytes 0010: 1024 + 32 bytes 0100: 2048 + 64 bytes 1000: 4096 + 128 bytes
Other: invalid

The second to fourth bytes are redundant data used to check the configuration data.

Table D.2.3.1 shows the logical equations for expressing the second to fourth bytes and configuration data samples.

Table D.2.3.1 Configuration Data Samples

Byte No.	Equation	• 8 bits • 512 + 8	• 16 bits • 512 + 8	• 8 bits • 2048 + 64	• 16 bits • 2048 + 64
1st byte	– [DATA]	0x21	0x11	0x24	0x14
2nd byte	1st byte [DATA]	0xDE	0xEE	0xDB	0xEB
3rd byte	1st byte × 2nd byte [DATA × DATA]	0x9E	0xCE	0xCC	0x5C
4th byte	2nd byte ^ 3rd byte [DATA ^ (DATA × DATA)]	0x40	0x20	0x17	0xB7

^: Indicates XOR.

ECC data

An ECC (Error Correction Code) generator is embedded in the card interface 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 NAND Flash data area. So two 22-bit ECC parity data are generated per 512-byte data. 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.

The ECC check is performed in the boot sequence. If an error occurs, read operation from the current page is aborted and the sequence reads MBR again from the next page.

For this comparison, an 8-byte ECC data must be written for every 512 bytes of NAND Flash data. Figure D.2.3.3 shows the configuration of 8-byte ECC.

ECC byte 1	ECC byte 2	ECC byte 3	ECC byte 4	ECC byte 5	ECC byte 6	ECC byte 7	ECC byte 8
CP0[5:0]	0x00	LP0[7:0]	LP0[15:8]	CP1[5:0]	0x00	LP1[7:0]	LP1[15:8]

Figure D.2.3.3 ECC Data in NAND Flash

For the ECC algorithm, see Section V.4.6, “ECC Generator.”

D.3 NOR Flash/External ROM Boot

D.3.1 Configuration of NOR Flash/External ROM Boot System

When the S1C33E08 is turned on or reset with the BOOT1 pin set to 1 (V_{DDH}) and BOOT0 pin set to 0 (V_{SS}), the S1C33E08 reads the reset vector from address 0xC00000 in the external NOR Flash or external ROM and jumps to the user reset handler routine. This boot sequence is similar to the standard function of the C33 PE Core. However, the S1C33E08 supports booting from an 8-bit NOR Flash in contrast to the C33 PE Core that supports only a 16-bit device. The S1C33E08 reads the reset vector by executing the boot program written in the Specific ROM (area 1), and configures the #CE10 device size to 8 or 16 bits according to the LSB of the reset vector that is ignored in 16-bit boot. Then it jumps to the reset handler routine.

Figure D.3.1.1 shows a NOR Flash/external ROM boot system connection diagram.

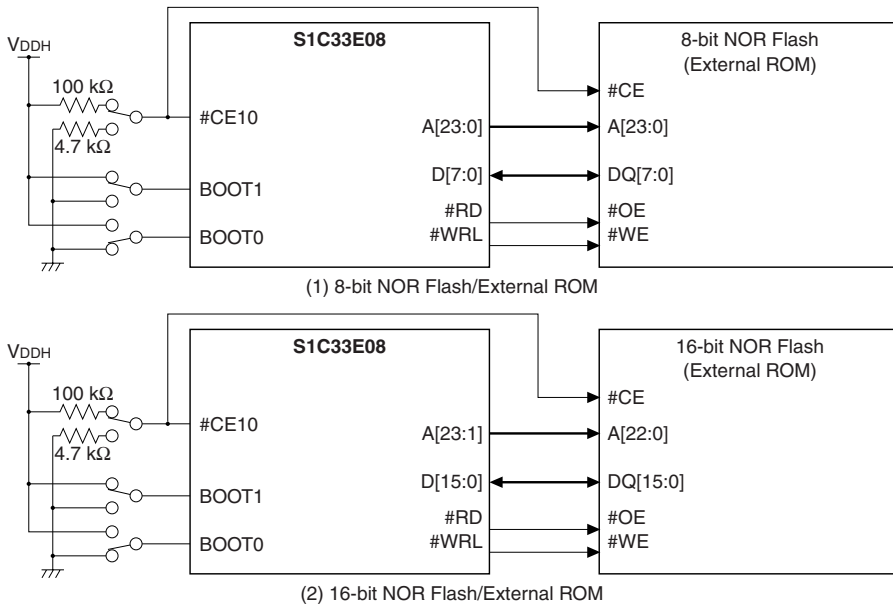


Figure D.3.1.1 NOR Flash/External ROM Boot System

This system uses only the external bus signals for #CE10 that are configured by default.

D.3.2 NOR Flash/External ROM Boot Sequence

Figure D.3.2.1 shows the NOR Flash/external ROM boot flowchart.

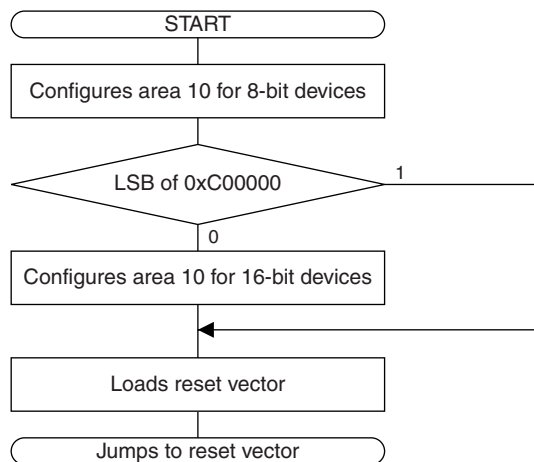


Figure D.3.2.1 NOR Flash/External ROM Boot Flowchart

- (1) When the BOOT1 pin is set to 1 and BOOT0 pin is set to 0 at power-on or reset, the NOR Flash/external ROM boot sequence programmed in the Specific ROM (area 1) is executed.
- (2) The boot sequence configures the SRAMC for setting the #CE10 area device size to 8 bits.
- (3) Checks the LSB of the reset vector written at address 0xC00000.
- (4) Sets the #CE10 area device size back to 16 bits if the LSB of the reset vector is 0.
Leaves it unchanged (8 bits) if the LSB of the reset vector is 1.
- (5) Reads the reset vector again and jumps to that address.

Figures D.3.2.2 and D.3.2.3 show 16-bit and 8-bit NOR Flash boot sequences.

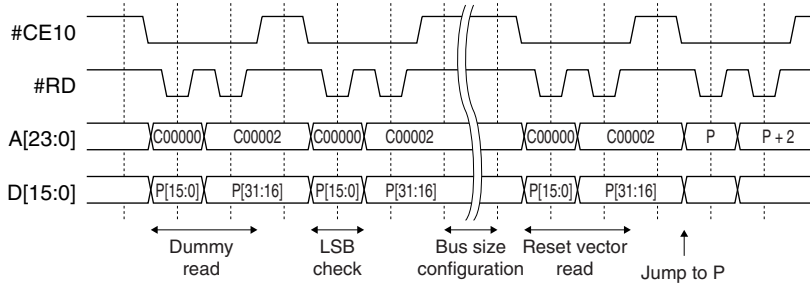


Figure D.3.2.2 16-bit NOR Flash Boot

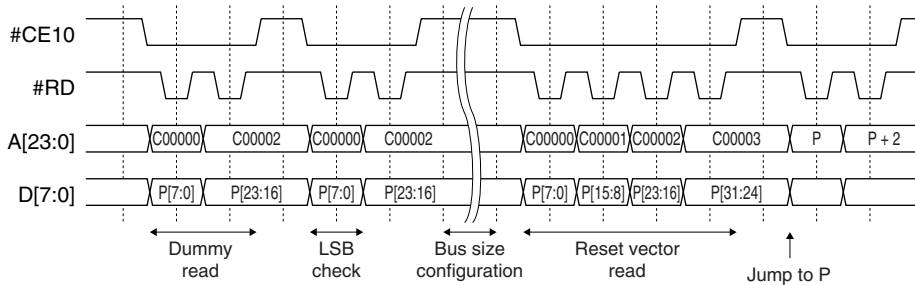


Figure D.3.2.3 8-bit NOR Flash Boot

D.3.3 Reset Vector for NOR Flash/External ROM Boot

To boot up the system from a 16-bit NOR Flash/external ROM, write a reset vector in which the LSB is set to 0 to address 0xC00000.

To boot up the system from an 8-bit NOR Flash/external ROM, write a reset vector in which the LSB is set to 1 to address 0xC00000.

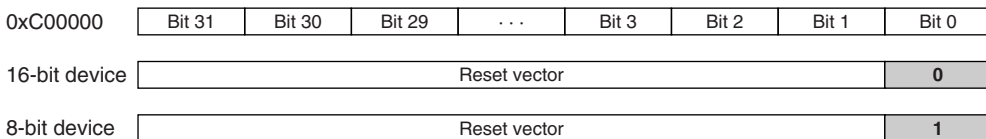


Figure D.3.3.1 Reset Vector for NOR Flash/External ROM Boot

The LSB of the reset vector is ignored when the program jumps to the user reset handler routine. Therefore, the jump destination is always a 16-bit boundary address even if the LSB is set to 1 for 8-bit boot.

D.4 SPI-EEPROM Boot

D.4.1 Configuration of SPI-EEPROM Boot System

When the S1C33E08 is turned on or reset with all the BOOT1, BOOT0, and #CE10 pins set to 1 (V_{DDH}), the S1C33E08 boots up by executing the MBR after loading it from the EEPROM, FRAM, or Serial Flash connected to the SPI bus to A0RAM.

Figure D.4.1.1 shows an SPI-EEPROM boot system connection diagram.

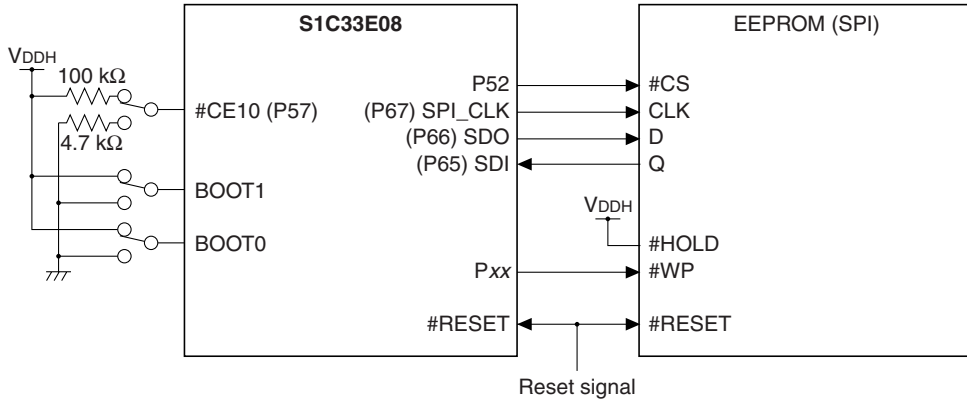


Figure D.4.1.1 SPI-EEPROM Boot System

Note: The QFP24-144pin package model does not support SPI-EEPROM boot.

Table D.4.1.1 Pins Used for SPI-EEPROM Boot

S1C33E08 pins	EEPROM pins	Pin status before booting	Pin status after booting
P52/BCLK/#CE6/CMU_CLK	#CS	Input (pulled up internally)	High output
P65/ SDI /FPDAT8	Q	Input (Hi-Z)	Input (Hi-Z)
P66/ SDO /FPDAT9	D	Input (Hi-Z)	Output
P67/ SPI_CLK /FPDAT10	CLK	Input (Hi-Z)	Low output
Pxx*1	#WP	(Input)	(Input)
#CE10/P57 *2	—	Input (pulled up internally)	Input

*1: It should be controlled according to the application by the user program after booting.

*2: Used to select SPI-EEPROM boot or PC RS232C boot during boot mode configuration with the BOOT pins.

The pins listed in the table are configured for SPI (pin names in **boldface**) in the boot sequence. Therefore, these pins cannot be used for general-purpose I/O or other peripheral functions.

This boot sequence supports up to 4GB (4-cycle address) of SPI-bus EEPROM. The SPI module is configured as below in the boot sequence.

Bit rate: OSC3 / 16 (e.g., 3 MHz when OSC3 = 48 MHz)

SPI mode: CPOL = 0, CPHA = 0

Data bit length: 8 bits

Master/slave mode: Master mode

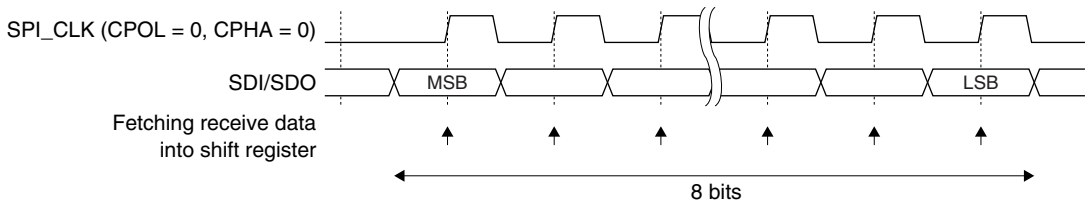


Figure D.4.1.2 SPI Mode

D.4.2 SPI-EEPROM Boot Sequence

Figure D.4.2.1 shows the SPI-EEPROM boot flowchart.

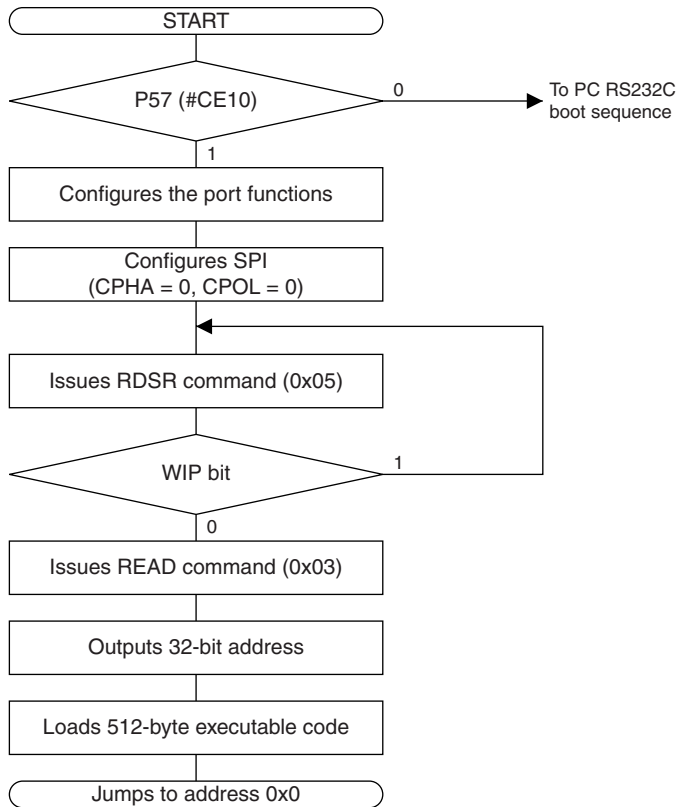


Figure D.4.2.1 SPI-EEPROM Boot Flowchart

- (1) When the BOOT1, BOOT0, and P57 (#CE10) pins are set to 1 at power-on or reset, the SPI-EEPROM boot sequence programmed in the Specific ROM (area 1) is executed.
- (2) The boot sequence configures the port and the SPI module.
- (3) Issues the RDSR (Read Status Register) command (0x05) to the EEPROM and reads the WIP (Write In Progress) bit to check the EEPROM status.
Waits for the EEPROM be ready status if it is busy.
- (4) Issues the READ command (0x03) with a 32-bit address (0x00 × 4 bytes) to the EEPROM.
- (5) Reads 512 bytes of executable codes.
The executable codes are loaded from the beginning of A0RAM (0x0–).
- (6) Jumps to address 0x0 to execute the loaded codes.

Figure D.4.2.2 shows MBR reading start sequences for a SPI-EEPROM.

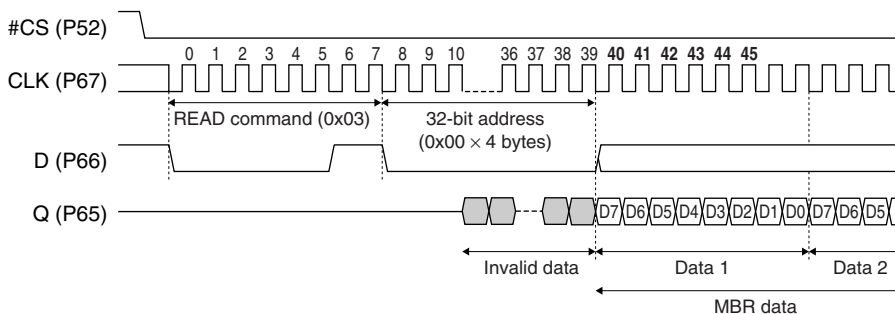


Figure D.4.2.2 EEPROM Read

D.4.3 EEPROM Data

The SPI-EEPROM boot sequence issues a 32-bit address regardless of the EEPROM size. Depending on the EEPROM size, data may be output during an address output period. Therefore, MBR codes must be followed by an appropriate offset. The boot sequence ignores the offset bytes.

Table D.4.3.1 and Figure D.4.3.1 show the data locations according to the EEPROM size.

Table D.4.3.1 EEPROM Size and Address Size

EEPROM size	Address size	MBR data location
1 to 256 bytes	1 byte	Byte 4 to Byte 256
0.25K to 64K bytes	2 bytes	Byte 3 to Byte 514
64K to 16M bytes	3 bytes	Byte 2 to Byte 513
16M to 4G bytes	4 bytes	Byte 1 to Byte 512

1-cycle address EEPROM

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	...	Byte 255	Byte 256
Invalid			253-byte executable code				

2-cycle address EEPROM

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	...	Byte 514	...
Invalid		512-byte executable code					

3-cycle address EEPROM

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	...	Byte 513	...
Invalid	512-byte executable code						

4-cycle address EEPROM

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	...	Byte 512	...
512-byte executable code							

Figure D.4.3.1 Data Location According to the EEPROM Size

D.5 PC RS232C Boot

D.5.1 Configuration of PC RS232C Boot System

When the S1C33E08 is turned on or reset with the BOOT1 and BOOT0 pins set to 1 (V_{DDH}) and the #CE10 pin set to 0 (V_{SS}), the S1C33E08 boots up by executing the MBR after loading it from the PC (RS232C) to A0RAM via the serial interface Ch.0.

Figure D.5.1.1 shows a PC RS232C boot system connection diagram.

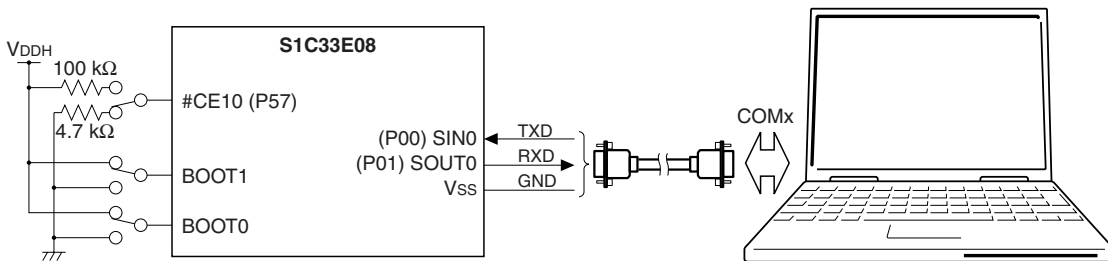


Figure D.5.1.1 PC RS232C Boot System

Note: The QFP24-144pin package model does not supports PC RS232C boot.

Table D.5.1.1 Pins Used for PC RS232C Boot

S1C33E08 pins	RS232C pins	Pin status before booting	Pin status after booting
P00/SIN0 / #DMAACK2	TXD	Input (Hi-Z)	Input (Hi-Z)
P01/SOUT0 / #DMAACK3	RXD	Input (Hi-Z)	Output
#CE10/P57 *1	–	Input (pulled up internally)	Input

*1: Used to select SPI-EEPROM boot or PC RS232C boot during boot mode configuration with the BOOT pins.

The pins listed in the table are configured for the serial interface Ch.0 (pin names in **boldface**) in the boot sequence. Therefore, these pins cannot be used for general-purpose I/O or other peripheral functions.

The baud rate and RS232C parameters are configured as below in the boot sequence.

Baud rate: Automatically detected, 9600 bps typ. (Note)

Data bit length: 8 bits

Start bit: 1 bit

Stop bit: 1 bit

Parity: None

Note: Table D.5.1.2 shows the maximum baud rate that can be set according to the S1C33E08 system clock (OSC3) frequency. The baud rate values in the vicinity of maximum frequency may have an error on the order of 5 percent. Use a lower baud rate to reduce an error.

Table D.5.1.2 System Clock Frequency and Baud Rate

OSC3 frequency	> 500 kHz	> 4 MHz	> 7 MHz	> 15 MHz	> 20 MHz	> 40 MHz
Maximum baud rate (bps)	1200	9600	19200	38400	57600	115200

D.5.2 PC RS232C Boot Sequence

Figure D.5.2.1 shows the PC RS232C boot flowchart.

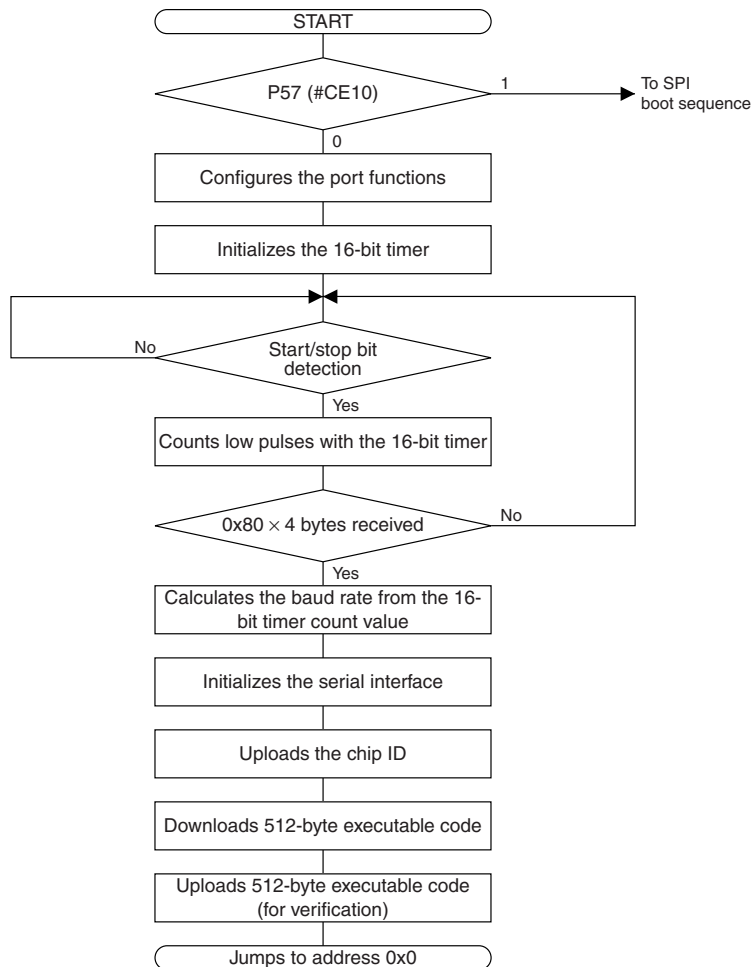


Figure D.5.2.1 PC RS232C Boot Flowchart

- (1) When the BOOT1 and BOOT0 pins are set to 1 and the P57 (#CE10) pin is set to 0 at power-on or reset, the PC RS232C boot sequence programmed in the Specific ROM (area 1) is executed.
- (2) The boot sequence configures the port and the 16-bit timer.
- (3) Waits for the P00 (SIN0) input pulled-down to 0 (start bit). When a start bit is input, the boot sequence starts the 16-bit timer to measure the low level width.
- (4) After 4 bytes of 0x80 sent from the PC have been input, the baud rate is calculated from the 16-bit timer count value, and the serial interface is enabled with the calculated baud rate.
- (5) The S1C33E08 uploads a 4-byte chip ID code to the PC.
The PC sends 512 bytes of executable codes to the S1C33E08 after the chip ID is verified.
- (6) The S1C33E08 downloads the 512 bytes of executable codes.
The executable codes are loaded from the beginning of A0RAM (0x0-).
- (7) The S1C33E08 uploads the downloaded 512 bytes of executable codes to the PC.
The PC verifies the codes with the original data for checking error.
- (8) The boot sequence jumps to address 0x0 to execute the loaded codes.

D.5.3 Transfer Data

First the PC sends 4 bytes of 0x80 to the S1C33E08. Then the PC sends 512-byte MBR data after verifying the 4-byte chip ID code received from the S1C33E08.

The S1C33E08 calculates the baud rate by counting the 4 bytes of 0x80 received from the PC and configures its serial interface. Then the S1C33E08 sends the 4-byte chip ID code to the PC. After the 512-byte MBR data is received, the S1C33E08 returns it to the PC for verification.

Figure D.5.3.1 shows the transfer data.

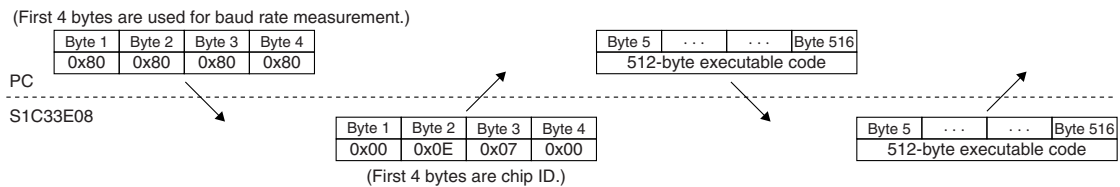


Figure D.5.3.1 Transfer Data for PC RS232C Boot

D.6 Precautions

The S1C33E08 supports various boot modes as described above.

In these boot modes, the following modes use the specific sequencer built into the S1C33E08.

- Small/large-page NAND flash boot
- SPI-EEPROM boot

The function and operation of the specific sequencer have been checked using the devices listed below, note, however, that we cannot guarantee that all NAND flash and SPI-EEPROM products work. Therefore, be sure to evaluate the function and check the operation of the boot sequence on the product system.

<Reference> NAND flash and SPI-EEPROM models that have completed the basic operation check for the S1C33E08

8-bit large page NAND flash	Toshiba	TC58NVG1S3BTG00
	Samsung	K9F2G08U0M-Y,P
8-bit small page NAND flash	STMicroelectronics	NAND512W3A2BN6
16-bit small page NAND flash	Samsung	K9F1216U0A
SPI-EEPROM	STMicroelectronics	ST M45PE80
	Winbond	NX25P16

THIS PAGE IS BLANK.

Appendix E Summary of Precautions

Memory

The areas listed below are reserved for debugging or system use. Do not access these areas from the user program or the debugger during debugging.

- Area 0, addresses 0x0 to 0xF (when the debug monitor is used)
- Area 1 (reserved for system)
- Area 2 (reserved for debugging)
- Area 3, addresses 0x84700 to 0x847FF and 0x90000 to 0xFFFFF (reserved for debugging/system)

High-Speed DMA (HSDMA)

- When setting the transfer conditions, always make sure the DMA controller is inactive ($HSx_EN (D0/0x30112C + 0x10 \cdot x) = 0$).
 - * **HSx_EN**: Ch.x Enable Bit in the HSDMA Ch.x Enable Register ($D0/0x30112C + 0x10 \cdot x$)
- After an initial reset, the cause-of-interrupt flag ($FHDMx (Dx/0x300281)$) becomes indeterminate. Always be sure to reset the flag to prevent interrupts or IDMA requests from being generated inadvertently.
 - * **FHDMx**: HSDMA Ch.x Cause-of-Interrupt Flag in the DMA Interrupt Cause Flag Register ($Dx/0x300281$)
- To prevent an interrupt from being generated repeatedly for the same source, be sure to reset the cause-of-interrupt flag before setting up the PSR again or executing the reti instruction.
- 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.
- In dual-address mode, A0RAM (area 0), Specific ROM (area 1), and IVRAM (area 0) cannot be specified as the source or destination for DMA transfer. While IVRAM (area 3), DST RAM (area 3) and the internal peripheral I/O registers (area 6) can be used for dual-address transfer.
- In single-address mode, A0RAM (area 0), Specific ROM (area 1), area 2, IVRAM (area 0 or area 3), DST RAM (area 3) and the internal peripheral I/O registers (area 6) cannot be used for DMA transfer.
- Single-address mode does not allow data transfer between memory devices. An external logic circuit is required to perform single-address transfer between memory devices.
- Single-address mode does not support the external memory area that is configured for SDRAM.
- Be sure to disable the HSDMA before setting the chip in SLEEP mode (executing the slp instruction). HALT mode can be set even if the HSDMA is enabled.

Intelligent DMA (IDMA)

- The control information must be placed in DST RAM (area 3) or an external RAM. Area 0 (A0RAM) and area 2 cannot be used for IDMA transfer and storing control information.
- The address you set in the IDMA base address registers must always be 4-word units boundary address.
- Be sure to disable DMA transfers ($IDMAEN (D0/0x301105) = 0$) before setting the base address. Writing to the IDMA base address register is ignored when the DMA transfer is enabled ($IDMAEN (D0/0x301105) = 1$). When the register is read, the read data is indeterminate.
 - * **IDMAEN**: IDMA Enable Bit in the IDMA Enable Register ($D0/0x301105$)
- Do not start an IDMA transfer and change the IDMA channel number simultaneously. When setting $DCHN[6:0] (D[6:0]/0x301104)$, write 0 to $DSTART (D7/0x301104)$.
 - * **DCHN[6:0]**: IDMA Channel Number Set-up Bits in the IDMA Start Register ($D[6:0]/0x301104$)
 - * **DSTART**: IDMA Start Control Bit in the IDMA Start Register ($D7/0x301104$)

APPENDIX E SUMMARY OF PRECAUTIONS

- 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.
- Since the C33 PE Core performs look-ahead operations, do not specify another channel immediately after a software trigger has invoked a channel.
- Be sure to disable the IDMA before setting the chip in SLEEP mode (executing the slp instruction). HALT mode can be set even if the IDMA is enabled.

SRAM Controller (SRAMC)

The BCLK pin output clock will not be divided regardless of how the BCLK divide-by ratio is set using BCLK (D0/0x301500); it is always the same as the SRAMC_CLK clock.

* **BCLK**: BCLK Divide Control Bit in the BCLK and Setup Time Control Register (D0/0x301500)

SDRAM Controller (SDRAMC)

If the operating clock (SDCLK) is stopped while the SDRAM is being accessed, a system failure may occur due to stoppage of the SDRAM operation in uncontrolled status. The following operations stop the SDCLK, therefore, do not perform these operations when the SDRAM may be accessed.

- Setting the S1C33E08 in SLEEP status
- Switching the P21 port function from SDCLK output to general-purpose input/output
- Disabling the clock supply to the SDRAMC module

Besides the CPU, the DMA controller (when DMA transfer from/to the SDRAM is enabled) and the LCD controller (when SDRAM is configured as the VRAM for the LCDC) access the SDRAM. In this case, before performing an above operation, disable the DMA transfer and the LCDC so that the SDRAM will not be accessed.

Clock Management Unit (CMU)

- The clock control registers (0x301B00–0x301B14) are write-protected. Before these registers can be rewritten, write protection must be removed by writing data 0x96 to the Clock Control Protect Register (0x301B24). Once write protection is removed, the clock control registers can be written to any number of times until the protect register is reset to other than 0x96. Note that since unnecessary rewriting of the clock control registers could lead to erratic system operation, the Clock Control Protect Register (0x301B24) should be set to other than 0x96 unless the clock control registers must be rewritten.
- When clock sources are changed, the clock control registers must be set so that the CMU is supplied with a clock from the selected clock source upon returning from SLEEP mode immediately after the change. Otherwise, the chip may not restart after return from SLEEP mode.
Furthermore, note that the timer, which generates an oscillation stabilization wait time after the SLEEP mode is released, operates with the clock after switching over. Be sure to use the correct clock frequency for calculating the wait time to be set to OSCTM[7:0] (D[15:8]/0x301B14) and TMSHP (D2/0x301B14).
 - * **OSCTM[7:0]**: OSC Oscillation Stabilization-Wait Timer in the Clock Option Register (D[15:8]/0x301B14)
 - * **TMSHP**: Stabilization-Wait Timer High-Speed Mode Select Bit in the Clock Option Register (D2/0x301B14)
- When SOSC3 (D1/0x301B08) or SOSC1 (D0/0x301B08) is set from 0 to 1 for initiating oscillation by the oscillator, a finite time is required until the oscillation stabilizes (e.g., 25 ms for OSC3 and 3 seconds for OSC1 in the S1C33E08). To prevent erratic operation, do not use the oscillator-derived clock until the oscillation start time stipulated in the electrical characteristics table elapses.
 - * **SOSC3**: High-speed Oscillation (OSC3) On/Off Control Bit in the System Clock Control Register (D1/0x301B08)
 - * **SOSC1**: Low-speed Oscillation (OSC1) On/Off Control Bit in the System Clock Control Register (D0/0x301B08)
- Immediately after the PLL is started by setting PLLPOWR (D0/0x301B0C) to 1, an output clock stabilization wait time is required (e.g., 200 μs in the S1C33E08). When the clock source for the system is switched over to the PLL, allow for this wait time after the PLL has turned on.
 - * **PLLPOWR**: PLL On/Off Control Bit in the PLL Control Register (D0/0x301B0C)

- The frequency multiplication rate of the PLL that can be set depends on the upper-limit operating clock frequency (90 MHz) and the OSC3 oscillation frequency. When setting the frequency multiplication rate, be sure not to exceed the upper-limit operating clock frequency.
- The PLL can only be set up when the PLL is turned off (PLLPOWR (D0/0x301B0C) = 0) and the clock source is other than the PLL (OSCSEL[1:0] (D[3:2]/0x301B08) = 0–2). If settings are changed while the system is operating with the PLL clock, the system may operate erratically.
 - * **OSCSEL[1:0]**: OSC Clock Select Bits in the System Clock Control Register (D[3:2]/0x301B08)
- Even if the #RESET pin is pulled low (= 0), the chip may not be reset unless supplied with a clock. To reset the chip for sure, #RESET should be held low for at least 3 OSC3 clock cycles. However, the input/output port pins will be initialized by reset regardless of whether the chip is supplied with a clock.
- The oscillation start time of the high-speed (OSC3) oscillator circuit varies with the device used, board patterns, and operating environment. Therefore, a sufficient time should be provided before the reset signal is deasserted.
- NMI cannot be nested. The CPU keeps NMI input masked out until the reti instruction is executed after an NMI exception occurred.
- When using the SSCG, always set SSMCITM[3:0] (D[15:12]/0x301B10) to 0b0001.
 - * **SSMCITM[3:0]**: SSCG Macro Interval Timer Setting Bits in the SSCG Macro Control Register (D[15:12]/0x301B10)
- SSMCIDT[3:0] (D[11:8]/0x301B10) must be set according to the PLL output clock frequency as shown in Table III.1.7.2.1. Using the SSCG with an improper setting may cause a malfunction of the IC.
 - * **SSMCIDT[3:0]**: SSCG Macro Maximum Frequency Change Width Setting Bits in the SSCG Macro Control Register (D[11:8]/0x301B10)
- When the PLL is off, the initial values and the written values cannot be read correctly from SSMCIDT[3:0] (D[11:8]/0x301B10) and SSMCITM[3:0] (D[15:12]/0x301B10) since the source clock is not supplied from the PLL (different values are read out). The correct values can be read out when the PLL is on.
- A stabilized clock must be supplied to the SSCG module when turning the SSCG on and off. The following shows the operation procedure.

To turn the SSCG on <ol style="list-style-type: none"> 1. Turn the PLL on. 2. Wait more than the PLL stabilization time. 3. Turn the SSCG on. 	To turn the SSCG off <ol style="list-style-type: none"> 1. Turn the SSCG off. 2. Turn the PLL off.
--	--
- The SS modulation is effective only for the PLL output clock, and is not performed for other source clocks. When the PLL output clock is not used for the system clock, turn the SSCG off.

Interrupt Controller (ITC)

- In SLEEP mode, there is a time lag between input of an interrupt signal for wakeup and the start of the clock supply to the ITC, so a delay will occur until the ITC sets the cause-of-interrupt flag. Therefore, no interrupt will occur if the interrupt signal is deasserted before the clock is supplied to the ITC, as the cause-of-interrupt flag in the ITC is not set.

Furthermore, additional time is needed for the CPU to accept the interrupt request from the ITC, the CPU may execute a few instructions that follow the slp instruction before it starts the interrupt processing. The same problem may occur when the CPU wakes up from SLEEP mode by NMI. No interrupt will occur if the #NMI signal is deasserted before the clock is supplied, as the NMI flag is not set.
- If the cause of interrupt used to restart from the standby mode has been set to invoke the IDMA, the IDMA is started up by that interrupt.

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.
- As the C33 PE Core 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.

APPENDIX E SUMMARY OF PRECAUTIONS

- When the reset-only method is used to reset the cause-of-interrupt flag (by writing 1), if a read-modify-write instruction (e.g., bset, bclr, or bnot) is executed, the other cause-of-interrupt 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 cause-of-interrupt flag (by writing 0), all cause-of-interrupt flags for which 0 has been written are reset. When a read-modify-write operation is performed, a cause of interrupt 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.

- After an initial reset, the cause-of-interrupt 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.
- To prevent another interrupt from being generated for the same cause again after generation of an interrupt, be sure to reset the cause-of-interrupt flag before enabling interrupts and setting the PSR again or executing the reti instruction.
- There is a time lag between latching the interrupt signal and latching the interrupt vector and level signals caused by the interface specifications between the CPU and the ITC.
 1. The CPU latches the interrupt signal sent from the ITC.
↓
 2. The CPU latches the interrupt vector and level signals sent from the ITC.
↓
 3. The CPU executes the interrupt handler.

An illegal interrupt exception (vector No. 11) occurs when a register related to the interrupt signal (ITC's interrupt enable and cause-of-interrupt flag registers) is altered before the CPU latches the interrupt vector and level signals (between Steps 1 and 2).

Therefore, it is very rare but an illegal interrupt exception may occur if an interrupt related register is altered when interrupts to the CPU are in enabled status (IE bit in PSR = 1).

However, the illegal interrupt exception that occurs does not affect the program execution if any processing is not performed in the exception handler.

To avoid an illegal interrupt exception occurring, disable interrupts to the CPU (set IE bit in PSR = 0) before altering an interrupt related register.

Real-Time Clock (RTC)

- The contents of all RTC control registers are indeterminate when power is turned on and are not initialized to specific values by initial reset. Be sure to initialize these registers in software.
- While 1 is being carried over to the next-digit counter, the correct counter value may not be read out. Moreover, attempting to write to the counters or other control registers may corrupt the counter value. Therefore, do not write to the counters while 1 is being carried over. For the correct method of operation, see Section III.3.3.5, "Counter Hold and Busy Flag," and Section III.3.3.6, "Reading from and Writing to Counters in Operation."
- Note that rewriting RTC24H (D4/0x301908) to switch between 12-hour mode and 24-hour mode may corrupt the count data for hours, days, months, years, or days of the week. Therefore, after changing the RTC24H (D4/0x301908) setting, be sure to set data in these counters back again.

* **RTC24H:** 24H/12H Mode Select Bit in the RTC Control Register (D4/0x301908)

- Avoid the settings below that may cause timekeeping errors.

- Settings exceeding the effective range

Do not set count data exceeding 60 seconds, 60 minutes, 12 or 24 hours, 31 days, 12 months, or 99 years.

- Settings nonexistent in the calendar

Do not set nonexistent dates such as April 31 or February 29, 2006. Even if such settings are made, the counters operate normally, so that when 1 is carried over from the hour counter to the 1-day counter, the day counter counts up to the first day of the next month. (For April 31, the day counter counts up to May 1; for February 29, 2006, the day counter counts up to March 1, 2006.)

- The contents of all RTC interrupt control bits are indeterminate when power is turned on, and are not initialized to specific values by initial reset.
After power-on, be sure to set RTCIEN (D0/0x301904) to 0 (interrupt disabled) for preventing the occurrence of unwanted RTC interrupts. Also be sure to write 1 to RTCIRQ (D0/0x301900) to reset it.
 - * **RTCIEN**: RTC Interrupt Enable Bit in the RTC Interrupt Mode Register (D0/0x301904)
 - * **RTCIRQ**: Interrupt Status Bit in the RTC Interrupt Status Register (D0/0x301900)
- Immediately after the OSC1 oscillator circuit is activated (as at power-on), a finite time (of about 3 seconds) is required for OSC1 oscillation to stabilize. Do not let the RTC start counting until this time elapses.

Misc Registers

- The Misc registers at addresses 0x300010–0x30001A are write-protected. Before the Misc registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to the Misc Protect Register (0x300020). Note that since unnecessary rewrites to addresses 0x300010–0x30001A could lead to erratic system operation, the Misc Protect Register (0x300020) should be set to other than 0x96 unless said Misc registers must be rewritten.
- The low-drive control bit is disabled when the pin is used as the general-purpose I/O port (Pxx).
- If the bus 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.

16-Bit Timers (T16)

- When setting the count clock or operation mode, make sure the 16-bit timer is turned off.
- 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.
- 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 $\times 1/2$.
- 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 $A > 2 \times B + 1$.
- After an initial reset, the cause-of-interrupt flag becomes indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset this flag and register in the software.
- To prevent another interrupt from being generated by the same cause of interrupt after an interrupt has occurred, be sure to reset the cause-of-interrupt flag before setting the PSR again or executing the reti instruction.

Watchdog Timer (WDT)

- When NMI or reset signal output by the watchdog timer is enabled, the watchdog timer must be reset within the set NMI/reset generation cycle.
- Do not set a value equal to or less than 0x0000001F in the comparison data register.
- Depending on the counter and comparison register values, an NMI or reset signal may be generated after the NMI or reset function is enabled, or immediately after the watchdog timer starts. Always be sure to set comparison data and reset the watchdog timer before writing 1 to NMIEN (D1/0x300662), RESEN (D0/0x300662), or RUNSTP (D4/0x300662).
 - * **NMIEN**: Watchdog Timer NMI Enable Bit in the Watchdog Timer Enable Register (D1/0x300662)
 - * **RESEN**: Watchdog Timer RESET Enable Bit in the Watchdog Timer Enable Register (D0/0x300662)
 - * **RUNSTP**: Watchdog Timer Run/Stop Control Bit in the Watchdog Timer Enable Register (D4/0x300662)

General-purpose Serial Interface (EFSIO)

- Before setting various serial-interface parameters, make sure the transmit and receive operations are disabled (TXEN_x = RXEN_x = 0).
 - * **TXEN_x**: Serial I/F Ch.*x* Transmit Enable Bit in the Serial I/F Ch.*x* Control Register (D7/0x300Bx3)
 - * **RXEN_x**: Serial I/F Ch.*x* Receive Enable Bit in the Serial I/F Ch.*x* Control Register (D6/0x300Bx3)
- When the serial interface is transmitting or receiving data, do not set TXEN_x or RXEN_x to 0, and do not execute the slp instruction.
- In clock-synchronized transfers, the mode of communication is half-duplex, in which the clock line is shared between the transmit and receive units. Therefore, RXEN_x and TXEN_x cannot be enabled simultaneously.
- After an initial reset, the cause-of-interrupt flags become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, reset these flags in the program.
- If a receive error occurs, the receive-error interrupt and receive-buffer full interrupt causes 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 cause flag through the use of the receive-error interrupt processing routine.
- To prevent the regeneration of interrupts due to the same cause of interrupt following the occurrence of an interrupt, always be sure to reset the cause-of-interrupt flag before setting the PSR again or executing the reti instruction.
- Follow the procedure described below to initialize the serial interface.

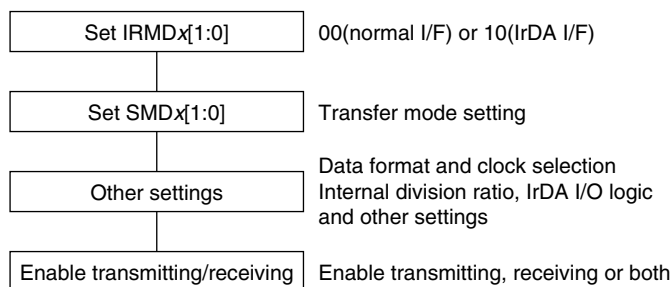


Figure E.1 Serial Interface Initialize Procedure

- When transmitting data in clock-synchronized master mode, transmit data is written to the transmit data register after the initial setting is performed following the flow above. 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.
- The maximum transfer rate of the serial interface is limited to 8 Mbps in clock-synchronized mode or 1 Mbps in asynchronous mode. Do not set a transfer rate (baud rate) that exceeds the limit.
- If the receive circuit is stopped during reception, set both transmission and reception to the disabled status.
- 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.
- When the transmit-enable bit TXEN_x is set to 0 to disable transmit operations, the transmit data buffer (FIFO) is cleared (initialized). Similarly, when the receive-enable bit RXEN_x 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.
- During IrDA receive operations, the RZI circuit recognizes low pulses by means of the signal edge (rising edge when IRRL_x = 0; falling edge when IRRL_x = 1). Note that noise may cause a malfunction.
 - * **IRRL_x**: Serial I/F Ch.*x* IrDA I/F Input Logic Inversion Bit in the Serial I/F Ch.*x* IrDA Register (D2/0x300Bx4)

Serial Peripheral Interface (SPI)

- Be sure to use 32-bit access instructions for reading/writing from/to the SPI control registers (0x301700 to 0x30171C). The SPI control registers do not allow reading/writing using 16-bit and 8-bit access instructions.
- Do not access the SPI Control Register 1 (0x301708), SPI Control Register 2 (0x30170C), and SPI Wait Register (0x301710) while the BSYF (D6/0x301714) is set to 1 (during data transfer).
 - * **BSYF**: Transfer Busy Flag in the SPI Status Register (D6/0x301714)
- To prevent malfunctions, write 0x0 to the SPI Interrupt Control Register (0x301718) to disable all the SPI interrupt requests, before disabling the SPI circuit (before setting ENA (D0/0x301708) to 0).
 - * **ENA**: SPI Enable Bit in the SPI Control Register 1 (D0/0x301708)

Direction Control Serial Interface (DCSIO)

- Be sure to use 32-bit access instructions for reading/writing from/to the DCSIO Data Load Register (0x301804). Transmit data for Line A and Line B must be written simultaneously.
- To prevent malfunctions, write 0x0 to the DCSIO Interrupt Control Register (0x301814) to disable all the DCSIO interrupt requests, before disabling the DCSIO circuit (before setting DCSIOEN (D0/0x301800) to 0).
 - * **DCSIOEN**: DCSIO Enable Bit in the DCSIO Control Register (D0/0x301800)
- When using the DCSIO ports as input/open-drain output ports (default configuration), be sure to enable the internal pull-up resistor (refer to Section III.4.4, “Pin Control Registers”) or to connect an external pull-up resistor to the pin.

Card Interface (CARD)

- The interface supports 16-bit PC cards, such as ATA (CF), LAN (Ethernet, wireless), or modem connected as an I/O card. SRAM cards, etc. are not supported.
- Live or hot-line card insertion and removal are not supported. Power must be turned off before inserting or removing a card. The automatic recognition of cards is also not supported.
- DMA, ZV, and CardBus are also not supported.
- To accommodate differences in power supply voltage between the PC card (5 V or 3.3 V) and the S1C33E08, use a buffer IC (e.g., S1C37120).
- The ECC generator supports two 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 card I/O signals must be connected to the D[15:0] pins when using the ECC function.
- ECCRST (D0/0x300311) should be set before using the ECC function.
 - * **ECCRST**: ECC Circuit Reset Bit in the ECC Reset/Ready Register (D0/0x300311)
- The ECCEN (D0/0x300312) should be set only while reading or writing card data. It should be disabled during command input, address input or status reading.
 - * **ECCEN**: ECC Circuit Enable Bit in the ECC Enable Register (D0/0x300312)

General-Purpose I/O Ports (GPIO)

- After an initial reset, the cause-of-interrupt flags become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset the flags in a program.
- To prevent regeneration of interrupts due to the same cause of interrupt following the occurrence of an interrupt, always be sure to reset the cause-of-interrupt flag before resetting the PSR or executing the reti instruction.
- When using an port input interrupt as the trigger to restart from the 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 SLEEP mode using a port input interrupt.

- To use the P15–P17 and P34–P36 pins that are configured as the debug interface pins by default for general-purpose inputs/outputs, clear TRCMUX (D0/0x300014) to 0.

* **TRCMUX:** P15–17, P34–36 Debug Function Select Bit in the Debug Port MUX Register (D0/0x300014)

Note, however, that the PC trace function of the debugger cannot be used when TRCMUX (D0/0x300014) is set to 0.

- Even if the port input interrupt condition is set to falling edge, the input pulse width must be longer than 1 cycle of the port operating clock (= MCLK) to be certain an interrupt will be generated.

A/D Converter (ADC)

- Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to disable ADE (D2/0x300544). A change in settings while the A/D converter is enabled could cause it to operate erratically.

* **ADE:** A/D Enable Bit in the A/D Control/Status Register (D2/0x300544)

- In consideration of the conversion accuracy, we recommend that the A/D conversion clock be min. 16 kHz to max. 2 MHz.
- 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.
- After an initial reset, FADE (D1/0x300287) and FADC (D0/0x300287) become indeterminate. To prevent generation of an unwanted interrupt or IDMA request, be sure to reset these flags in a program.

* **FADE:** A/D Conversion Completion Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D1/0x300287)

* **FADC:** A/D Out-of-Range Interrupt Cause Flag in the Port Input 4–7, RTC, A/D Interrupt Cause Flag Register (D0/0x300287)

- To prevent the regeneration of interrupts due to the same cause of interrupt following the occurrence an interrupt, always be sure to reset the cause-of-interrupt flag before setting the PSR again or executing the `reti` instruction.
- When the A/D converter is set to enabled state, a current flows between AVDD 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 (D2/0x300544)).
- When the 16-bit timer 0 compare match B signal is used as a trigger factor, the division ratio of the prescaler in the 16-bit timer module must not be set to MCLK/1.
- 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.
- Software controllable pull-up resistors are provided for the input ports. Disable the pull-up resistors of the ports used for analog inputs.
- 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).

LCD Controller (LCDC)

- The LCDC clock supply cannot be stopped while the LCD displays a screen. Before the LCDC clock supply can be stopped, the LCDC must enter power save mode.
- When using an STN panel, the registers for setting the HR-TFT timing parameters must be set to 0x0.
- Display addresses and positions are specified with a word boundary address or in word units, therefore the Main Window Line Address Offset Register (D[9:0]/0x301A74) must be set to a multiple of (32 bits ÷ bpp). Depending on the LCD horizontal resolution and the bpp mode selected, it may be necessary to reserve a larger image area than the LCD panel resolution and set the appropriate line address offset even if the application does not need a larger image than the LCD panel to be displayed.

For example, if the LCD width and image width are 240 pixels in 1-bpp mode,

$$\text{Line address offset} = 240 \times 1 / 32 = 7.5 \text{ [words]}$$

In this case, MWLADR[9:0] (D[9:0]/0x301A74) must be set to 8. Furthermore, the image must be prepared in 256 (8 × 32) pixels wide.

* **MWLADR[9:0]**: Main Window Line Address Offset Bits in the Main Window Line Address Offset Register (D[9:0]/0x301A74)

- When a TFT LCD panel is used with the S1C33E08 TFT LCD interface selected (TFTSEL (D31/0x301A60) = 1), the VNDPF flag (D7/0x301A04) may be fixed at 1 (vertical non-display period) in some rare cases depending on the timing. This means that the flag cannot normally indicate the vertical display and vertical non-display periods in the TFT interface.

* **TFTSEL**: HR-TFT Panel Select Bit in the LCDC Display Mode Register (D31/0x301A60)

* **VNDPF**: Vertical Display Status Flag in the Status and Power Save Configuration Register (D7/0x301A04)

How to solve this problem

For example, when the application needs to know a vertical non-display period to perform the following processing (switching the system clock in a vertical non-display period):

- (1) The system clock is 60 MHz (using the PLL) and the TFT LCD is displaying.
- (2) The program detects a vertical non-display period and switches the system clock to 12 MHz while keeping the TFT LCD displayed.
- (3) The program sets the CPU to enter HALT mode.

To realize the above processing, use the following procedure to detect a vertical non-display period:

- (1) The system clock is 60 MHz (using the PLL) and the TFT LCD is displaying.
- (2) Set the Clock Control Register of a 16-bit timer.
- (3) Set the Comparison Data B Setup Register of the 16-bit timer.
- (4) Clear the 16-bit timer compare-match B interrupt flag.
- (5) Detect the FPFFRAME signal by reading the data register of the I/O port that has been configured for the FPFFRAME output and run the 16-bit timer when the FPFFRAME signal is asserted (high level is detected when FPFFRAME is configured to high active, or low level is detected when it is configured to low active).
- (6) Detect that the 16-bit timer compare-match B interrupt flag is set (it means that a vertical non-display period is detected).
- (7) Switch the system clock to 12 MHz.
- (8) Enter HALT mode.

Use the following equation to calculate the comparison data B to be set to the 16-bit timer:

$$\text{Comparison Data B} = \frac{1}{f_{\text{pshift}}} \times ((\text{VDPS} + \text{VDP}) \times \text{HT} - \text{VPS}') \times f_{\text{t16}}$$

f_{pshift} : LCDC FPSHIFT clock frequency [Hz]

VDPS: Vertical display period start position (VDPS = VDPSCNT[9:0] [lines])

VDP: Vertical display period (VDP = VDPCNT[9:0] + 1 [lines])

HT: Horizontal total period (HT = (HTCNT[6:0] + 1) × 8 [Ts])

VPS': Vertical sync pulse start position (VPS' = FPFST[9:0] × HT + FPFSTO[9:0] [Ts])

VDPSCNT[9:0]: D[9:0]/0x301A24

VDPCNT[9:0]: D[9:0]/0x301A14

HTCNT[6:0]: D[22:16]/0x301A10

FPFST[9:0]: D[25:16]/0x301A2C

FPFSTO[9:0]: D[9:0]/0x301A30

f_{t16} : 16-bit timer \times clock frequency [Hz]

APPENDIX E SUMMARY OF PRECAUTIONS

- The LCDC does not support the frame interrupt (INTEN (D0/0x301A00), INTF (D31/0x301A04)) when TFT interface is selected (TFTSEL (D31/0x301A60) = 1).
 - * **INTF**: Frame Interrupt Flag in the Status and Power Save Configuration Register (D31/0x301A04)
 - * **INTEN**: Frame Interrupt Enable Bit in the Frame Interrupt Register (D0/0x301A00)

Appendix F Supplementary Description for Clock Control

Notes on clock control

- Use the Gated Clock Control Register 0 (0x301B00) and the Gated Clock Control Register 1 (0x301B04) to control the clock supply to the peripheral modules.
- The clocks must be supplied to operate the peripheral modules.
- The clocks are also required for accessing the control registers in the peripheral modules, in addition to operating the peripheral module.
- Be aware that the default clock supply status (supplied or not supplied) is not the same for all peripheral modules (see the Table below).

List of clock control functions

Module	Control bit name	Address : bit	Function
RTC	RTCSAPB_CKE	0x301B04 : D0	RTC SAPB bus interface clock (MCLK) supply control
DMA	DMA_CKE	0x301B04 : D1	DMA controller clock (MCLK) supply control
ITC	ITC_CKE	0x301B04 : D2	ITC clock (MCLK) supply control
ADC	ADC_CKE	0x301B04 : D3	A/D converter clock (MCLK) supply control
CARD	CARD_CKE	0x301B04 : D4	Card interface clock (MCLK) supply control
EFSIO	EFSIOSAPB_CKE	0x301B04 : D5	EFSIO SAPB bus interface clock (MCLK) supply control
SPI	SPI_CKE	0x301B04 : D6	SPI clock (MCLK) supply control
SRAMC	SRAMSAPB_CKE	0x301B04 : D7	SRAMC SAPB bus interface clock (MCLK) supply control
GPIO	GPIO_CKE	0x301B04 : D8	GPIO clock (MCLK) supply control
WDT	WDT_CKE	0x301B04 : D9	Watchdog timer clock (MCLK) supply control
DCSIO	DCSIO_CKE	0x301B04 : D10	DCSIO clock (MCLK) supply control
I ² S	I ² S_CKE	0x301B04 : D11	I ² S interface clock (MCLK) supply control
EGPIO	EGPIO_CKE	0x301B04 : D12	EGPIO clock (MCLK) supply control
T16	TM0_CKE	0x301B04 : D13	16-bit timer 0 clock (MCLK) supply control
T16	TM1_CKE	0x301B04 : D14	16-bit timer 1 clock (MCLK) supply control
T16	TM2_CKE	0x301B04 : D15	16-bit timer 2 clock (MCLK) supply control
T16	TM3_CKE	0x301B04 : D16	16-bit timer 3 clock (MCLK) supply control
T16	TM4_CKE	0x301B04 : D17	16-bit timer 4 clock (MCLK) supply control
T16	TM5_CKE	0x301B04 : D18	16-bit timer 5 clock (MCLK) supply control
LCDC	IVRAMARB_CKE	0x301B04 : D19	IVRAM arbiter clock (MCLK) supply control
MISC	MISC_HCKE	0x301B04 : D24	Misc register clock (MCLK) supply control in HALT mode
EFSIO	EFSIOBR_HCKE	0x301B04 : D25	EFSIO baud-rate timer clock (MCLK) supply control in HALT mode
SRAMC	SRAMC_HCKE	0x301B04 : D26	SRAMC clock (MCLK) supply control in HALT mode
GPIO	GPIONSTP_HCKE	0x301B04 : D27	GPIO input/interrupt circuit clock (MCLK) supply control in HALT mode
LCDC	LCDCAHB_HCKE	0x301B04 : D28	LCDC_AHB bus clock (MCLK) supply control in HALT mode
CPU	CPUAHB_HCKE	0x301B04 : D29	CPU_AHB bus clock (MCLK) supply control in HALT mode
LCDC	LCDC_CKE	0x301B00 : D0	LCDC module clock (LCDC_CLK) supply control *1
LCDC	LCDCSAPB_CKE	0x301B00 : D1	LCDC SAPB bus interface clock (MCLK) supply control *1
LCDC	LCDCAHBIF_CKE	0x301B00 : D2	LCDC_AHB bus interface clock (MCLK) supply control *1
LCDC	DSTRAM_CKE	0x301B00 : D3	Area 3 DST RAM clock (MCLK) supply control
SDRAMC	SDSAPB_CKE	0x301B00 : D4	SDRAMC SAPB bus interface clock (MCLK) supply control
SDRAMC	SDAPLCDC_CKE	0x301B00 : D5	SDRAMC LCDC_AHB bus interface clock (MCLK) supply control
SDRAMC	SDAPCPU_CKE	0x301B00 : D6	SDRAMC CPU_AHB bus interface clock (MCLK) supply control
SDRAMC	SDAPCPU_HCKE	0x301B00 : D7	SDRAMC CPU_AHB bus interface clock (MCLK) supply control in HALT mode
USB	USB_CKE	0x301B00 : D8	USB module clock (OSC3 = 48MHz) supply control
USB	USBSAPB_CKE	0x301B00 : D9	USB SAPB bus interface clock (MCLK) supply control

*1: These bits must be set to 1 when switching IVRAM to/from A0RAM (see next page).

Clock control when relocating the IVRAM (A0RAM) area

The IVRAM, which is located in Area 3 as the display memory for the LCDC by default, can be relocated to Area 0 to use it as A0RAM by setting the control bit in the LCDC module.

The LCDC module clocks must be supplied when relocating the IVRAM even if the LCDC module is not used. The following shows the procedure to relocate the IVRAM from Area 3 to Area 0.

1. Enabling clock supply to the LCDC module

Before IRAM (D0/0x301A64) can be accessed, set the control bits listed below to 1 to supply the clocks to the LCDC module.

- * **LCDC_CKE**: LCDC Main Clock Control Bit in the Gated Clock Control Register 0 (D0/0x301B00)
- * **LCDCSAPB_CKE**: LCDC SAPB Bus Interface Clock Control Bit in the Gated Clock Control Register 0 (D1/0x301B00)
- * **LCDCAHBIF_CKE**: LCDC AHB Bus Interface Clock Control Bit in the Gated Clock Control Register 0 (D2/0x301B00)

These bits are set to 1 by default.

2. Relocating Area 3 (IVRAM) to Area 0 (A0RAM)

Use the control bit IRAM (D0/0x301A64) in the LCDC module to relocate the IVRAM.

- * **IRAM**: IRAM Assignment Bit in the IRAM Select Register (D0/0x301A64)

Set IRAM (D0/0x301A64) to 1 to relocate the IVRAM to Area 0 (A0RAM).

IRAM = 0: IVRAM/Area 3 (default)

IRAM = 1: A0RAM/Area 0

3. Disabling clock supply to the LCDC module

Set the control bits listed in Step 1 to 0 to stop the clocks for the LCDC module.

4. Disabling clock supply to the IVRAM arbiter

The IVRAM arbiter is not required when the IVRAM is used as the A0RAM. Set IVRAMARB_CKE (D19/0x301B04) to 0 to stop the clock for the IVRAM arbiter.

- * **IVRAMARB_CKE**: IVRAM Arbiter Clock Control Bit in the Gated Clock Control Register 1 (D19/0x301B04)

This bit is set to 1 by default.

AMERICA

EPSON ELECTRONICS AMERICA, INC.

HEADQUARTERS

2580 Orchard Parkway
San Jose, CA 95131, U.S.A.
Phone: +1-800-228-3964 Fax: +1-408-922-0238

SALES OFFICE

Northeast

301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 Fax: +1-781-246-5443

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

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 Electronic Technology Development (Shenzhen) LTD.

12/F, Dawning Mansion, Keji South 12th Road
Hi-Tech Park, Shenzhen
Phone: +86-755-2699-3828 Fax: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.

14F, No. 7, Song Ren Road
Taipei 110
Phone: +886-2-8786-6688 Fax: +886-2-8786-6677

EPSON SINGAPORE PTE., LTD.

1 HarbourFront Place
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 Fax: +65-6271-3182

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

2F, Grand B/D, 457-4 Songjeong-dong
Gumi-City, KOREA
Phone: +82-54-454-6027 Fax: +82-54-454-6093

SEIKO EPSON CORPORATION SEMICONDUCTOR OPERATIONS DIVISION

IC Sales Dept.

IC International Sales Group

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 Fax: +81-42-587-5117

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

■ EPSON Electronic Devices Website

http://www.epson.jp/device/semicon_e