

CMOS 32-BIT SINGLE CHIP MICROCONTROLLER

S1C33L26

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 Economy, Trade and Industry or other approval from another government agency.

All brands or product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

© SEIKO EPSON CORPORATION 2010, All rights reserved.

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
 Yx : Writer software

Corresponding model number

[33L01: for S1C33L01]

Tool classification

[C: microcomputer use]

Product classification

[S5U1: development tool for semiconductor products]

– Contents –

1 Overview	1-1
1.1 Features	1-1
1.2 Block Diagram	1-8
1.3 Pin Descriptions	1-9
1.3.1 Pin Arrangement	1-9
1.3.2 Pin Functions	1-12
1.3.3 Package	1-20
1.3.4 Thermal Resistance of the Package	1-23
2 CPU	2-1
2.1 Features of the C33 PE Core	2-1
2.2 CPU Registers	2-2
2.3 Instruction Set	2-2
2.4 Debug Mode.....	2-5
2.5 Chip ID	2-6
3 Memory Map	3-1
3.1 Boot Address.....	3-2
3.2 Area 0 (IRAM, Cache Memory).....	3-3
3.2.1 IRAM	3-3
3.2.2 Cache Memory	3-3
3.3 Areas 1 and 2 (Reserved for System).....	3-3
3.4 Area 3 (IVRAM, DSTRAM).....	3-3
3.4.1 IVRAM	3-3
3.4.2 DSTRAM.....	3-3
3.5 Area 6 (I/O Area).....	3-4
3.6 External Memory Area	3-4
3.7 Bus Masters and Accessible Memories	3-4
3.8 Memory Access Rate	3-4
4 Power Supply	4-1
4.1 Power Supply Pins	4-1
4.2 Operating Voltage (LV _{DD}).....	4-2
4.3 Power Supply for PLL (PLL _{VDD} , PLL _{VSS}).....	4-2
4.4 Power Supply for RTC (RTC _{VDD})	4-2
4.5 I/O Interface Voltage (HV _{DD}).....	4-2
4.6 Power Supply for Analog Circuits (AV _{DD})	4-2
4.7 Precautions on Power Supply	4-3
5 Reset and NMI	5-1
5.1 Initial Reset	5-1
5.1.1 #RESET Pin.....	5-1
5.1.2 Resetting by the Watchdog Timer	5-1
5.1.3 Initial Reset Sequence	5-2
5.1.4 Initial Reset Status	5-2
5.1.5 Precautions to be Taken during Initial Reset.....	5-3
5.2 NMI Input	5-3
5.2.1 #NMI Pin	5-4
5.2.2 NMI by the Watchdog Timer.....	5-4

6	Clock Management Unit (CMU)	6-1
6.1	CMU Module Overview	6-1
6.2	CMU Pins	6-2
6.3	Oscillators	6-2
6.3.1	OSC3 Oscillator Circuit	6-2
6.3.2	OSC1 Oscillator Circuit	6-3
6.4	PLL	6-4
6.4.1	PLL On/Off Control	6-4
6.4.2	Selecting the PLL Input Clock	6-5
6.4.3	Setting the Frequency Multiplication Rate	6-5
6.4.4	Other PLL Settings	6-6
6.4.5	Power Supply for PLL	6-7
6.5	SSCG	6-7
6.5.1	SSCG On/Off Control	6-7
6.5.2	SS Modulation Parameter Settings	6-8
6.6	System Clock Settings	6-9
6.6.1	System Clock Source Selection	6-9
6.6.2	System Clock Frequency Setting	6-10
6.6.3	Main System Clock (MCLK) Setting	6-10
6.7	Clock Supply Control	6-11
6.7.1	Core Clock (CCLK)	6-11
6.7.2	Bus Clock (BCLK)	6-11
6.7.3	Peripheral Module Clocks (PCLK1, PCLK2)	6-12
6.7.4	GE Module Clock (GCLK)	6-12
6.7.5	LCDC Module Clock (LCLK)	6-13
6.7.6	SRAMC and SDRAMC Clock (SDCLK)	6-13
6.7.7	USB Clocks (USBCLK, USBREGCLK)	6-14
6.8	Clock External Output (CMU_CLK)	6-14
6.9	Standby Modes	6-15
6.9.1	HALT Mode	6-15
6.9.2	SLEEP Mode	6-15
6.10	Control Register Details	6-16
	Clock Source Select Register (CMU_OSCSEL)	6-16
	Oscillation Control Register (CMU_OSCCTL)	6-17
	LCDC Clock Division Ratio Select Register (CMU_LCLKDIV)	6-18
	Clock Control Register (CMU_CLKCTL)	6-19
	System Clock Division Ratio Select Register (CMU_SYSCLKDIV)	6-21
	CMU_CLK Select Register (CMU_CMUCLK)	6-22
	PLL Input Clock Division Ratio Select Register (CMU_PLLINDIV)	6-23
	PLL Control Register 0 (CMU_PLLCTL0)	6-24
	PLL Control Register 1 (CMU_PLLCTL1)	6-25
	PLL Control Register 2 (CMU_PLLCTL2)	6-26
	SSCG Macro Control Register 0 (CMU_SSCG0)	6-26
	SSCG Macro Control Register 1 (CMU_SSCG1)	6-26
	CMU Write Protect Register (CMU_PROTECT)	6-27
7	Prescaler (PSC)	7-1
7.1	PSC Module Overview	7-1
7.2	Control Register Details	7-1
	PSC Control Register (PSC_CTL)	7-1
8	Real-Time Clock (RTC)	8-1
8.1	RTC Module Overview	8-1
8.2	RTC Counters	8-2

8.3	RTC Control	8-4
8.3.1	Operating Clock Control.....	8-4
8.3.2	RTC Initial Sequence	8-4
8.3.3	12/24-hour Mode and Counter Settings.....	8-5
8.3.4	Start/Stop and Software Reset	8-5
8.3.5	Counter Hold and Busy Flag.....	8-6
8.3.6	30-second Correction	8-7
8.3.7	Counter Read	8-7
8.4	RTC Interrupts.....	8-8
8.5	WAKEUP and #STBY Pins.....	8-9
8.6	Details of Control Registers	8-11
	RTC Interrupt Status Register (RTC_INTSTAT)	8-11
	RTC Interrupt Mode Register (RTC_INTMODE)	8-12
	RTC Control 0 Register (RTC_CNTL0)	8-13
	RTC Control 1 Register (RTC_CNTL1)	8-14
	RTC Second Register (RTC_SEC).....	8-15
	RTC Minute Register (RTC_MIN).....	8-15
	RTC Hour Register (RTC_HOUR).....	8-16
	RTC Day Register (RTC_DAY).....	8-16
	RTC Month Register (RTC_MONTH).....	8-17
	RTC Year Register (RTC_YEAR).....	8-17
	RTC Days of Week Register (RTC_WEEK).....	8-18
	RTC Wakeup Configuration Register (RTC_WAKEUP).....	8-18
9	SRAM Controller (SRAMC)	9-1
9.1	SRAMC Module Overview	9-1
9.2	SRAMC Pins	9-1
9.3	SRAMC Operating Clock.....	9-2
9.4	External Memory Areas	9-2
9.4.1	Chip Enable Signals.....	9-3
9.4.2	Area Condition Settings	9-3
9.5	Connection of External Devices and Bus Operation	9-5
9.5.1	Connecting External Devices.....	9-5
9.5.2	Data Configuration in Memory	9-5
9.5.3	External Bus Operation.....	9-6
9.6	Bus Access Timing Charts	9-7
9.6.1	SRAM Read/Write Timing with No External #WAIT	9-7
9.6.2	SRAM Read/Write Timings with External #WAIT.....	9-9
9.7	Control Register Details	9-10
	#CE[7:4] Access Timing Configuration Register (SRAMC_TM47)	9-10
	#CE[10:8] Access Timing Configuration Register (SRAMC_TM810)	9-11
	#CE[10:4] Device Configuration Register (SRAMC_TYPE)	9-13
10	SDRAM Controller (SDRAMC)	10-1
10.1	SDRAMC Module Overview	10-1
10.2	SDRAMC Pins.....	10-1
10.3	SDRAM Clock and Double Frequency Mode	10-2
10.4	Configuration of SDRAM.....	10-2
10.4.1	SDRAM Area	10-2
10.4.2	SDRAM Size and Access Condition Settings	10-2
10.5	Control and Operation of SDRAM Interface	10-7
10.5.1	Initializing SDRAM	10-7
10.5.2	SDRAM Commands.....	10-9
10.5.3	SDRAM Bus Operations	10-9

CONTENTS

10.5.4	Read/Write Cycles	10-10
10.5.5	SDRAM Refresh	10-12
10.5.6	Power-Down Mode.....	10-14
10.6	Data Queue Buffer (DQB)	10-14
10.7	Control Register Details	10-15
	SDRAM Initialization Register (SDRAMC_INIT).....	10-15
	SDRAM Configuration Register (SDRAMC_CFG)	10-17
	SDRAM Refresh Register (SDRAMC_REF)	10-18
	SDRAM Application Configuration Register (SDRAMC_APP).....	10-20
10.8	Precautions	10-20
11	Cache Controller (CCU)	11-1
11.1	CCU Module Overview.....	11-1
11.2	Cache Configuration	11-2
11.3	Cache Settings and Operations	11-3
	11.3.1 Cache Enable	11-3
	11.3.2 Selecting Area to Be Cached.....	11-3
	11.3.3 Comparing Addresses and Cache Hit/Mishit	11-3
	11.3.4 Reading Operation.....	11-4
	11.3.5 Writing Operation	11-5
	11.3.6 Flush	11-5
11.4	Cache Lock with Interrupt Level Specified	11-5
11.5	Caching Operation during Debugging.....	11-6
11.6	Cache Data Integrity	11-6
11.7	Control Register Details	11-6
	Cache Configuration Register (CCU_CFG).....	11-6
	Cacheable Area Select Register (CCU_AREA)	11-7
	Cache Lock Register (CCU_LK).....	11-8
	Cache Status Register (CCU_STAT)	11-8
	Cache Write Buffer Status Register (CCU_WB_STAT)	11-9
	CCLK Division Ratio Select Register (CCU_CCLKDV)	11-10
12	Interrupt Controller (ITC)	12-1
12.1	ITC Module Overview.....	12-1
12.2	Vector Table	12-2
12.3	Control of Maskable Interrupts	12-3
	12.3.1 Interrupt Control Bits in Peripheral Modules	12-3
	12.3.2 ITC Interrupt Request Processing	12-3
	12.3.3 Interrupt Processing by the C33 PE Core.....	12-4
12.4	NMI.....	12-5
12.5	Software Exception	12-5
12.6	HALT and SLEEP Mode Cancellation	12-5
12.7	Control Register Details	12-6
	Interrupt Level Registers (ITC_XXX_LV).....	12-6
13	DMA Controller (DMAC)	13-1
13.1	DMAC Module Overview	13-1
13.2	DMAC Operating Clock	13-1
13.3	Programming Control Information	13-2
	13.3.1 Setting the Base Address	13-2
	13.3.2 Control Information	13-2
	13.3.3 Auto-Reload Data	13-6
13.4	DMAC Invocation	13-6
13.5	Operation of DMAC	13-8

13.5.1	Single Transfer Mode	13-8
13.5.2	Successive Transfer Mode	13-9
13.6	DMAC Interrupt	13-10
13.7	Control Register Details	13-11
DMAC Channel Enable Register (DMAC_CH_EN)		13-11
DMAC Control Table Base Address Register (DMAC_TBL_BASE)		13-11
DMAC Interrupt Enable Register (DMAC_IE)		13-12
DMAC Trigger Select Register (DMAC_TRG_SEL)		13-13
DMAC Trigger Flag Register (DMAC_TRG_FLG)		13-14
DMAC End-of-Transfer Flag Register (DMAC_END_FLG)		13-15
DMAC Running Status Register (DMAC_RUN_STAT)		13-15
DMAC Pause Status Register (DMAC_PAUSE_STAT)		13-16
14	8-bit Timers (T8)	14-1
14.1	T8 Module Overview	14-1
14.2	Count Clock	14-2
14.3	Count Mode	14-2
14.4	Reload Data Register and Underflow Cycle	14-2
14.5	Timer Reset	14-3
14.6	RUN/STOP Control	14-3
14.7	T8 Output Signals	14-4
14.8	Fine Mode (Ch.0 to Ch.3)	14-4
14.9	T8 Interrupts	14-5
14.10	Control Register Details	14-6
T8 Ch.x Input Clock Select Registers (T8_CLKx)		14-7
T8 Ch.x Reload Data Registers (T8_TRx)		14-7
T8 Ch.x Counter Data Registers (T8_TCx)		14-8
T8 Ch.x Control Registers (T8_CTLx)		14-8
T8 Ch.x Interrupt Control Registers (T8_INTx)		14-9
15	16-bit PWM Timer (T16A5)	15-1
15.1	T16A5 Module Overview	15-1
15.2	T16A5 Input/Output Pins	15-2
15.3	Count Clock	15-2
15.4	T16A5 Operating Modes	15-3
15.4.1	Comparator Mode and Capture Mode	15-3
15.4.2	Repeat Mode and One-Shot Mode	15-4
15.5	Counter Control	15-4
15.5.1	Counter Reset	15-4
15.5.2	Counter RUN/STOP Control	15-5
15.5.3	Reading Counter Values	15-5
15.5.4	Timing Charts	15-5
15.6	Timer Output Control	15-6
15.7	T16A5 Interrupts and DMA	15-7
15.7.1	Interrupts	15-7
15.7.2	DMA Transfer	15-9
15.8	Control Register Details	15-9
T16A5 Ch.x Counter Control Registers (T16A_CTLx)		15-10
T16A5 Ch.x Counter Data Registers (T16A_TCx)		15-12
T16A5 Ch.x Comparator/Capture Control Registers (T16A_CCCTLx)		15-13
T16A5 Ch.x Comparator/Capture A Data Registers (T16A_CCAx)		15-15
T16A5 Ch.x Comparator/Capture B Data Registers (T16A_CCBx)		15-15
T16A5 Ch.x Comparator/Capture Interrupt Enable Registers (T16A_IENx)		15-16
T16A5 Ch.x Comparator/Capture Interrupt Flag Registers (T16A_IFLGx)		15-16

16 16-bit Audio PWM Timer (T16P)	16-1
16.1 T16P Module Overview	16-1
16.2 T16P Input/Output Pins	16-2
16.3 Setting T16P Operating Conditions	16-2
16.3.1 Count Clock	16-2
16.3.2 PCM Data Configuration	16-3
16.3.3 Operating Mode Selection	16-3
16.3.4 PWM Output Condition Settings	16-4
16.4 Control and T16P Operations.....	16-5
16.4.1 Resetting T16P	16-5
16.4.2 Run/Stop Control	16-5
16.4.3 Setting Compare Data	16-5
16.4.4 Volume Control.....	16-6
16.4.5 Counter Value	16-7
16.4.6 Timing Charts.....	16-7
16.5 T16P Interrupts and DMA.....	16-9
16.5.1 Interrupts.....	16-9
16.5.2 DMA Transfer	16-10
16.6 Control Register Details	16-10
T16P Compare A Buffer Register (T16P_A)	16-10
T16P Compare B Buffer Register (T16P_B)	16-11
T16P Counter Data Register (T16P_CNT_DATA)	16-11
T16P Volume Control Register (T16P_VOL_CTL)	16-11
T16P Control Register (T16P_CTL)	16-12
T16P Running Control Register (T16P_RUN).....	16-14
T16P Internal Clock Control Register (T16P_CLK).....	16-15
T16P Interrupt Control Register (T16P_INT).....	16-15
17 Watchdog Timer (WDT)	17-1
17.1 WDT Module Overview	17-1
17.2 WDT Input/Output Pins	17-1
17.3 WDT Operating Clock	17-2
17.4 Control of the Watchdog Timer.....	17-2
17.4.1 Setting Up the Watchdog Timer	17-2
17.4.2 Starting/Stopping the Watchdog Timer	17-3
17.4.3 Resetting the Watchdog Timer	17-3
17.4.4 Operation in Standby Mode	17-3
17.4.5 Clock Output of the Watchdog Timer	17-3
17.4.6 External NMI Output	17-4
17.5 Control Register Details	17-4
WDT Write Protect Register (WD_PROTECT).....	17-4
WDT Enable and Setup Register (WD_EN)	17-5
WDT Comparison Data L/H Registers (WD_CMP_L, WD_CMP_H).....	17-6
WDT Count Data L/H Registers (WD_CNT_L, WD_CNT_H).....	17-6
WDT Control Register (WD_CTL)	17-6
18 Universal Serial Interface (USI)	18-1
18.1 USI Module Overview	18-1
18.2 USI Pins	18-2
18.3 USI Clock Sources	18-2
18.4 USI Module Settings	18-4
18.4.1 USI Module Software Reset.....	18-4
18.4.2 Interface Mode.....	18-4
18.4.3 General Settings for All Interface Modes	18-4

18.4.4	Settings for UART Mode	18-5
18.4.5	Settings for SPI Mode	18-5
18.4.6	Settings for I ² C Mode	18-6
18.5	Data Transfer Control	18-7
18.5.1	Data Transfer in UART Mode	18-7
18.5.2	Data Transfer in SPI Mode	18-8
18.5.3	Data Transfer in I ² C Mode	18-10
18.6	Receive Errors.....	18-19
18.7	USI Interrupts and DMA	18-20
18.7.1	Interrupts in UART Mode	18-20
18.7.2	Interrupts in SPI Mode	18-21
18.7.3	Interrupts in I ² C Master Mode.....	18-22
18.7.4	Interrupts in I ² C Slave Mode	18-22
18.7.5	DMA Transfer	18-23
18.8	Control Register Details	18-23
USI Global Configuration Register (USI_GCFG).....		18-24
USI Transmit Data Buffer Register (USI_TD)		18-24
USI Receive Data Buffer Register (USI_RD).....		18-25
USI UART Mode Configuration Register (USI_UCFG).....		18-25
USI UART Mode Interrupt Enable Register (USI_UIE).....		18-26
USI UART Mode Interrupt Flag Register (USI_UIF)		18-26
USI SPI Master/Slave Mode Configuration Register (USI_SCFG).....		18-28
USI SPI Master/Slave Mode Interrupt Enable Register (USI_SIE).....		18-29
USI SPI Master/Slave Mode Interrupt Flag Register (USI_SIF)		18-30
USI I ² C Master Mode Trigger Register (USI_IMTG)		18-31
USI I ² C Master Mode Interrupt Enable Register (USI_IMIE).....		18-32
USI I ² C Master Mode Interrupt Flag Register (USI_IMIF)		18-32
USI I ² C Slave Mode Trigger Register (USI_ISTG).....		18-33
USI I ² C Slave Mode Interrupt Enable Register (USI_ISIE).....		18-34
USI I ² C Slave Mode Interrupt Flag Register (USI_ISIF).....		18-35
18.9	Precautions	18-36
19	Universal Serial Interface with LCD Interface (USIL)	19-1
19.1	USIL Module Overview	19-1
19.2	USIL Pins	19-2
19.3	USIL Clock Sources	19-3
19.4	USIL Module Settings	19-5
19.4.1	USIL Module Software Reset.....	19-5
19.4.2	Interface Mode	19-5
19.4.3	General Settings for All Interface Modes	19-5
19.4.4	Settings for UART Mode	19-5
19.4.5	Settings for SPI Mode	19-6
19.4.6	Settings for I ² C Mode	19-7
19.4.7	Settings for LCD SPI Mode	19-7
19.4.8	Settings for LCD Parallel Mode	19-9
19.5	Data Transfer Control	19-10
19.5.1	Data Transfer in UART Mode	19-10
19.5.2	Data Transfer in SPI Mode	19-11
19.5.3	Data Transfer in I ² C Mode.....	19-13
19.5.4	Data Transmission in LCD SPI Mode.....	19-23
19.5.5	Data Transfer in LCD Parallel Mode.....	19-24
19.6	Receive Errors.....	19-27
19.7	USIL Interrupts and DMA	19-28
19.7.1	Interrupts in UART Mode	19-28

19.7.2	Interrupts in SPI Mode	19-28
19.7.3	Interrupts in I ² C Master Mode	19-29
19.7.4	Interrupts in I ² C Slave Mode	19-30
19.7.5	Interrupts in LCD SPI Mode	19-31
19.7.6	Interrupts in LCD Parallel Mode	19-31
19.7.7	DMA Transfer	19-31
19.8	Control Register Details	19-32
	USIL Global Configuration Register (USIL_GCFG)	19-32
	USIL Transmit Data Buffer Register (USIL_TD)	19-33
	USIL Receive Data Buffer Register (USIL_RD)	19-33
	USIL UART Mode Configuration Register (USIL_UCFG)	19-33
	USIL UART Mode Interrupt Enable Register (USIL_UIE)	19-34
	USIL UART Mode Interrupt Flag Register (USIL_UIF)	19-35
	USIL SPI Master/Slave Mode Configuration Register (USIL_SCFG)	19-36
	USIL SPI Master/Slave Mode Interrupt Enable Register (USIL_SIE)	19-37
	USIL SPI Master/Slave Mode Interrupt Flag Register (USIL_SIF)	19-38
	USIL I ² C Master Mode Trigger Register (USIL_IMTG)	19-39
	USIL I ² C Master Mode Interrupt Enable Register (USIL_IMIE)	19-40
	USIL I ² C Master Mode Interrupt Flag Register (USIL_IMIF)	19-40
	USIL I ² C Slave Mode Trigger Register (USIL_ISTG)	19-41
	USIL I ² C Slave Mode Interrupt Enable Register (USIL_ISIE)	19-42
	USIL I ² C Slave Mode Interrupt Flag Register (USIL_ISIF)	19-43
	USIL LCD SPI Mode Configuration Register (USIL_LSCFG)	19-44
	USIL LCD SPI Mode Interrupt Enable Register (USIL_LSIE)	19-45
	USIL LCD SPI Mode Interrupt Flag Register (USIL_LSIF)	19-45
	USIL LCD SPI Mode Data Configuration Register (USIL_LSDCFG)	19-46
	USIL LCD Parallel I/F Mode Configuration Register (USIL_LPCFG)	19-47
	USIL LCD Parallel I/F Mode Interrupt Enable Register (USIL_LPIE)	19-48
	USIL LCD Parallel I/F Mode Interrupt Flag Register (USIL_LPIF)	19-49
	USIL LCD Parallel I/F Mode Access Timing Register (USIL_LPAC)	19-49
19.9	Precautions	19-51
20	General-Purpose Serial Interface (FSIO)	20-1
20.1	FSIO Module Overview	20-1
20.2	FSIO Pins	20-2
20.3	FSIO Operating Clock	20-2
20.4	Mode Settings	20-2
	20.4.1 Interface Mode and Transfer Mode	20-2
	20.4.2 Standard Mode and Advanced Mode	20-3
20.5	Baud-Rate Timer (Baud Rate Setting)	20-3
20.6	Clock-Synchronized Interface	20-4
	20.6.1 Outline of Clock-Synchronized Interface	20-4
	20.6.2 Setting Clock-Synchronized Interface	20-5
	20.6.3 Control and Operation of Clock-Synchronized Transfer	20-6
20.7	Asynchronous Interface	20-12
	20.7.1 Outline of Asynchronous Interface	20-12
	20.7.2 Setting Asynchronous Interface	20-13
	20.7.3 Control and Operation of Asynchronous Transfer	20-15
20.8	IrDA Interface	20-18
	20.8.1 Outline of IrDA Interface	20-18
	20.8.2 Setting IrDA Interface	20-19
	20.8.3 Control and Operation of IrDA Interface	20-20
20.9	FSIO Interrupts and DMA	20-21
	20.9.1 Interrupts	20-21
	20.9.2 DMA Transfer	20-22

20.10 Control Register Details	20-22
FSIO Ch.x Transmit Data Registers (FSIO_TXDx)	20-23
FSIO Ch.x Receive Data Registers (FSIO_RXDx)	20-23
FSIO Ch.x Status Registers (FSIO_STATUSx)	20-23
FSIO Ch.x Control Registers (FSIO_CTLx)	20-25
FSIO Ch.x IrDA Registers (FSIO_IRDAx)	20-26
FSIO Ch.x Baud-rate Timer Control Registers (FSIO_BRTRUNx)	20-28
FSIO Ch.x Baud-rate Timer Reload Data L Registers (FSIO_BRTRDLx)	20-28
FSIO Ch.x Baud-rate Timer Reload Data H Registers (FSIO_BRTRDHx)	20-28
FSIO Ch.x Baud-rate Timer Count Data L Registers (FSIO_BRTCDLx)	20-29
FSIO Ch.x Baud-rate Timer Count Data H Registers (FSIO_BRTCDHx)	20-29
FSIO Ch.x Interrupt Flag Registers (FSIO_INTFx)	20-29
FSIO Ch.x Interrupt Enable Registers (FSIO_INTEx)	20-30
FSIO Ch.x STD/ADV Mode Select Registers (FSIO_ADVx)	20-30
21 I²S.....	21-1
21.1 I ² S Module Overview	21-1
21.2 I ² S Output Pins	21-1
21.3 I ² S Module Operating Clock	21-2
21.4 Setting the I ² S Module	21-2
21.5 Data Output Control	21-6
21.6 I ² S Interrupt and DMA	21-11
21.6.1 Interrupts	21-11
21.6.2 DMA Transfer	21-12
21.7 Control Register Details	21-12
I ² S Control Register (I2S_CTL)	21-13
I ² S Master Clock Division Ratio Register (I2S_DV_MCLK)	21-16
I ² S Audio Clock Division Ratio Register (I2S_DV_AUDIO_CLK)	21-16
I ² S Start/Stop Register (I2S_START)	21-18
I ² S FIFO Status Register (I2S_FIFO_STAT)	21-18
I ² S Interrupt Control Register (I2S_INT)	21-19
I ² S FIFO Register (I2S_FIFO)	21-21
21.8 Setting the I ² S Clocks	21-22
22 Remote Controller (REMC)	22-1
22.1 REMC Module Overview	22-1
22.2 REMC Input/Output Pins	22-1
22.3 Carrier Generation	22-1
22.4 Data Length Counter Clock Settings	22-2
22.5 Data Transfer Control	22-3
22.6 REMC Interrupts	22-5
22.7 Control Register Details	22-6
REMC Configuration Register (REMC_CFG)	22-6
REMC Carrier Length Setup Register (REMC_CAR)	22-7
REMC Length Counter Register (REMC_LCNT)	22-8
REMC Interrupt Control Register (REMC_INT)	22-8
23 Card Interface (CARD).....	23-1
23.1 CARD Module Overview	23-1
23.2 CARD Output Pins	23-1
23.3 Card Interface Control Signals	23-1
24 I/O Ports (GPIO)	24-1
24.1 GPIO Module Overview	24-1
24.2 Input/Output Pin Function Selection (Port MUX)	24-2

24.3 Data Input/Output.....	24-3
24.4 Pull-up Control	24-4
24.5 Port Input Interrupt and DMA	24-4
24.6 Input Port Noise Filters.....	24-9
24.7 Bus Drive Control	24-9
24.8 Control Register Details	24-9
Px Port Data Registers (GPIO_Px_DAT).....	24-11
Px Port I/O Control Registers (GPIO_Px_IOC)	24-11
Bus Drive Control Register (GPIO_BUS_DRV).....	24-12
Px Port Pull-up Control Registers (GPIO_Px_PUP).....	24-12
FPT0–3 Interrupt Port Select Register (GPIO_FPT03_SEL)	24-13
FPT4–7 Interrupt Port Select Register (GPIO_FPT47_SEL)	24-13
FPT8–B Interrupt Port Select Register (GPIO_FPT8B_SEL).....	24-14
FPTC–F Interrupt Port Select Register (GPIO_FPTCF_SEL).....	24-15
FPT0–3 Interrupt Polarity Select Register (GPIO_FPT03_POL).....	24-15
FPT4–7 Interrupt Polarity Select Register (GPIO_FPT47_POL).....	24-16
FPT8–B Interrupt Polarity Select Register (GPIO_FPT8B_POL).....	24-16
FPTC–F Interrupt Polarity Select Register (GPIO_FPTCF_POL).....	24-16
FPT0–3 Interrupt Mode Select Register (GPIO_FPT03_MOD)	24-16
FPT4–7 Interrupt Mode Select Register (GPIO_FPT47_MOD)	24-17
FPT8–B Interrupt Mode Select Register (GPIO_FPT8B_MOD)	24-17
FPTC–F Interrupt Mode Select Register (GPIO_FPTCF_MOD).....	24-17
FPT0–3 Interrupt Mask Register (GPIO_FPT03_MSK)	24-18
FPT4–7 Interrupt Mask Register (GPIO_FPT47_MSK)	24-18
FPT8–B Interrupt Mask Register (GPIO_FPT8B_MSK)	24-18
FPTC–F Interrupt Mask Register (GPIO_FPTCF_MSK).....	24-18
FPT0–3 Interrupt Flag Register (GPIO_FPT03_FLG).....	24-19
FPT4–7 Interrupt Flag Register (GPIO_FPT47_FLG).....	24-19
FPT8–B Interrupt Flag Register (GPIO_FPT8B_FLG)	24-19
FPTC–F Interrupt Flag Register (GPIO_FPTCF_FLG).....	24-20
FPT0–1 Interrupt Chattering Filter Control Register (GPIO_FPT01_CHAT)	24-20
FPT2–3 Interrupt Chattering Filter Control Register (GPIO_FPT23_CHAT)	24-21
FPT4–5 Interrupt Chattering Filter Control Register (GPIO_FPT45_CHAT)	24-22
FPT6–7 Interrupt Chattering Filter Control Register (GPIO_FPT67_CHAT)	24-22
FPT8–9 Interrupt Chattering Filter Control Register (GPIO_FPT89_CHAT)	24-23
FPTA–B Interrupt Chattering Filter Control Register (GPIO_FPTAB_CHAT)	24-23
FPTC–D Interrupt Chattering Filter Control Register (GPIO_FPTCD_CHAT).....	24-24
FPTE–F Interrupt Chattering Filter Control Register (GPIO_FPTEF_CHAT).....	24-24
Port DMA Trigger Source Select Register (GPIO_DMA).....	24-25
P0[3:0] Port Function Select Register (PMUX_P0_03)	24-26
P0[7:4] Port Function Select Register (PMUX_P0_47)	24-27
P1[3:0] Port Function Select Register (PMUX_P1_03)	24-28
P1[7:4] Port Function Select Register (PMUX_P1_47)	24-29
P2[1:0] Port Function Select Register (PMUX_P2_01)	24-29
P3[3:0] Port Function Select Register (PMUX_P3_03)	24-30
P3[6:4] Port Function Select Register (PMUX_P3_46)	24-31
P4[2:0] Port Function Select Register (PMUX_P4_02)	24-31
P5[3:0] Port Function Select Register (PMUX_P5_03)	24-32
P5[6:4] Port Function Select Register (PMUX_P5_46)	24-33
P60 Port Function Select Register (PMUX_P6_0)	24-33
P7[3:0] Port Function Select Register (PMUX_P7_03)	24-34
P7[5:4] Port Function Select Register (PMUX_P7_45)	24-34
P8[3:0] Port Function Select Register (PMUX_P8_03)	24-35
P9[3:0] Port Function Select Register (PMUX_P9_03)	24-36
P9[7:4] Port Function Select Register (PMUX_P9_47)	24-37
PA[3:0] Port Function Select Register (PMUX_PA_03)	24-38
PA[6:4] Port Function Select Register (PMUX_PA_46)	24-38
PB[3:0] Port Function Select Register (PMUX_PB_03)	24-39

PB[7:4] Port Function Select Register (PMUX_PB_47)	24-40
PC[3:0] Port Function Select Register (PMUX_PC_03)	24-41
PC[7:4] Port Function Select Register (PMUX_PC_47)	24-42
Port Noise Filter Control Register (GPIO_FILTER)	24-42
GPIO/PMUX Write Protect Register (GPIO_PROTECT)	24-43
25 A/D Converter (ADC10)	25-1
25.1 ADC10 Module Overview	25-1
25.2 ADC10 Input Pins	25-2
25.3 A/D Converter Settings	25-2
25.3.1 A/D Conversion Clock Setting	25-2
25.3.2 Selecting A/D Conversion Start and End Channels	25-3
25.3.3 A/D Conversion Mode Setting	25-4
25.3.4 Trigger Selection	25-4
25.3.5 Sampling Time Setting	25-4
25.3.6 Setting Conversion Result Storing Mode	25-5
25.4 A/D Conversion Control and Operations	25-5
25.4.1 Activating A/D Converter	25-5
25.4.2 Starting A/D Conversion	25-5
25.4.3 Reading A/D Conversion Results	25-5
25.4.4 Terminating A/D Conversion	25-6
25.4.5 Timing Charts	25-6
25.5 A/D Converter Interrupts and DMA	25-8
25.6 Control Register Details	25-8
A/D Conversion Result Register (ADC10_ADD)	25-9
A/D Trigger/Channel Select Register (ADC10_TRG)	25-9
A/D Control/Status Register (ADC10_CTL)	25-11
A/D Clock Control Register (ADC10_CLK)	25-13
26 LCD Controller (LCDC)	26-1
26.1 LCDC Module Overview	26-1
26.2 Block Diagram	26-2
26.3 LCDC Output Pins	26-3
26.4 System Settings	26-5
26.4.1 Configuration of Display Data Memory (VRAM)	26-5
26.4.2 Setting the LCDC Clock	26-6
26.5 Setting the LCD Panel	26-7
26.5.1 Types of Panels	26-7
26.5.2 STN Panel Timing Parameters	26-7
26.5.3 HR-TFT Panel Timing Parameters	26-11
26.5.4 Display Modes	26-14
26.5.5 VRAM Data Format	26-15
26.5.6 LUT Bypass Mode	26-20
26.5.7 Look-up Tables	26-26
26.5.8 Frame Rates	26-28
26.5.9 Other Settings	26-28
26.6 Display Control	26-29
26.6.1 Controlling LCD Power Up/Down	26-29
26.6.2 Main Window Display Start Address and Virtual Screen Settings	26-29
26.6.3 Writing Display Data	26-31
26.6.4 Inverting and Blanking the Display	26-31
26.6.5 Picture-in-Picture Plus and Sub-Window	26-31
26.7 LCDC Interrupt	26-34
26.8 Power Save	26-34

26.9	Reload Functions	26-35
26.10	Control Register Details	26-36
	LCDC Interrupt Enable Register (LCDC_INT)	26-37
	Status and Power Save Configuration Register (LCDC_PSAVE)	26-37
	Horizontal Display Register (LCDC_HDISP)	26-38
	Vertical Display Register (LCDC_VDISP)	26-38
	MOD Rate Register (LCDC_MODR)	26-39
	Horizontal Display Start Position Register (LCDC_HDPS)	26-39
	Vertical Display Start Position Register (LCDC_VDPS)	26-39
	FPLINE Pulse Setting Register (LCDC_FPLINE)	26-40
	FPFRAME Pulse Setting Register (LCDC_FPFR)	26-40
	FPFRAME Pulse Offset Register (LCDC_FPFROFS)	26-41
	TFT Special Output Register (LCDC_TFTSO)	26-41
	TFT_CTL1 Pulse Register (LCDC_TFT_CTL1)	26-42
	TFT_CTL0 Pulse Register (LCDC_TFT_CTL0)	26-43
	TFT_CTL2 Register (LCDC_TFT_CTL2)	26-43
	LCDC Reload Control Register (LCDC_RLDCTL)	26-43
	LCDC Reload Table Base Address Register (LCDC_RLDADR)	26-44
	LCDC Display Mode Register (LCDC_DISPMOD)	26-45
	Main Window Display Start Address Register (LCDC_MAINADR)	26-47
	Main Screen Address Offset Register (LCDC_MAINOFS)	26-47
	Sub-window Display Start Address Register (LCDC_SUBADR)	26-47
	Sub-Screen Address Offset Register (LCDC_SUBOFS)	26-47
	Sub-Window Start Position Register (LCDC_SUBSP)	26-48
	Sub-Window End Position Register (LCDC_SUBEP)	26-48
	Monochrome Look-up Table Registers 0 and 1 (LCDC_MLUT0/1)	26-49
27	Graphics Engine (GE)	27-1
27.1	GE Module Overview	27-1
27.2	Operating Clock	27-2
27.3	Drawing Functions	27-2
	27.3.1 Drawing Area	27-2
	27.3.2 Drawing Basic Objects	27-5
	27.3.3 Drawing Characters and Symbols	27-9
	27.3.4 Decompression/Copy Functions	27-10
	27.3.5 Drawing Effects	27-14
27.4	Controlling GE	27-20
	27.4.1 Initial Settings	27-20
	27.4.2 Command Execution and Termination	27-20
	27.4.3 Status Check During Execution	27-21
	27.4.4 Updated Range in Work Area	27-22
	27.4.5 Errors	27-22
	27.4.6 GE Reset	27-24
	27.4.7 GE Interrupts	27-25
27.5	Command Details	27-26
	27.5.1 Command Format	27-26
	27.5.2 List of Commands	27-27
	27.5.3 STOP1–4 Commands (Stop)	27-27
	27.5.4 GE_CONFIG Command (GE Configuration)	27-28
	27.5.5 DOT Command (Dot Drawing)	27-28
	27.5.6 LINE Command (Straight Line Drawing)	27-30
	27.5.7 RECT Command (Rectangle Drawing)	27-31
	27.5.8 TRI_FILL Command (Solid Filled Triangle Drawing)	27-33
	27.5.9 RECT_FILL Command (Solid Filled Rectangle Drawing)	27-35
	27.5.10 QUAD_FILL Command (Solid Filled Quadrilateral Drawing)	27-37
	27.5.11 CIRCLE Command (Circle Drawing)	27-39

27.5.12	CIRCLE_FILL Command (Solid Filled Circle Drawing)	27-41
27.5.13	FONT_CFG Command (Font Configuration)	27-42
27.5.14	CHAR Command (Character Drawing)	27-43
27.5.15	DECOMP Command (Decompression and Drawing)	27-45
27.5.16	COPY Command (Copy)	27-47
27.5.17	BLKCOPY Command (Block Transfer)	27-49
27.6	Control Register Details	27-51
	GE Command Address Register (GE_CMD_ADDR)	27-51
	GE Control Register (GE_CTL)	27-52
	GE Interrupt Enable Register (GE_IE)	27-53
	GE Interrupt Flag Register 1 (GE_IF1)	27-54
	GE Interrupt Flag Register 2 (GE_IF2)	27-55
	GE Interrupt Flag Register 3 (GE_IF3)	27-57
	VRAM Work Area Width Register (GE_REAL_W)	27-57
	VRAM Work Area Start Address Register (GE_WK_ADDR)	27-57
	VRAM Work Area Size Register (GE_WK_SIZE)	27-58
	Display Configuration Register (GE_DISP_CFG)	27-58
	VRAM Rotation Control Register (GE_ROTATE)	27-60
	Clipping Area Start Position Register (GE_CLIP_ST)	27-60
	Clipping Area End Position Register (GE_CLIP_END)	27-61
	Mesh Configuration Register (GE_MESH)	27-61
	Transparent Color Register (GE_MAGIC)	27-62
	Updated Area Start Position Register (GE_UPDT_ST)	27-62
	Updated Area End Position Register (GE_UPDT_END)	27-63
	Palette 1 (GE_PALETTE1)	27-63
	CCT1 4-bit Entries (GE_CCT1_4BIT)	27-63
	CCT1 2-bit Entries (GE_CCT1_2BIT)	27-64
	CCT1 1-bit Entries (GE_CCT1_1BIT)	27-64
27.7	LCDC Settings	27-64
28	USB Function Controller (USB)	28-1
28.1	USB Function Controller Overview	28-1
28.2	Pins for the USB Interface	28-2
28.3	USB Operating Clocks	28-2
28.4	Settings in Other Modules	28-2
28.5	Functional Description	28-2
	28.5.1 USB Control	28-3
	28.5.2 FIFO Management	28-16
	28.5.3 Port Interface	28-18
	28.5.4 Snooze	28-22
28.6	Registers	28-23
	28.6.1 List of Registers	28-23
	28.6.2 Detailed Description of Registers	28-26
	MainIntStat (Main Interrupt Status)	28-26
	SIE_IntStat (SIE Interrupt Status)	28-27
	EPrIntStat (EPr Interrupt Status)	28-28
	DMA_IntStat (DMA Interrupt Status)	28-28
	FIFO_IntStat (FIFO Interrupt Status)	28-28
	EP0IntStat (EP0 Interrupt Status)	28-29
	EPaIntStat (EPa Interrupt Status)	28-30
	EPbIntStat (EPb Interrupt Status)	28-30
	EPcIntStat (EPc Interrupt Status)	28-31
	EPdIntStat (EPd Interrupt Status)	28-32
	MainIntEnb (Main Interrupt Enable)	28-32
	SIE_IntEnb (SIE Interrupt Enable)	28-33
	EPrIntEnb (EPr Interrupt Enable)	28-33

DMA_IntEnb (DMA Interrupt Enable)	28-33
FIFO_IntEnb (FIFO Interrupt Enable).....	28-33
EP0IntEnb (EP0 Interrupt Enable).....	28-33
EPaIntEnb (EPa Interrupt Enable).....	28-34
EPbIntEnb (EPb Interrupt Enable).....	28-34
EPcIntEnb (EPc Interrupt Enable).....	28-34
EPdIntEnb (EPd Interrupt Enable).....	28-34
RevisionNum (Revision Number)	28-34
USB_Control (USB Control)	28-35
USB_Status (USB Status).....	28-35
XcvrControl (Xcvr Control).....	28-36
USB_Test (USB Test)	28-36
EPnControl (Endpoint Control).....	28-37
EPrFIFO_Clr (EPr FIFO Clear).....	28-38
FrameNumber_H (Frame Number HIGH).....	28-39
FrameNumber_L (Frame Number LOW)	28-39
EP0Setup_0 (EP0 Setup 0)–EP0Setup_7 (EP0 Setup 7)	28-39
USB_Address (USB Address).....	28-40
EP0Control (EP0 Control)	28-40
EP0ControlIN (EP0 Control IN)	28-41
EP0ControlOUT (EP0 Control OUT)	28-42
EP0MaxSize (EP0 Max Packet Size)	28-42
EPaControl (EPa Control).....	28-43
EPbControl (EPb Control)	28-44
EPcControl (EPc Control).....	28-44
EPdControl (EPd Control)	28-45
EPaMaxSize_H (EPa Max Packet Size HIGH)	28-46
EPaMaxSize_L (EPa Max Packet Size LOW).....	28-46
EPaConfig_0 (EPa Configuration 0).....	28-47
EPaConfig_1 (EPa Configuration 1)	28-47
EPbMaxSize_H (EPb Max Packet Size HIGH).....	28-47
EPbMaxSize_L (EPb Max Packet Size LOW)	28-47
EPbConfig_0 (EPb Configuration 0).....	28-48
EPbConfig_1 (EPb Configuration 1).....	28-48
EPcMaxSize_H (EPc Max Packet Size HIGH)	28-49
EPcMaxSize_L (EPc Max Packet Size LOW).....	28-49
EPcConfig_0 (EPc Configuration 0)	28-49
EPcConfig_1 (EPc Configuration 1)	28-49
EPdMaxSize_H (EPd Max Packet Size HIGH).....	28-50
EPdMaxSize_L (EPd Max Packet Size LOW)	28-50
EPdConfig_0 (EPd Configuration 0).....	28-50
EPdConfig_1 (EPd Configuration 1).....	28-50
EPaStartAdrs_H (EPa FIFO Start Address HIGH).....	28-51
EPaStartAdrs_L (EPa FIFO Start Address LOW)	28-51
EPbStartAdrs_H (EPb FIFO Start Address HIGH).....	28-51
EPbStartAdrs_L (EPb FIFO Start Address LOW)	28-51
EPcStartAdrs_H (EPc FIFO Start Address HIGH)	28-52
EPcStartAdrs_L (EPc FIFO Start Address LOW).....	28-52
EPdStartAdrs_H (EPd FIFO Start Address HIGH).....	28-52
EPdStartAdrs_L (EPd FIFO Start Address LOW)	28-52
CPU_JoinRd (CPU Join FIFO Read)	28-53
CPU_JoinWr (CPU Join FIFO Write).....	28-53
EnEPnFIFO_Access (EPn FIFO Access Enable)	28-54
EPnFIFOforCPU (EPn FIFO for CPU).....	28-54
EPnRdRemain_H (EPn FIFO Read Remain HIGH).....	28-55
EPnRdRemain_L (EPn FIFO Read Remain LOW)	28-55
EPnWrRemain_H (EPn FIFO Write Remain HIGH)	28-55
EPnWrRemain_L (EPn FIFO Write Remain LOW).....	28-55
DescAdrs_H (Descriptor Address HIGH)	28-55

DescAdrs_L (Descriptor Address LOW).....	28-55
DescSize_H (Descriptor Size HIGH).....	28-56
DescSize_L (Descriptor Size LOW).....	28-56
DescDoor (Descriptor Door).....	28-56
DMA_FIFO_Control (DMA FIFO Control).....	28-57
DMA_Join (DMA Join FIFO).....	28-57
DMA_Control (DMA Control).....	28-57
DMA_Config_0 (DMA Configuration 0).....	28-58
DMA_Config_1 (DMA Configuration 1).....	28-58
DMA_Latency (DMA Latency).....	28-60
DMA_Remain_H (DMA FIFO Remain HIGH).....	28-60
DMA_Remain_L (DMA FIFO Remain LOW).....	28-60
DMA_Count_HH (DMA Transfer Byte Counter HIGH/HIGH).....	28-60
DMA_Count_HL (DMA Transfer Byte Counter HIGH/LOW).....	28-60
DMA_Count_LH (DMA Transfer Byte Counter LOW/HIGH).....	28-60
DMA_Count_LL (DMA Transfer Byte Counter LOW/LOW).....	28-60
29 Misc Registers (MISC).....	29-1
29.1 RTC Wait Control.....	29-1
29.2 Internal RAM Wait Control.....	29-1
29.3 USB Settings.....	29-2
29.3.1 USB Wait Control.....	29-2
29.3.2 Snooze Control.....	29-2
29.3.3 USB Interrupt Enable.....	29-2
29.4 RAM Location.....	29-2
29.5 Boot Register.....	29-2
29.6 Control Register Details.....	29-3
RTC Wait Control Register (MISC_RTCWT).....	29-3
USB Configuration Register (MISC_USB).....	29-3
Internal RAM Wait Control Register (MISC_RAMWT).....	29-4
Boot Register (MISC_BOOT).....	29-5
RAM Location Select Register (MISC_RAM_LOC).....	29-5
Misc Protect Register (MISC_PROTECT).....	29-6
30 Divider (DIV).....	30-1
31 Electrical Characteristics.....	31-1
31.1 Absolute Maximum Rating.....	31-1
31.2 Recommended Operating Conditions.....	31-1
31.3 DC Characteristics.....	31-2
31.4 Current Consumption.....	31-3
31.5 A/D Converter Characteristics.....	31-5
31.6 Oscillation Characteristics.....	31-6
31.7 PLL Characteristics.....	31-7
31.8 AC Characteristics.....	31-7
31.8.1 External Clock Input Characteristics.....	31-7
31.8.2 SRAMC AC Characteristics.....	31-7
31.8.3 SDRAMC AC Characteristics.....	31-8
31.8.4 USI/USIL AC Characteristics.....	31-11
31.8.5 LCDC AC Characteristics.....	31-13
31.8.6 #STBY AC Characteristics.....	31-24
31.9 USB DC and AC Characteristics.....	31-24
32 Basic External Connection Diagram.....	32-1

Appendix A List of I/O Registers.....	AP-A-1
0x300010–0x300020	Misc Registers (MISC)..... AP-A-9
0x300100–0x300110	Clock Management Unit (CMU)..... AP-A-9
0x300210–0x30022c	Interrupt Controller (ITC) AP-A-11
0x300300–0x30083f	GPIO & Port MUX..... AP-A-13
0x300400–0x300472	USI..... AP-A-25
0x300600–0x30069f	USIL..... AP-A-27
0x300700–0x30070f	FSIO Ch.0..... AP-A-30
0x300710–0x30071f	FSIO Ch.1..... AP-A-31
0x300a00–0x300a0f	Real-time Clock (RTC)..... AP-A-32
0x300b00–0x300b0f	BBRAM..... AP-A-33
0x300c00–0x300c9f	USB Function Controller (USB) AP-A-33
0x300e00	Prescaler (PSC)..... AP-A-39
0x301000–0x30100c	Watchdog Timer (WDT) AP-A-39
0x301100–0x301108	8-bit Timer (T8) Ch.0 (with Fine mode)..... AP-A-40
0x301110–0x301118	8-bit Timer (T8) Ch.1 (with Fine mode)..... AP-A-41
0x301120–0x301128	8-bit Timer (T8) Ch.2 (with Fine mode)..... AP-A-41
0x301130–0x301138	8-bit Timer (T8) Ch.3 (with Fine mode)..... AP-A-42
0x301140–0x301148	8-bit Timer (T8) Ch.4 AP-A-43
0x301150–0x301158	8-bit Timer (T8) Ch.5 AP-A-43
0x301160–0x301168	8-bit Timer (T8) Ch.6 AP-A-44
0x301170–0x301178	8-bit Timer (T8) Ch.7 AP-A-44
0x301180–0x30118c	16-bit PWM Timer (T16A5) Ch.0 AP-A-45
0x301190–0x30119c	16-bit PWM Timer (T16A5) Ch.1 AP-A-46
0x301200–0x30120e	16-bit Audio PWM Timer (T16P)..... AP-A-47
0x301300–0x301306	A/D Converter (ADC10)..... AP-A-48
0x301400–0x301412	I ² S..... AP-A-49
0x301500–0x301506	Remote Controller (REMC) AP-A-50
0x302000–0x302094	LCD Controller (LCDC)..... AP-A-51
0x302100–0x30211c	DMA Controller (DMAC) AP-A-54
0x302200–0x302210	SDRAM Controller (SDRAMC) AP-A-55
0x302220–0x302228	SRAM Controller (SRAMC) AP-A-56
0x302300–0x302360	Cache Controller (CCU) AP-A-57
0x30240c–0x302925	Graphics Engine (GE) AP-A-58
Appendix B Power Saving	AP-B-1
Appendix C Mounting Precautions.....	AP-C-1
Appendix D Boot	AP-D-1
D.1 Boot Mode	AP-D-1
D.2 NOR Flash/External ROM Boot.....	AP-D-1
D.2.1 Configuration of NOR Flash/External ROM Boot System	AP-D-1
D.2.2 NOR Flash/External ROM Boot Sequence	AP-D-2
D.2.3 Reset Vector for NOR Flash/External ROM Boot	AP-D-2
D.3 SPI-EEPROM Boot.....	AP-D-3
D.3.1 Configuration of SPI-EEPROM Boot System	AP-D-3
D.3.2 SPI-EEPROM Boot Sequence	AP-D-4
D.3.3 EEPROM Data	AP-D-5
D.4 PC RS232C Boot.....	AP-D-6
D.4.1 Configuration of PC RS232C Boot System	AP-D-6
D.4.2 PC RS232C Boot Sequence	AP-D-6
D.4.3 Transfer Data	AP-D-8

Revision History

1 Overview

The S1C33L26 is a 32-bit application specific RISC controller that features extensive peripheral circuits such as an enhanced drawing graphics module, GPIO ports, serial interface modules, a USB module, PWM generators, and an A/D converter. It is suitable for applications that require a high-resolution LCD display, e.g. control panels on OA/FA equipment and intelligent remote controllers.

The S1C33L26 incorporates an LCD controller and VRAM supporting four-level gray scale QVGA display in single-chip. Adding an external SDRAM expands this capability into a higher resolution and with more displayable colors (e.g., 64K-color VGA display). An LCD driver interface with DMA function is also implemented allowing efficient data transfer to LCD modules that include a built-in VRAM LCD driver.

In addition, the embedded Graphics Engine (GE) provides rich graphic features, such as drawing functions for dots, straight lines, triangles, rectangles, and circles, resizing, and rotation, that can be used simply by calling commands. The GE also supports drawing of lossless compressed image data, this makes it possible to reduce CPU load and image data ROM size.

As for DSP functions, a 32-bit \times 32-bit multiplier (MUL) and a 16-bit \div 16-bit divider (DIV) are implemented. These functions help reduce CPU load for ADPCM audio data playback processing. Also the embedded I²S interface module is capable of being used to connect an external audio DAC.

The S1C33L26 has adopted the EPSON SoC (System on Chip) design technology using 0.18 μ m low power CMOS process to install these features.

1.1 Features

The features of the S1C33L26 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
- Maximum operating frequency: 60 MHz (36 MHz in SDRAM double frequency mode)
- Internal two-stage pipeline
- Instruction set: 125 instructions (16-bit fixed length)
- Dual AMBA bus system for CPU and GE

DSP

- Multiplier (MUL)
 - 32 \times 32 bits (seven cycles) or 16 \times 16 bits (five cycles)
- Divider (DIV)
 - 16 \div 16 bits (18 cycles)

Internal Memories

- IRAM (Internal RAM)
 - 12K bytes
- IVRAM (Internal VRAM)
 - 20K bytes
 - Configurable as a 32K-byte general-purpose RAM sequentially addressed with IRAM
- Cache RAM
 - 1K bytes (instruction cache RAM)
 - 1K bytes (data cache RAM)
 - Usable as a general-purpose RAM when not used as cache RAM
- DSTRAM (DMA descriptor RAM)/LUTRAM (look-up table RAM)
 - 512 bytes (can exclusively be used as either DSTRAM or LUTRAM.)
 - DMA descriptor RAM for storing DMA control table (128 \times 32 bits)

1 OVERVIEW

- Color look-up table RAM for LCDC (256 × 16 bits)
16 bits = (R: 5 bits, G: 6 bits, B: 5 bits)
- The DMA control table can be located in the IVRAM or an external RAM when this RAM is used for LCDC color look-up table.
- BBRAM (Battery backup RAM)
 - 16 bytes
 - The RAM contents can be maintained while the system power is off using the separated power supply for RTC.

Input clock

- High-speed clock (OSC3)
 - Maximum input clock frequency: 48 MHz
 - Internal oscillator circuit (crystal or ceramic resonator) or external clock input
- Low-speed clock (OSC1)
 - 32.768 kHz (typ.) clock for RTC and low-power operations
 - Internal oscillator circuit (crystal resonator) or external clock input

Cache Controller (CCU)

- 1K-byte instruction cache and 1K-byte data cache with a four-way associative frame structure (four frames/way, four lines/frame, four words/line)
- LRU replacement algorithm
- Automatic lock function during debug mode and the interrupt process of specified priority
- The instruction cache RAM and data cache RAM can be used as a general-purpose RAM when the cache function is disabled.

DMA Controller (DMAC)

- Eight channels of table DMA
- Supports table reloading and low-priority channel pausing functions.
- Trigger sources
 - USI (SPI/UART)
 - USIL (SPI/UART/Built-in RAM LCD interface)
 - FSIO (Asynchronous/Synchronous)
 - I²S
 - 16-bit audio PWM timer (T16P)
 - A/D converter (ADC10)
 - I/O ports (GPIO)
 - USB function controller (USB)
 - 16-bit PWM timer (T16A5)
 - Software

Graphics Engine (GE)

- Drawing objects
 - Shapes with a line width specified: straight lines (vertical, horizontal, and sloped lines), rectangle frames, and circle rings
 - Solid Shapes: points, triangles, rectangles, quadrilaterals, circles
 - Texts with a font specified
 - Compressed image (original run-length encoding)
- Image data block transfer
 - Rectangle area copy within VRAM
 - Data copy between VRAM and other memory
- Drawing effects
 - Clipping draw
 - Line width setting
 - Drawing color setting with transparency feature
 - Fill/Mesh/Rewrite/XOR

- Color conversion with palette/color depth conversion
- Resize/repeat/rotation (texts/compressed image)
- Word boundary commands (variable length) are provided for drawing control and commands list.

SRAM Controller (SRAMC)

- Allows direct connection of SRAM, ROM, and Flash memories.
- 26-bit address bus and 8/16-bit selectable data bus
- Up to six chip enable signals are available to connect external devices.
- Up to 64M-byte (A[25:0]) address space can be accessible with each chip enable signal.
- Programmable bus access wait cycle (0 to 15 cycles)
- Supports little endian access.
- Memory mapped I/O
- Supports both A0 and BS (Bus Strobe) type devices.
- Supports external wait request via the #WAIT pin.

SDRAM Controller (SDRAMC)

- Supports SDRAM direct interface. (Max. 72 MHz SDRAM clock)
- 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: one, two, or three programmable
- Supports burst and single read/write operations.
- Equipped with a four-stage × 16-bit DQB (Data Queue Buffer).
- Supports up to four SDRAM banks and bank active mode.
- Built-in 12-bit auto-refresh counter
- Intelligent self-refresh function for low power operation
- Arbitrates ownership of the external bus between the CPU, DMAC, LCDC, and GE.

Clock Management Unit/Oscillators/PLL (CMU)

- Selects the system clock source (OSC3, PLL, OSC1).
- Turns the OSC3 and OSC1 oscillator circuits on and off.
- Controls frequency multiplication rate of the PLL (×1 to ×16).
- Controls clocks according to the standby mode (SLEEP and HALT).
- Controls the external clock.
- Controls clock supply to the core and peripheral modules.
- OSC3 oscillator circuit
 - Crystal oscillation: 5 MHz min. to 48 MHz max.
 - Ceramic oscillation: 5 MHz min. to 48 MHz max.
 - 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.
- * Before using a ceramic resonator, please be sure to contact Murata Manufacturing Co., Ltd. for further information on conditions of use for ceramic resonators.
- PLL
 - PLL input frequency: 5 MHz min. to 48 MHz max. (OSC3 ×1, ×1/2, ×1/3, ... ×1/9, ×1/10)
 - PLL output frequency: 20 MHz min. to 72 MHz max.
 - Multiplication rate: ×1, ×2, ×3, ... ×15, ×16
- OSC1 oscillator circuit
 - Crystal oscillation: 32.768 kHz typ.
 - External clock input: 32.768 kHz typ.

Interrupt Controller (ITC)

- Five non-maskable interrupts
- 31 maskable interrupts (including four software interrupts)

16-bit Audio PWM Timer (T16P)

- One channel of 16-bit timer/counter with PWM output function
- Three bit division modes are provided. (10 bits + 6 bits, 9 bits + 7 bits, 8 bits + 8 bits)
- Supports 8, 16, 22.05, 32, 44.1, and 48 kHz sampling rates.
- Audio PWM function supporting 8-bit and 16-bit PCM data (mono)
- Can output monophonic sound without using an external DAC (external resistors and capacitors are required).
- Supports fine mode to control pulse widths with high accuracy.
- Supports a digital volume control function.
- Can generate two types of compare-match interrupts.
- Supports DMA transfer.

8-bit Timers (T8, T8F)

- T8F: four channels of 8-bit timer with fine mode (presetable down counter)
T8: four channels of 8-bit timer without fine mode
- Clocks generated with the counter underflow can be output to internal devices.
 - The T8F Ch.0 can be used as the USI clock generator.
 - The T8F Ch.3 can be used as the USIL clock generator.
 - The T8F Ch.2 can be used to trigger the ADC10.
- Each timer can generate underflow interrupts.

16-bit PWM Timer (T16A5)

- Two channels of 16-bit timer with a counter capture/comparison functions
- Each channel has built-in two comparison/capture data buffers.
- Can generate compare/capture interrupts.
- Supports DMA transfer.

Watchdog Timer (WDT)

- 30-bit watchdog timer to generate an NMI or a reset
- Programmable watchdog timer overflow period (NMI or reset interrupt period)
- The watchdog timer overflow signal can be output outside the IC.

Real Time Clock (RTC)

- Contains time counters (seconds, minutes, and hours) and calendar counters (days, days of the week, months, and year).
- 24-hour or 12-hour mode can be selected.
- Operates with an independent power supply (RTCVDD) separated from system power (operable while the system power is off).
- Provides the WAKEUP output pin and #STBY input pin to control standby mode.
- Can generate periodic interrupts.

Universal Serial Interface (USI)

- Multi-serial I/O that can be used as a UART, SPI, or I²C module
- Contains 1-byte receive data buffer and 1-byte transmit data buffer.
- UART mode
 - Character length: 7 or 8 bits
 - Parity mode: even, odd, or no parity
 - Stop bit: 1 or 2 bits (start bit: 1 bit fixed)
 - Supports both MSB first and LSB first modes.
 - Parity error, framing error, and overrun error detectable
 - Can generate receive buffer full, transmit buffer empty, and receive error interrupts.
 - Supports DMA transfer.
- SPI mode
 - Supports both master and slave modes.
 - Data length: 8 or 9 bits (master mode), 8 bits fixed (slave mode)

- Supports both MSB first and LSB first modes.
- Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
- Can generate receive buffer full, transmit buffer empty, and overrun error interrupts.
- Supports DMA transfer.
- I²C mode
 - Supports both master (single master only) and slave modes.
 - 7-bit addressing mode (10-bit addressing is possible by software control.)
 - Supports clock stretch/wait functions.
 - Can generate start/stop, data transfer, ACK/NAK transfer, and overrun error interrupts.

Universal Serial Interface with Built-in RAM LCD interface (USIL)

- Multi-serial I/O that can be used as a UART, SPI, I²C, or built-in RAM LCD interface module
- Contains 1-byte receive data buffer and 1-byte transmit data buffer.
- UART mode
 - Character length: 7 or 8 bits
 - Parity mode: even, odd, or no parity
 - Stop bit: 1 or 2 bits (start bit: 1 bit fixed)
 - Supports both MSB first and LSB first modes.
 - Parity error, framing error, and overrun error detectable
 - Can generate receive buffer full, transmit buffer empty, and receive error interrupts.
 - Supports DMA transfer.
- SPI mode
 - Supports both master and slave modes.
 - Data length: 8 bits fixed
 - Supports both MSB first and LSB first modes.
 - Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
 - Can generate receive buffer full, transmit buffer empty, and overrun error interrupts.
 - Supports DMA transfer.
- I²C mode
 - Supports both master (single master only) and slave modes.
 - 7-bit addressing mode (10-bit addressing is possible by software control.)
 - Supports clock stretch/wait functions.
 - Can generate start/stop, data transfer, ACK/NAK transfer, and overrun error interrupts.
- LCD SPI mode
 - Data length is configurable for 8 bits, 16 bits, 18 bits (4 data format) and 24 bits + CMD bit.
 - CMD bit or A0 is selectable.
 - Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
 - Can generate transmit buffer empty interrupts.
 - Supports DMA transfer.
- LCD parallel interface mode
 - Provides 8-bit data bus, #CS, #RD, #WR and A0 control signals.
 - Supports byte read/write access mode only.
 - Can generate transmit buffer empty and receive buffer full interrupts.
 - Supports DMA transfer for both data transmission and reception.
 - Access timings can be controlled using T8F. The setup cycle (1 to 4), hold cycle (1 to 4), and wait cycle (1 to 16) are configurable.

Serial Interface with FIFO (FSIO)

- Two channels of clock synchronous/asynchronous serial interface
- Contains FIFO data buffers (4-byte receive data buffer and 2-byte transmit data buffer are available for each channel).
- Supports IrDA1.0-equivalent communications by software control or using an external IrDA driver.
- Contains a baud-rate generator (12-bit programmable timer).
- Can generate receive buffer full, transmit buffer empty, and receive error interrupts.
- Supports DMA transfer.

I²S Bus Interface (I2S)

- Supports universal audio I²S bus interface.
- Contains a 16-byte transmit FIFO (16 bits × 2 channels × 4)
- I²S output: one channel
- Resolution: 16 bits (PCM data output format)
- Operates as master that generates the bit clock, word-select signal, data and master clock.
- Clock polarity and data shift direction (MSB first/LSB first) are software configurable.
- Can generate I²S FIFO empty interrupts.
- Supports DMA transfer.

Card Interface (CARD)

- Generates 8-bit or 16-bit NAND Flash interface signals.
- The ECC and EDC functions should be implemented in the application program.

Infrared Remote Controller (REMC)

- Outputs a modulated carrier signal and inputs remote control pulses.
- Embedded carrier signal generator and data length counter
- Can generate counter underflow interrupts for data transmission and input rising/falling edge detection interrupts for data reception.

LCD Controller (LCDC)

- Supports STN LCD panels with 4/8-bit data lines or TFT LCD panels with up to 24-bit data lines.
- Supports generic panel resolutions up to VGA, such as 640 × 480 pixels (VGA) and 320 × 240 pixels (QVGA) (can be configured according to the panel used).
- Supports up to 16M-color (for color TFT), 4K-color (for color STN), and 16-level gray scale (for monochrome STN) display modes.
- Display configuration when the internal VRAM (20KB) is used
 - 320 × 240 pixels, 2 bpp (4-level gray scale display)
- Display configuration when an external memory is used
 - 320 × 240 pixels, 16 bpp (QVGA 64K-colors display)
 - 400 × 240 pixels, 16 bpp (WQVGA 64K-colors display)
 - 640 × 480 pixels, 16 bpp (VGA 64K-colors display)
- Two-image overlay display via the Picture-in-Picture Plus function
- Brightness/gray scale control via the mono display LUT (Look Up Table with 16 × 4 bits), or palette control via the color display LUT (Look Up Table RAM with 256 × 16 bits)
- Virtual display function to handle images with a different resolution from the LCD panel (any area in the virtual screen can be displayed on the LCD.)

A/D Converter (ADC10)

- 10-bit A/D converter with up to six analog input channels
- Conversion time: 10 μs min. (when 2 MHz input clock is selected)
1,250 μs max. (when 16 kHz input clock is selected)
- Can generate conversion completion and data overwrite error interrupts.

USB Function Controller (USB)

- Supports USB2.0 full speed (12M bps) mode.
- Supports auto negotiation function.
- Supports control, bulk, isochronous and interrupt transfers.
- Supports four general-purpose endpoints and endpoint 0 (control).
- Embedded 1K-byte programmable FIFO
- Can generate USB interrupts.
- Supports DMA transfer.

General-purpose I/O Ports (GPIO)

- Maximum 71 I/O ports and six input ports are available (144-pin package).
 - Can generate maximum 16 input interrupts from 64 I/O ports.
 - Supports DMA transfer.
- * The GPIO ports are shared with other peripheral function pins (USI, PWM etc.). Therefore, the number of GPIO ports depends on the peripheral functions used.

Operating Voltage

- HV_{DD} (I/O power voltage)
2.7 V to 3.6 V (3.3 V typ.)
or 3.0 V to 3.6 V (3.3 V typ.) when the USB module is used.
 - AV_{DD} (analog power voltage)
2.7 V to 3.6 V (3.3 V typ.) or 3.0 V to 3.6 V (3.3 V typ.)
 - LV_{DD} (core/internal logic power voltage)
1.65 V to 1.95 V (1.8 V typ.)
or 1.7 V to 1.9 V (1.8 V typ.) when a ceramic resonator is used.
 - PLL_{DD} (PLL power voltage)
1.65 V to 1.95 V (1.8 V typ.)
or 1.7 V to 1.9 V (1.8 V typ.) when a ceramic resonator is used.
 - RTC_{VDD} (RTC/BBRAM power voltage)
1.65 V to 1.95 V (1.8 V typ.)
or 1.7 V to 1.9 V (1.8 V typ.) when a ceramic resonator is used.
- * $LV_{DD} = PLL_{DD} = RTC_{VDD}$
The S1C33L26 does not support 5 V tolerant I/O.

Operating Temperatures

- -40 to 85°C
- 0 to 70°C when the USB module and a ceramic resonator are used.

Current Consumption

(No I/O current is included.)

- During SLEEP: 2.9 μ A typ. when RTC is running.
1.6 μ A typ. when RTC is not used.
- During HALT: 4.2 mA typ. when 48 MHz OSC3 clock is used as the system and CPU clocks,
all peripheral clocks = Off.
- During execution: 22 mA typ. when 48 MHz OSC3 clock is used as the system and CPU clocks,
CPU is running, all peripheral clocks = Off.
40 mA typ. when 48 MHz OSC3 clock is used as the system and CPU clocks,
CPU is running, GE is enabled and idles.

* Current consumption can be reduced by controlling the clocks through the clock management unit (CMU).

Shipping Form

- Die form
- Plastic package: TQFP15-128pin
TQFP24-144pin
PFBGA12U-180

1.2 Block Diagram

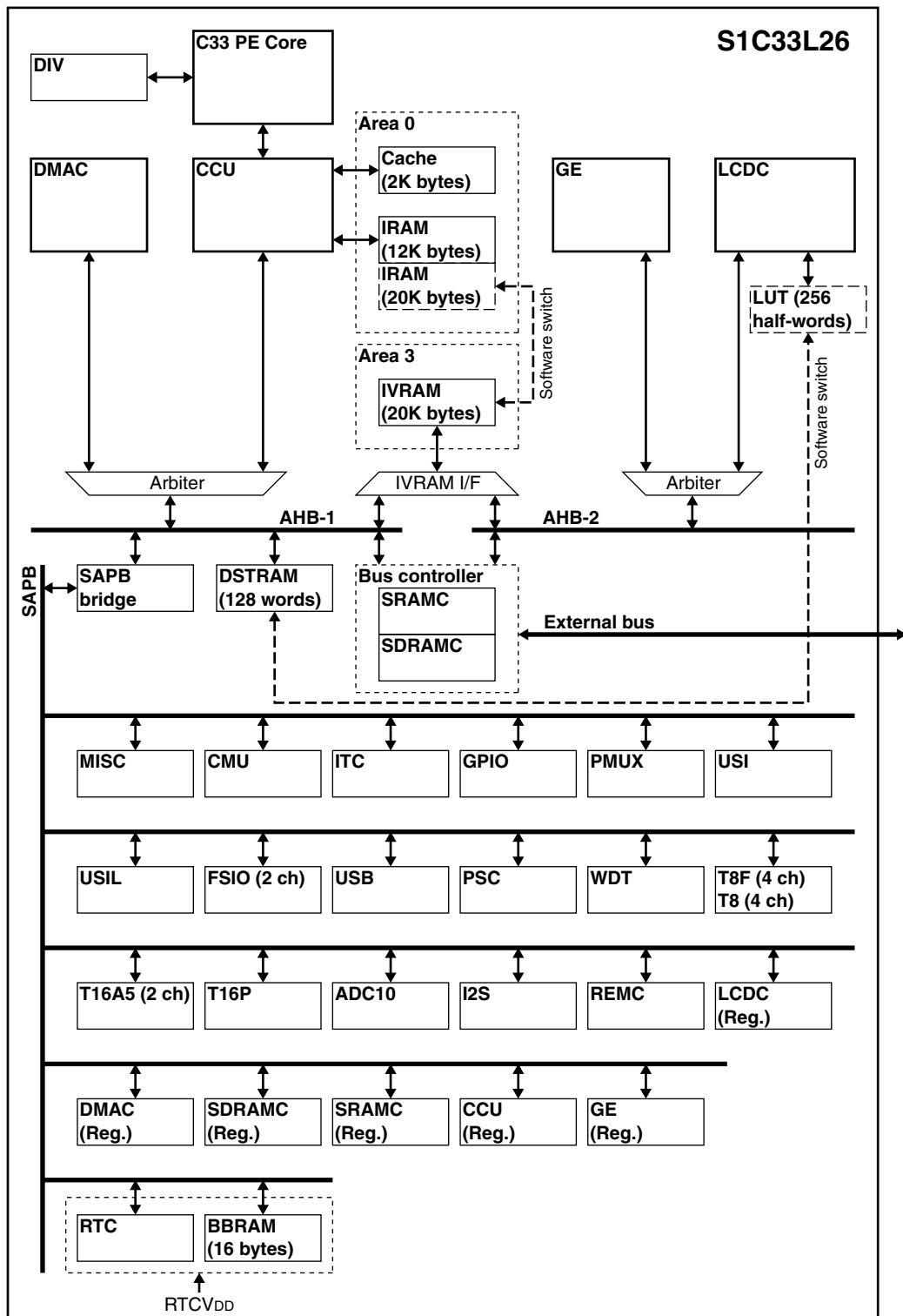


Figure 1.2.1 Block Diagram

TQFP24-144pin package

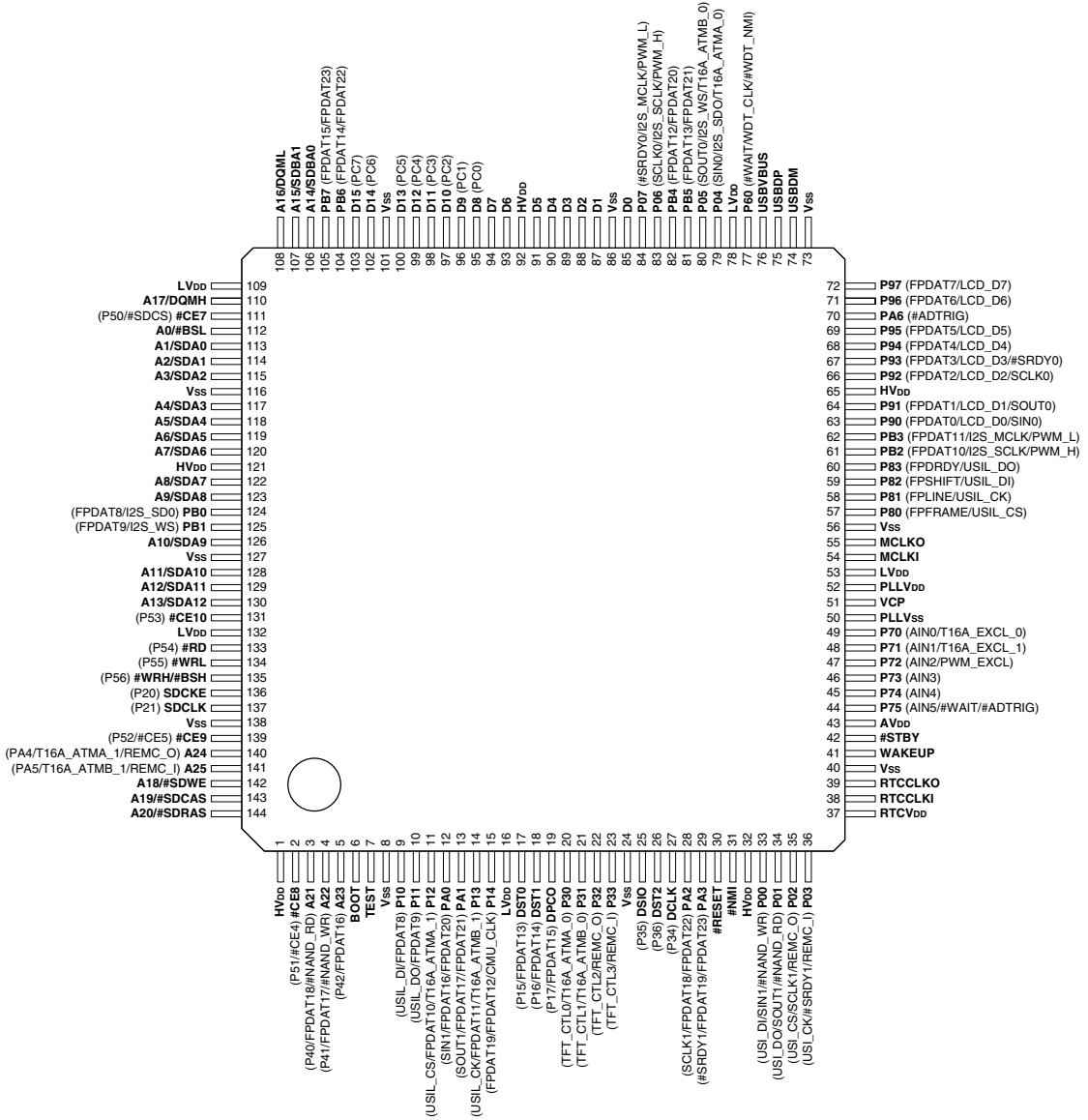


Figure 1.3.1.2 Pin Arrangement (TQFP24-144pin)

PFBGA12U-180 package

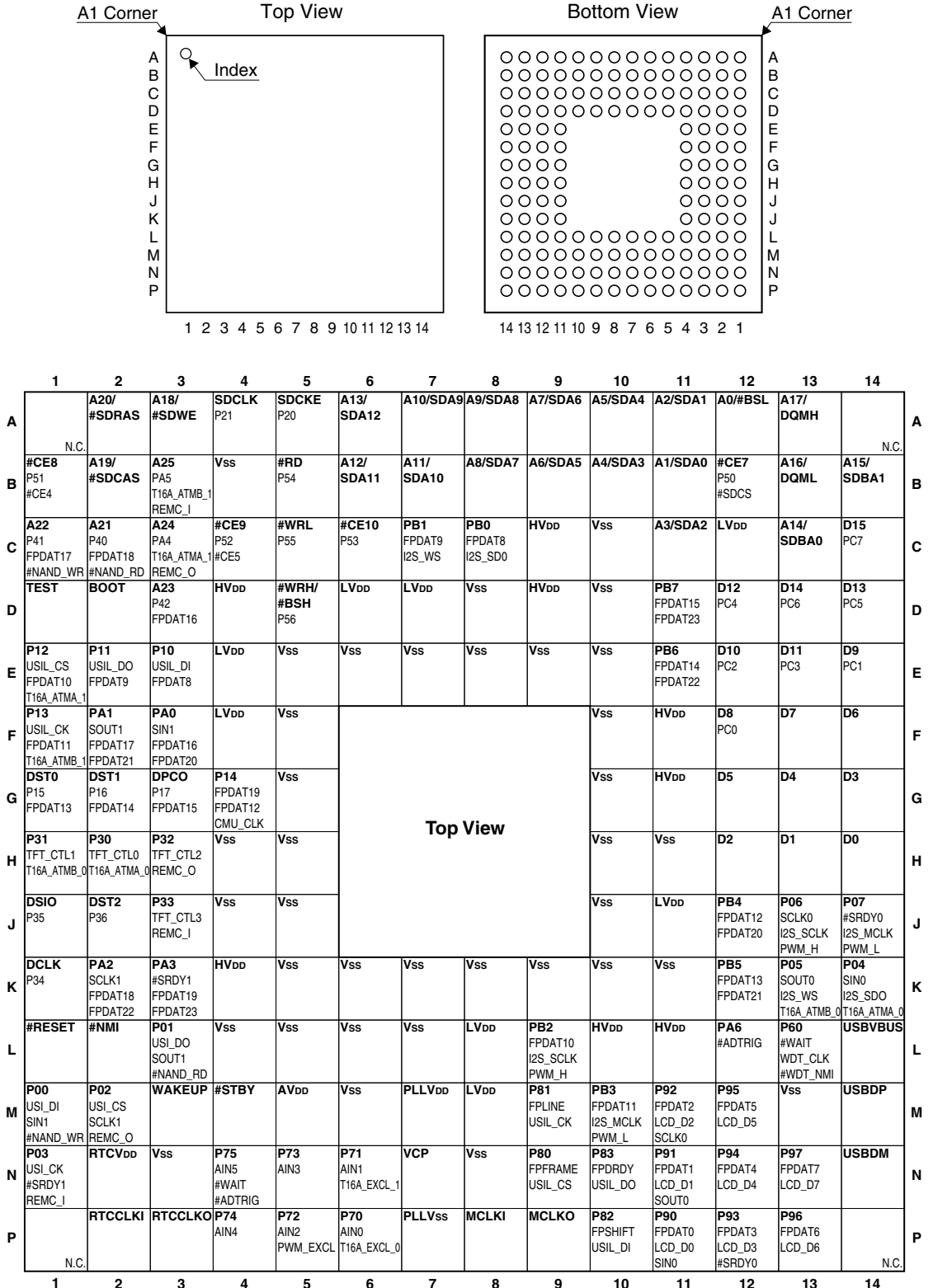


Figure 1.3.1.3 Pin Arrangement (PFBGA12U-180)

1.3.2 Pin Functions

The tables below list the S1C33L26 pin functions.

Notes:

- Pin name

- The # prefixed to pin names indicates that the pin inputs/outputs an active low signal.
- The pin names listed in boldface denote the default pin (signal) name.
Each pin is assigned one to four functions and the function to be used must be selected using the corresponding port function select bit. See the “I/O Ports (GPIO)” chapter for more information on the pin function selections.

- Be sure to avoid assigning the same signal to more than one pin.

- I/O

- The I/O listed in boldface and uppercase denote the default input/output direction.
- (H), (L), and (X) denote the initial output level, High, Low, and Undefined, respectively.

- PWR (power system)

- P1: LV_{DD}
- P2: HV_{DD}
- P3: RTCV_{DD}
- P4: AV_{DD}
- P5: PLLV_{DD}

- DC characteristics

- Output Type 1: I_{OH}/I_{OL} = 2 mA (HV_{DD} = 3.0 V)*
Type 2: I_{OH}/I_{OL} = 4 mA (HV_{DD} = 3.0 V)*
Type 3: I_{OH}/I_{OL} = 8 mA (HV_{DD} = 3.0 V)*
* See “DC Characteristics” in the “Electrical Characteristics” chapter.
- PU/PD PU: Pull-up
PD: Pull-down
PUC: Pull-up with software control
(en): Enabled by default
(dis): Disabled by default

Table 1.3.2.1 List of Power Supply Pins

No.	Pin name	I/O	Description	Pin No.			Power voltage
				TQFP15 128	TQFP24 144	PFBGA 180	
1	HV_{DD}	–	I/O power supply pin	1, 28, 58, 82, 109	1, 32, 65, 92, 121	C9, D4, D9, F11, G11, K4, L10, L11	3.3 V typ. (2.7 to 3.6 V or 3.0 to 3.6 V when the USB module is used.)
2	AV_{DD}	–	Analog power supply pin	38	43	M5	3.3 V typ. (2.7 to 3.6 V or 3.0 to 3.6 V)
3	LV_{DD}	–	Core power supply pin	14, 48, 70, 97, 118	16, 53, 78, 109, 132	C12, D6, D7, E4, F4, J11, L8, M8	1.8 V typ. (1.65 to 1.95 V or 1.7 to 1.9 V when a cer- amic resonator is used.)
4	RTCV_{DD}	–	RTC/BBRAM power supply pin	33	37	N2	1.8 V typ. (= LV _{DD})
5	V_{SS}	–	Ground pin	8, 22, 36, 51, 65, 76, 91, 104, 113, 124	8, 24, 40, 56, 73, 86, 101, 116, 127, 138	B4, C10, D8, D10, E5, E6, E7, E8, E9, E10, F5, F10, G5, G10, H4, H5, H10, H11, J4, J5, J10, K5, K6, K7, K8, K9, K10, K11, L4, L5, L6, L7, M6, M13, N3, N8	GND
6	PLLV_{DD}	–	PLL power supply pin	47	52	M7	1.8 V typ. (= LV _{DD})
7	PLLV_{SS}	–	PLL power supply ground pin	45	50	P7	GND (= V _{SS})

Table 1.3.2.2 List of Clock Pins

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
1	MCLKI	I	High speed (OSC3) oscillation input (crystal/ceramic resonator or external clock input)	49	54	P8	P1	–	–	–
2	MCLKO	O	High speed (OSC3) oscillation output (crystal/ceramic resonator)	50	55	P9	P1	–	–	–
3	RTCCLKI	I	RTC (OSC1) oscillation input (crystal resonator or external clock input)	34	38	P2	P3	–	–	–
4	RTCCLKO	O	RTC (OSC1) oscillation output (crystal resonator)	35	39	P3	P3	–	–	–

Table 1.3.2.3 List of External Bus Pins

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
1	D0	I/o	Data bus D0	75	85	H14	P2	LVTTTL	Type 2	Bus hold latch
2	D1	I/o	Data bus D1	77	87	H13				
3	D2	I/o	Data bus D2	78	88	H12				
4	D3	I/o	Data bus D3	79	89	G14				
5	D4	I/o	Data bus D4	80	90	G13				
6	D5	I/o	Data bus D5	81	91	G12				
7	D6	I/o	Data bus D6	83	93	F14				
8	D7	I/o	Data bus D7	84	94	F13				
9	D8	I/o	Data bus D8 (default)	85	95	F12				
	PC0	i/o	I/O port							
10	D9	I/o	Data bus D9 (default)	86	96	E14				
	PC1	i/o	I/O port							
11	D10	I/o	Data bus D10 (default)	87	97	E12				
	PC2	i/o	I/O port							
12	D11	I/o	Data bus D11 (default)	88	98	E13				
	PC3	i/o	I/O port							
13	D12	I/o	Data bus D12 (default)	89	99	D12				
	PC4	i/o	I/O port							
14	D13	I/o	Data bus D13 (default)	90	100	D14				
	PC5	i/o	I/O port							
15	D14	I/o	Data bus D14 (default)	92	102	D13				
	PC6	i/o	I/O port							
16	D15	I/o	Data bus D15 (default)	93	103	C14				
	PC7	i/o	I/O port							
17	A0#BSL	O	Address bus A0 / Bus strobe (low byte) signal output (L)	100	112	A12	P2	–	Type 2	–
18	A1/SDA0	O	Address bus A1 / SDRAM address A0 (L)	101	113	B11				
19	A2/SDA1	O	Address bus A2 / SDRAM address A1 (L)	102	114	A11				
20	A3/SDA2	O	Address bus A3 / SDRAM address A2 (L)	103	115	C11				
21	A4/SDA3	O	Address bus A4 / SDRAM address A3 (L)	105	117	B10				
22	A5/SDA4	O	Address bus A5 / SDRAM address A4 (L)	106	118	A10				
23	A6/SDA5	O	Address bus A6 / SDRAM address A5 (L)	107	119	B9				
24	A7/SDA6	O	Address bus A7 / SDRAM address A6 (L)	108	120	A9				
25	A8/SDA7	O	Address bus A8 / SDRAM address A7 (L)	110	122	B8				
26	A9/SDA8	O	Address bus A9 / SDRAM address A8 (L)	111	123	A8				
27	A10/SDA9	O	Address bus A10 / SDRAM address A9 (L)	112	126	A7				

1 OVERVIEW

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics						
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD				
28	A11/SDA10	O (L)	Address bus A11 / SDRAM address A10	114	128	B7	P2	-	Type 2	-				
29	A12/SDA11	O (L)	Address bus A12 / SDRAM address A11	115	129	B6								
30	A13/SDA12	O (L)	Address bus A13 / SDRAM address A12	116	130	A6								
31	A14/SDBA0	O (L)	Address bus A14 / SDRAM bank address BA0	94	106	C13								
32	A15/SDBA1	O (L)	Address bus A15 / SDRAM bank address BA1	95	107	B14								
33	A16/DQML	O (H)	Address bus A16 / SDRAM data mask (low-order byte) signal output	96	108	B13								
34	A17/DQMH	O (H)	Address bus A17 / SDRAM data mask (high-order byte) signal output	98	110	A13								
35	A18/#SDWE	O (H)	Address bus A18 / SDRAM write signal output	126	142	A3								
36	A19/#SDCAS	O (H)	Address bus A19 / SDRAM column address strobe signal output	127	143	B2								
37	A20/#SDRAS	O (H)	Address bus A20 / SDRAM row address strobe signal output	128	144	A2								
38	A21	O (L)	Address bus A21 (default)	3	3	C2					P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	P40	i/o	I/O port											
	FPDAT18	o	LCD data output											
	#NAND_RD	o	NAND Flash read signal output											
39	A22	O (L)	Address bus A22 (default)	4	4	C1								
	P41	i/o	I/O port											
	FPDAT17	o	LCD data output											
	#NAND_WR	o	NAND Flash write signal output											
40	A23	O (L)	Address bus A23 (default)	5	5	D3								
	P42	i/o	I/O port											
	FPDAT16	o	LCD data output											
41	A24	O (L)	Address bus A24 (default)	-	140	C3								
	PA4	i/o	I/O port											
	T16A_ATMA_1	i/o	T16A5 Ch.1 capture A signal input/compare A signal output											
	REMC_O	o	REMC transmit signal output											
42	A25	O (L)	Address bus A25 (default)	-	141	B3								
	PA5	i/o	I/O port											
	T16A_ATMB_1	i/o	T16A5 Ch.1 capture B signal input/compare B signal output											
	REMC_I	i	REMC receive signal input											
43	#CE7	O (H)	Area 7/19 chip enable signal output (default)	99	111	B12	P2	LVCMOS Schmitt	Type 2	100k PUc (dis)				
	P50	i/o	I/O port											
	#SDCS	o	SDRAM chip enable signal output											
44	#CE8	O (H)	Area 8/21 chip enable signal output (default)	2	2	B1								
	P51	i/o	I/O port											
	#CE4	o	Area 4/14 chip enable signal output											
45	#CE9	O (H)	Area 9/22 chip enable signal output (default)	125	139	C4								
	P52	i/o	I/O port											
	#CE5	o	Area 5/15/16 chip enable signal output											
46	#CE10	O (H)	Area 10/20 chip enable signal output (default)	117	131	C6					P2	LVCMOS Schmitt	Type 1	100k PUc (en)*
	P53	i/o	I/O port											

* The #CE10 pull-up resistor is enabled (en) when the BOOT pin is set to 0 or disabled (dis) when the BOOT pin is set to 1.

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
47	#RD	O (H)	Read signal output (default)	119	133	B5	P2	LVCMOS Schmitt	Type 2	100k PUc (dis)
	P54	i/o	I/O port							
48	#WRL	O (H)	Write (low-order byte) signal output (default)	120	134	C5	P2	LVCMOS Schmitt	Type 2	100k PUc (dis)
	P55	i/o	I/O port							
49	#WRH/#BSH	O (H)	Write (high-order byte) signal / Bus strobe (high-order byte) signal output (default)	121	135	D5	P2	LVCMOS Schmitt	Type 2	100k PUc (dis)
	P56	i/o	I/O port							
50	SDCKE	O (H)	SDRAM clock enable signal output (default)	122	136	A5	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	P20	i/o	I/O port							
51	SDCLK	O (L)	SDRAM clock output (default)	123	137	A4	P2	LVCMOS Schmitt	Type 2	100k PUc (dis)
	P21	i/o	I/O port							

Table 1.3.2.4 List of I/O Port and Peripheral Function Pins

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
1	P00	I/o	I/O port (default)	29	33	M1	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	USI_DI	i/o	USI data input/output (see Table 1.3.2.8.)							
	SIN1	i	FSIO Ch.1 data input (see Table 1.3.2.10.)							
	#NAND_WR	o	NAND Flash write signal output							
2	P01	I/o	I/O port (default)	30	34	L3	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	USI_DO	o	USI data output (see Table 1.3.2.8.)							
	SOUT1	o	FSIO Ch.1 data output (see Table 1.3.2.10.)							
	#NAND_RD	o	NAND Flash read signal output							
3	P02	I/o	I/O port (default)	31	35	M2	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	USI_CS	i/o	USI slave select input, data input/output (see Table 1.3.2.8.)							
	SCLK1	i/o	FSIO Ch.1 clock input/output (see Table 1.3.2.10.)							
	REMC_O	o	REMC transmit signal output							
4	P03	I/o	I/O port (default)	32	36	N1	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	USI_CK	i/o	USI clock input/output (see Table 1.3.2.8.)							
	#SRDY1	i/o	FSIO Ch.1 ready signal input/output (see Table 1.3.2.10.)							
	REMC_I	i	REMC receive signal input							
5	P04	I/o	I/O port (default)	71	79	K14	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	SIN0	i	FSIO Ch.0 data input (see Table 1.3.2.10.)							
	I2S_SDO	o	I ² S serial data output							
	T16A_ATMA_0	i/o	T16A5 Ch.0 capture A signal input/compare A signal output							
6	P05	I/o	I/O port (default)	72	80	K13	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	SOUT0	o	FSIO Ch.0 data output (see Table 1.3.2.10.)							
	I2S_WS	o	I ² S word select signal output							
	T16A_ATMB_0	i/o	T16A5 Ch.0 capture B signal input/compare B signal output							
7	P06	I/o	I/O port (default)	73	83	J13	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	SCLK0	i/o	FSIO Ch.0 clock input/output (see Table 1.3.2.10.)							
	I2S_SCLK	o	I ² S serial bit clock output							
	PWM_H	o	T16P PWM_H signal output							
8	P07	I/o	I/O port (default)	74	84	J14	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	#SRDY0	i/o	FSIO Ch.0 ready signal input/output (see Table 1.3.2.10.)							
	I2S_MCLK	o	I ² S master clock output							
	PWM_L	o	T16P PWM_L signal output							

1 OVERVIEW

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
9	P10	I/o	I/O port (default)	9	9	E3	P2	LVCMOS Schmitt	Type 1	100k PUC (dis)
	USIL_DI	i/o	USIL data input/output or LCD control signal output (see Table 1.3.2.9.)							
	FPDAT8	o	LCD data output							
10	P11	I/o	I/O port (default)	10	10	E2				
	USIL_DO	o	USIL data output or LCD control signal output (see Table 1.3.2.9.)							
	FPDAT9	o	LCD data output							
11	P12	I/o	I/O port (default)	11	11	E1				
	USIL_CS	i/o	USIL slave select input, data input/output, or LCD control signal output (see Table 1.3.2.9.)							
	FPDAT10	o	LCD data output							
	T16A_ATMA_1	i/o	T16A5 Ch.1 capture A signal input/compare A signal output							
12	P13	I/o	I/O port (default)	12	14	F1				
	USIL_CK	i/o	USIL clock input/output or LCD control signal output (see Table 1.3.2.9.)							
	FPDAT11	o	LCD data output							
	T16A_ATMB_1	i/o	T16A5 Ch.1 capture B signal input/compare B signal output							
13	P14	I/o	I/O port (default)	13	15	G4				
	FPDAT19	o	LCD data output							
	FPDAT12	o	LCD data output							
	CMU_CLK	o	CMU clock external output							
14	P30	I/o	I/O port (default)	18	20	H2				
	TFT_CTL0	o	LCDC TFT I/F control signal 0 output							
	T16A_ATMA_0	i/o	T16A5 Ch.0 capture A signal input/compare A signal output							
15	P31	I/o	I/O port (default)	19	21	H1				
	TFT_CTL1	o	LCDC TFT I/F control signal 1 output							
	T16A_ATMB_0	i/o	T16A5 Ch.0 capture B signal input/compare B signal output							
16	P32	I/o	I/O port (default)	20	22	H3	P2	LVCMOS Schmitt	Type 3	100k PUC (dis)
	TFT_CTL2	o	LCDC TFT I/F control signal 2 output							
	REMC_O	o	REMC transmit signal output							
17	P33	I/o	I/O port (default)	21	23	J3	P2	LVCMOS Schmitt	Type 1	100k PUC (dis)
	TFT_CTL3	o	LCDC TFT I/F control signal 3 output							
	REMC_I	i	REMC receive signal input							
18	P60	I/o	I/O port (default)	69	77	L13	P2	LVCMOS Schmitt	Type 1	100k PUC (en)
	#WAIT	i	Wait cycle request input							
	WDT_CLK	o	Watchdog timer clock output							
	#WDT_NMI	o	Watchdog timer NMI signal output							
19	P70	I	Input port (default)	44	49	P6	P4	LVCMOS Analog LVCMOS	-	100k PUC (dis)
	AIN0	i	ADC10 Ch.0 analog input							
	T16A_EXCL_0	i	T16A5 Ch.0/WDT external clock input							
20	P71	I	Input port (default)	43	48	N6	P4	LVCMOS Analog LVCMOS	-	100k PUC (dis)
	AIN1	i	ADC10 Ch.1 analog input							
	T16A_EXCL_1	i	T16A5 Ch.1 external clock input							
21	P72	I	Input port (default)	42	47	P5	P4	LVCMOS Analog LVCMOS	-	100k PUC (dis)
	AIN2	i	ADC10 Ch.2 analog input							
	PWM_EXCL	i	T16P external clock input							
22	P73	I	Input port (default)	41	46	N5	P4	LVCMOS Analog	-	100k PUC (dis)
	AIN3	i	ADC10 Ch.3 analog input							
23	P74	I	Input port (default)	40	45	P4	P4	LVCMOS Analog	-	100k PUC (dis)
	AIN4	i	ADC10 Ch.4 analog input							
24	P75	I	Input port (default)	39	44	N4	P4	LVCMOS Analog LVCMOS LVCMOS	-	100k PUC (en)
	AIN5	i	ADC10 Ch.5 analog input							
	#WAIT	i	Wait cycle request input							
	#ADTRIG	i	ADC10 trigger input							

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
25	P80	I/o	I/O port (default)	52	57	N9	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	FPFRAME	o	LCD frame clock output							
	USIL_CS	i/o	USIL slave select input, data input/output, or LCD control signal output (see Table 1.3.2.9.)							
26	P81	I/o	I/O port (default)	53	58	M9	P2	LVCMOS Schmitt	Type 1	100k PUc (dis)
	FPLINE	o	LCD line clock output							
	USIL_CK	i/o	USIL clock input/output or LCD control signal output (see Table 1.3.2.9.)							
27	P82	I/o	I/O port (default)	54	59	P10				
	FPSHIFT	o	LCD shift clock output							
	USIL_DI	i/o	USIL data input/output or LCD control signal output (see Table 1.3.2.9.)							
28	P83	I/o	I/O port (default)	55	60	N10				
	FPDRDY	o	LCD DRDY/MOD signal output							
	USIL_DO	o	USIL data output or LCD control signal output (see Table 1.3.2.9.)							
29	P90	I/o	I/O port (default)	56	63	P11				
	FPDAT0	o	LCD data output							
	LCD_D0	i/o	USIL LCD data input/output							
	SIN0	i	FSIO Ch.0 data input (see Table 1.3.2.10.)							
30	P91	I/o	I/O port (default)	57	64	N11				
	FPDAT1	o	LCD data output							
	LCD_D1	i/o	USIL LCD data input/output							
	SOUT0	o	FSIO Ch.0 data output (see Table 1.3.2.10.)							
31	P92	I/o	I/O port (default)	59	66	M11				
	FPDAT2	o	LCD data output							
	LCD_D2	i/o	USIL LCD data input/output							
	SCLK0	i/o	FSIO Ch.0 clock input/output (see Table 1.3.2.10.)							
32	P93	I/o	I/O port (default)	60	67	P12				
	FPDAT3	o	LCD data output							
	LCD_D3	i/o	USIL LCD data input/output							
	#SRDY0	i/o	FSIO Ch.0 ready signal input/output (see Table 1.3.2.10.)							
33	P94	I/o	I/O port (default)	61	68	N12				
	FPDAT4	o	LCD data output							
	LCD_D4	i/o	USIL LCD data input/output							
34	P95	I/o	I/O port (default)	62	69	M12				
	FPDAT5	o	LCD data output							
	LCD_D5	i/o	USIL LCD data input/output							
35	P96	I/o	I/O port (default)	63	71	P13				
	FPDAT6	o	LCD data output							
	LCD_D6	i/o	USIL LCD data input/output							
36	P97	I/o	I/O port (default)	64	72	N13				
	FPDAT7	o	LCD data output							
	LCD_D7	i/o	USIL LCD data input/output							
37	PA0	I/o	I/O port (default)	-	12	F3	P2	LVCMOS Schmitt	Type 2	50k PUs (dis)
	SIN1	i	FSIO Ch.1 data input (see Table 1.3.2.10.)							
	FPDAT16	o	LCD data output							
	FPDAT20	o	LCD data output							
38	PA1	I/o	I/O port (default)	-	13	F2	P2	LVCMOS Schmitt	Type 2	50k PUs (en)
	SOUT1	o	FSIO Ch.1 data output (see Table 1.3.2.10.)							
	FPDAT17	o	LCD data output							
	FPDAT21	o	LCD data output							
39	PA2	I/o	I/O port (default)	-	28	K2	P2	LVCMOS Schmitt	Type 2	100k PUs (dis)
	SCLK1	i/o	FSIO Ch.1 clock input/output (see Table 1.3.2.10.)							
	FPDAT18	o	LCD data output							
	FPDAT22	o	LCD data output							

1 OVERVIEW

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
40	PA3	I/o	I/O port (default)	-	29	K3	P2	LVCMOS Schmitt	Type 1	100k PUs (dis)
	#SRDY1	i/o	FSIO Ch.1 ready signal input/output (see Table 1.3.2.10.)							
	FPDAT19	o	LCD data output							
	FPDAT23	o	LCD data output							
41	PA6	I/o	I/O port (default)	-	70	L12				
	#ADTRIG	i	ADC10 trigger input							
42	PB0	I/o	I/O port (default)	-	124	C8				
	FPDAT8	o	LCD data output							
	I2S_SDO	o	I ² S serial data output							
43	PB1	I/o	I/O port (default)	-	125	C7	P2	LVCMOS Schmitt	Type 1	100k PUC (dis)
	FPDAT9	o	LCD data output							
	I2S_WS	o	I ² S word select signal output							
44	PB2	I/o	I/O port (default)	-	61	L9				
	FPDAT10	o	LCD data output							
	I2S_SCLK	o	I ² S serial bit clock output							
	PWM_H	o	T16P PWM_H signal output							
45	PB3	I/o	I/O port (default)	-	62	M10				
	FPDAT11	o	LCD data output							
	I2S_MCLK	o	I ² S master clock output							
	PWM_L	o	T16P PWM_L signal output							
46	PB4	I/o	I/O port (default)	-	82	J12				
	FPDAT12	o	LCD data output							
	FPDAT20	o	LCD data output							
47	PB5	I/o	I/O port (default)	-	81	K12				
	FPDAT13	o	LCD data output							
	FPDAT21	o	LCD data output							
48	PB6	I/o	I/O port (default)	-	104	E11				
	FPDAT14	o	LCD data output							
	FPDAT22	o	LCD data output							
49	PB7	I/o	I/O port (default)	-	105	D11				
	FPDAT15	o	LCD data output							
	FPDAT23	o	LCD data output							

Table 1.3.2.5 List of USB Pins

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
1	USBDP	I/o	USB D+ pin	67	75	M14	P2	-	-	-
2	USBDM	I/o	USB D- pin	66	74	N14	P2	-	-	-
3	USBVBUS	I	USB VBUS pin. Allows input of 5 V	68	76	L14	P2	-	-	-

Table 1.3.2.6 List of Debug Control Pins

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
1	DCLK	I/O (H)	DCLK signal output for debugging (default)	25	27	K1	P2	LVCMOS Schmitt	Type 2	50k PUC (dis)
	P34	i/o	I/O port							
2	DSIO	I/o	Serial input/output for debugging (default)	23	25	J1	P2	LVCMOS Schmitt	Type 2	50k PUC (en)
	P35	i/o	I/O port							
3	DST2	I/O (L)	DST2 signal output for debugging (default)	24	26	J2	P2	LVCMOS Schmitt	Type 2	100k PUC (dis)
	P36	i/o	I/O port							
4	DST1	I/O (H)	DST1 signal output for debugging (default)	16	18	G2				
	P16	i/o	I/O port							
	FPDAT14	o	LCD data output							

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
5	DST0	i/O (H)	DST0 signal output for debugging (default)	15	17	G1	P2	LVCMOS Schmitt	Type 2	100k P _{Uc} (dis)
	P15	i/o	I/O port							
	FPDAT13	o	LCD data output							
6	DPCO	i/O (H)	DPCO signal output for debugging (default)	17	19	G3				
	P17	i/o	I/O port							
	FPDAT15	o	LCD data output							

Table 1.3.2.7 List of Other Pins

No.	Pin name	I/O	Description	Pin No.			PWR	DC characteristics		
				TQFP15 128	TQFP24 144	PFBGA 180		Input	Output	PU/PD
1	#RESET	I	Reset input	26	30	L1	P2	LVCMOS Schmitt	-	50k PU
2	#NMI	I	NMI request input	27	31	L2				
3	BOOT	I	Boot mode select signal input	6	6	D2				
4	WAKEUP	O (X)	C33 wakeup signal output from RTC	-	41	M3	P3	-	Type 1	-
5	#STBY	I	C33 standby input (except for RTC)	37	42	M4	P3	LVCMS Schmitt	-	-
6	TEST	I	Test input. Connect to V _{ss} in user mode.	7	7	D1	P1	LVCMS	-	120k PD
7	VCP	O (H)	Test output	46	51	N7	P5	-	-	-

Table 1.3.2.8 USI Pin Configuration

Pin name	Function by interface mode				
	UART	SPI master	SPI slave	I ² C master	I ² C slave
USI_DI	Data input (uart_rx)	Data input (spi_di)	Data input (spi_di)	Data input/output (i2c_sda)	Data input/output (i2c_sda)
USI_DO	Data output (uart_tx)	Data output (spi_do)	Data output (spi_do) when #spi_ss = 0 Hi-Z when #spi_ss = 1	Unused (output)	Unused (output)
USI_CK	Unused (input)	Clock output (spi_ck)	Clock input (spi_ck)	Clock input/output (i2c_sck)	Clock input/output (i2c_sck)
USI_CS	Unused (input)	Unused (input)	Slave select input (#spi_ss)	Data input/output (i2c_sda)	Data input/output (i2c_sda)

Note: In I²C mode, both the USI_DI and USI_CS pins can be configured as I²C data input/output pins. However, they can not be used for I²C data input/output at the same time.

Table 1.3.2.9 USIL Pin Configuration

Pin name	Function by interface mode						
	UART	SPI master	SPI slave	I ² C master	I ² C slave	LCD Parallel	LCD SPI
USIL_DI	Data input (uart_rx)	Data input (spi_di)	Data input (spi_di)	Data input/output (i2c_sda)	Data input/output (i2c_sda)	A0 (command/data select signal) output (lcdp_a0)	A0 (command/data select signal) output (lcds_a0)
USIL_DO	Data output (uart_tx)	Data output (spi_do)	Data output (spi_do) when #spi_ss = 0 Hi-Z when #spi_ss = 1	Unused (output)	Unused (output)	Write signal output (lcdp_wr)	Data output (lcds_do)
USIL_CK	Unused (input)	Clock output (spi_ck)	Clock input (spi_ck)	Clock input/output (i2c_sck)	Clock input/output (i2c_sck)	Read signal output (lcdp_rd)	Clock output (lcds_ck)
USIL_CS	Unused (input)	Unused (input)	Slave select input (#spi_ss)	Data input/output (i2c_sda)	Data input/output (i2c_sda)	Chip select signal output (lcdp_cs)	Unused (input)
LCD_D[7:0]	Unused (input)	Unused (input)	Unused (input)	Unused (input)	Unused (input)	8-bit data input/output	Unused (input)

Note: In I²C mode, both the USIL_DI and USIL_CS pins can be configured as I²C data input/output pins. However, they can not be used for I²C data input/output at the same time.

1 OVERVIEW

Table 1.3.2.10 FSIO Pin Configuration

FSIO ch	Pin name	Function by transfer mode			
		Async. (internal clock)	Async. (external clock)	Clock sync master	Clock sync slave
Ch.0	SIN0	Data input	Data input	Data input	Data input
	SOUT0	Data output	Data output	Data output	Data output
	SCLK0	Unused (input)	Clock input	Clock output	Clock input
	#SRDY0	Unused (input)	Unused (input)	Ready signal input	Ready signal output
Ch.1	SIN1	Data input	Data input	Data input	Data input
	SOUT1	Data output	Data output	Data output	Data output
	SCLK1	Unused (input)	Clock input	Clock output	Clock input
	#SRDY1	Unused (input)	Unused (input)	Ready signal input	Ready signal output

1.3.3 Package

TQFP15-128pin Package

(Unit: mm)

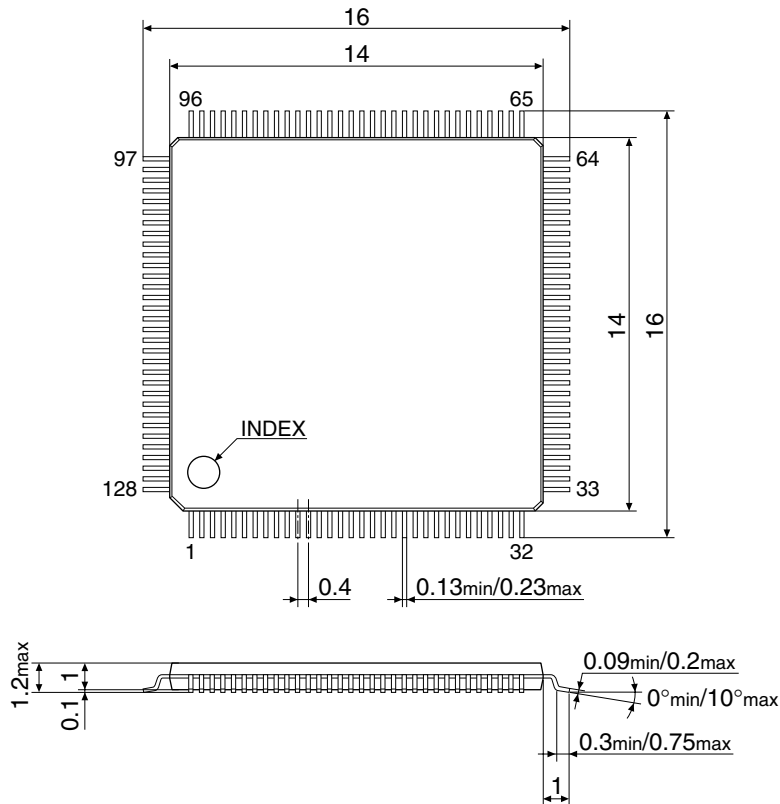


Figure 1.3.3.1 TQFP15-128pin Package Dimensions

TQFP24-144pin Package

(Unit: mm)

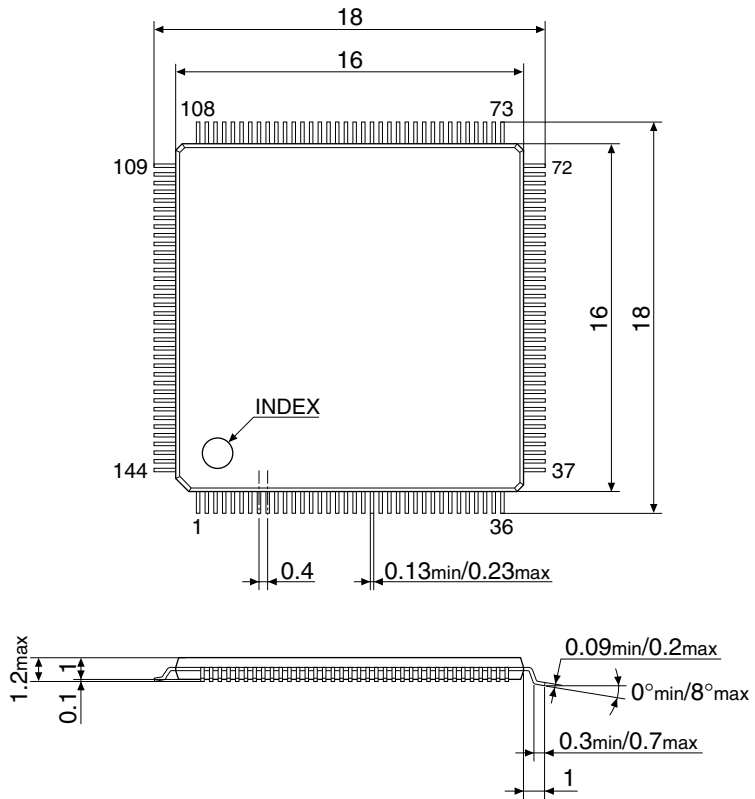
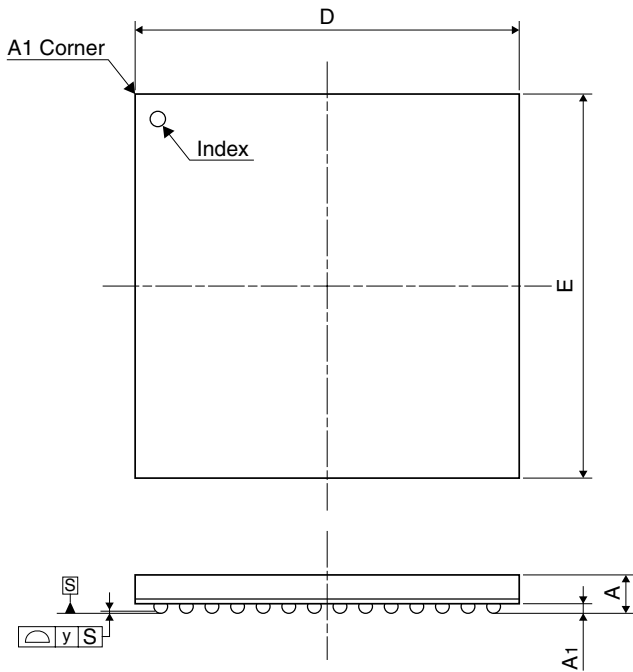


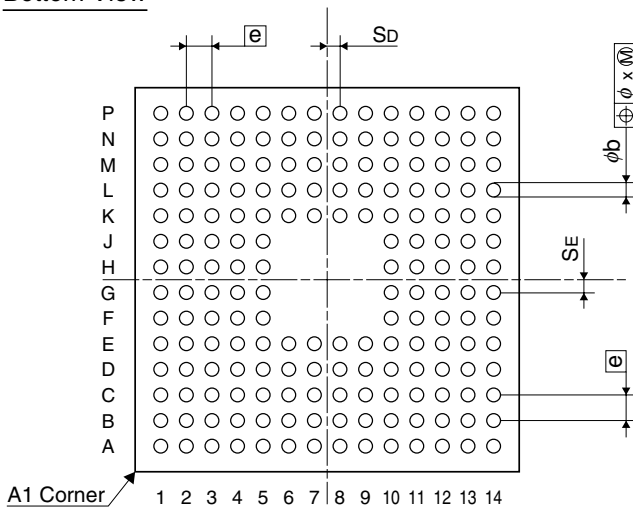
Figure 1.3.3.2 TQFP24-144pin Package Dimensions

PFBGA12U-180 Package

Top View



Bottom View



Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	-	12	-
E	-	12	-
A	-	-	1.2
A1	-	0.3	-
e	-	0.8	-
b	0.38	-	0.48
x	-	-	0.08
y	-	-	0.1
SD	-	0.4	-
SE	-	0.4	-

Figure 1.3.3.3 PFBGA12U-180 Package Dimensions

1.3.4 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).

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

Make sure that the chip temperature (T_j) is 125°C or less when the USB function controller is not used, or 100°C or less while the USB function controller is operating.

Thermal resistance of the TQFP package

1. When mounted on a board (windless condition)

Thermal resistance (θ_{j-a}) = 39°C/W (TQFP24), 42°C/W (TQFP15)

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)

Thermal resistance = 90–100°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 package

1. When mounted on a board (windless condition)

Thermal resistance (θ_{j-a}) = 24°C/W (PFBGA12U)

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)

Thermal resistance = 165°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.

2 CPU

The S1C33L26 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.”

2.1 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

- Depends 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 cycle |
| • 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

Coprocessor interface

- 16-bit ÷ 16-bit divider

2.2 CPU Registers

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

Special registers		General-purpose registers	
bit 31	bit 0	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 2.2.1 Registers

2.3 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 2.3.1 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
	jrgt.d		Delayed branching possible
	jrge	$sign8$	PC relative conditional jump
	jrge.d		Delayed branching possible
	jrlt	$sign8$	PC relative conditional jump
	jrlt.d		Delayed branching possible
	jrle	$sign8$	PC relative conditional jump
	jrle.d		Delayed branching possible
jrujt	$sign8$	PC relative conditional jump	
jrujt.d		Delayed branching possible	
jruga	$sign8$	PC relative conditional jump	
jruga.d		Delayed branching possible	

Classification	Mnemonic	Function		
Branch	jrult jrult.d	<i>sign8</i> PC relative conditional jump Delayed branching possible	Branch condition: C	
	jrule jrule.d	<i>sign8</i> PC relative conditional jump Delayed branching possible	Branch condition: Z C	
	jreq jreq.d	<i>sign8</i> PC relative conditional jump Delayed branching possible	Branch condition: Z	
	jrne jrne.d	<i>sign8</i> PC relative conditional jump Delayed branching possible	Branch condition: !Z	
	jp jp.d	<i>sign8</i> <i>%rb</i> Absolute jump	Delayed branching possible	
	call call.d	<i>sign8</i> <i>%rb</i> Absolute subroutine call	Delayed call possible	
	ret ret.d	 Subroutine return Delayed return possible		
	reti	Return from interrupt or exception handling		
	ret.d	Return from the debug processing routine		
	int	<i>imm2</i> Software exception		
	brk	Debug exception		
	Data transfer	ld.b	<i>%rd, %rs</i> General-purpose register (byte) → general-purpose register (sign-extended)	
			<i>%rd, [%rb]</i> Memory (byte) → general-purpose register (sign-extended)	
			<i>%rd, [%rb]+</i> Postincrement possible	
<i>%rd, [%sp+imm6]</i> Stack (byte) → general-purpose register (sign-extended)				
<i>[%rb], %rs</i> General-purpose register (byte) → memory				
<i>[%rb]+, %rs</i> Postincrement possible				
<i>[%sp+imm6], %rs</i> General-purpose register (byte) → stack				
ld.ub		<i>%rd, %rs</i> General-purpose register (byte) → general-purpose register (zero-extended)		
		<i>%rd, [%rb]</i> Memory (byte) → general-purpose register (zero-extended)		
		<i>%rd, [%rb]+</i> Postincrement possible		
		<i>%rd, [%sp+imm6]</i> Stack (byte) → general-purpose register (zero-extended)		
ld.h		<i>%rd, %rs</i> General-purpose register (halfword) → general-purpose register (sign-extended)		
		<i>%rd, [%rb]</i> Memory (halfword) → general-purpose register (sign-extended)		
		<i>%rd, [%rb]+</i> Postincrement possible		
		<i>%rd, [%sp+imm6]</i> Stack (halfword) → general-purpose register (sign-extended)		
		<i>[%rb], %rs</i> General-purpose register (halfword) → memory		
		<i>[%rb]+, %rs</i> Postincrement possible		
		<i>[%sp+imm6], %rs</i> General-purpose register (halfword) → stack		
ld.uh		<i>%rd, %rs</i> General-purpose register (halfword) → general-purpose register (zero-extended)		
		<i>%rd, [%rb]</i> Memory (halfword) → general-purpose register (zero-extended)		
		<i>%rd, [%rb]+</i> Postincrement possible		
		<i>%rd, [%sp+imm6]</i> Stack (halfword) → general-purpose register (zero-extended)		
ld.w		<i>%rd, %rs</i> General-purpose register (word) → general-purpose register		
		<i>%rd, sign6</i> Immediate → general-purpose register (sign-extended)		
		<i>%rd, [%rb]</i> Memory (word) → general-purpose register		
		<i>%rd, [%rb]+</i> Postincrement possible		
		<i>%rd, [%sp+imm6]</i> Stack (word) → general-purpose register		
		<i>[%rb], %rs</i> General-purpose register (word) → memory		
	<i>[%rb]+, %rs</i> Postincrement possible			
	<i>[%sp+imm6], %rs</i> General-purpose register (word) → stack			
System control	nop	No operation		
	halt	HALT		
	slp	SLEEP		
Immediate extension	ext	<i>imm13</i> Extend operand in the following instruction		
Bit manipulation	bst	<i>[%rb], imm3</i> Test a specified bit in memory data		
	bclr	<i>[%rb], imm3</i> Clear a specified bit in memory data		
	bset	<i>[%rb], imm3</i> Set a specified bit in memory data		
	bnot	<i>[%rb], imm3</i> Invert a specified bit in memory data		
Other	swap	<i>%rd, %rs</i> Bitwise swap on byte boundary in word		
	pushn	<i>%rs</i> Push general-purpose registers <i>%rs</i> – <i>%r0</i> onto the stack		
	popn	<i>%rd</i> Pop data for general-purpose registers <i>%rd</i> – <i>%r0</i> off the stack		

Table 2.3.2 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 2.3.3 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	$\%rd, \%rs$ Byte-wise 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 2.3.4 Instructions Removed

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

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

Table 2.3.5 Symbol Meanings

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

2.4 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 for each instruction.

Forcible Break

A debug exception is generated by an external input signal.

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 → 0x60008
R0 → 0x6000c
- (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.

2.5 Chip ID

The S1C33L26 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]/0x20008)

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 S1C33L26 has adopted the C33 PE little endian core, so the chip core ID is 0x06.

Product Series ID Bits (D[7:0]/0x20009)

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

ID	Product Series
0x03	S1C33xx Series
0x04	S1C334xx Series
0x0E	S1C33Exx Series
0x15	S1C33Lxx Series

The product series ID of the S1C33L26 is 0x15.

Model ID Bits (D[7:0]/0x2000a)

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

The model ID of the S1C33L26 is 0x26.

Version Bits (D[7:0]/0x2000b)

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

0x00 is a version number.

3 Memory Map

Figure 3.1 shows a memory map of the entire S1C33L26 address space. Figure 3.2 shows a memory map of the embedded memory and the internal I/O area.

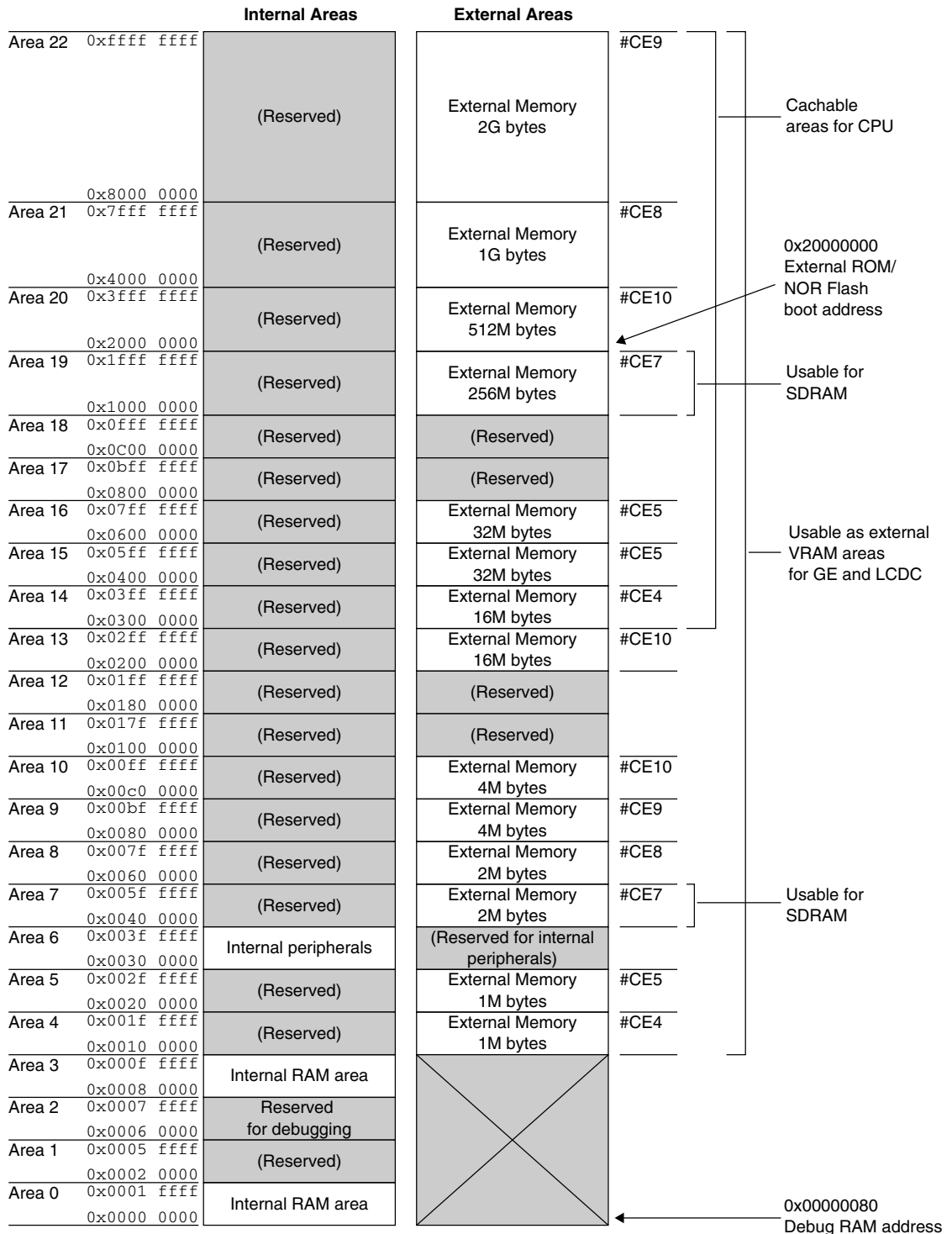


Figure 3.1 Entire Memory Map

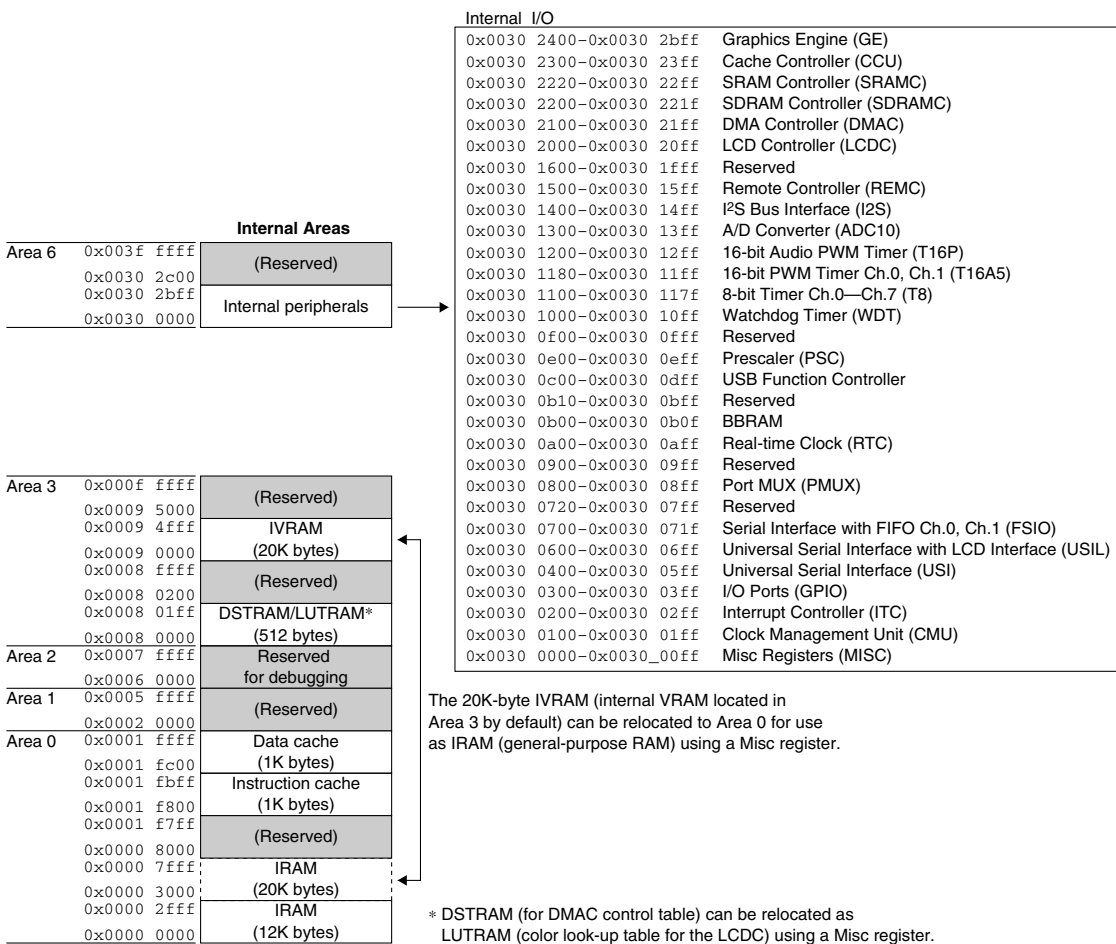


Figure 3.2 Internal Area Map

The following describes the area configuration of the S1C33L26.

3.1 Boot Address

When the S1C33L26 is powered on or reset, the system boots up from a NOR Flash/external ROM, SPI-EEPROM, or a PC connected via RS232C interface specified using the BOOT and #CE10 pins.

Table 3.1.1 lists the pin status and the boot mode selected. For more information on booting, see the “Boot” section in Appendix.

Table 3.1.1 Boot Mode Settings

BOOT pin	#CE10 pin	Boot mode	Program execution start address
0	Output	NOR Flash/external ROM	The system starts executing from the address written at address 0x20000000.
1	1 (Input)	SPI-EEPROM	The system loads MBR to IRAM (from address 0x100) and starts executing the code loaded.
	0 (Input)	PC RS232C	

Note: The #CE10 pin includes a pull-up resistor and it is enabled at initial reset. Note, however, that the #CE10 pin is configured as an input pin and its pull-up resistor is disabled in the initial process by the boot sequencer when the BOOT pin is set to 1. Therefore, connect an external pull-up or pull-down resistor to set the #CE10 pin input level to 1 or 0.

3.2 Area 0 (IRAM, Cache Memory)

3.2.1 IRAM

Area 0 includes a 12K-byte IRAM (internal RAM) that can be used as a general-purpose RAM to store data and execute instructions. It is located at addresses 0x0 to 0x2fff.

In addition to the 12K-byte IRAM, the 20K-byte IVRAM (internal VRAM), which is located in Area 3 by default, can be relocated to Area 0 to use it as a general-purpose RAM. Write 0 to IVRAM_LOC/MISC_IRAM_LOC register when using IVRAM as an additional IRAM. (See the “Misc Registers (MISC)” chapter for more information on the register.)

In most cases, the CPU accesses IRAM (excluding IVRAM relocated to Area 0) in one cycle regardless of access data size. While the IVRAM (relocated to Area 0) access cycle can be configured as one or two cycles according to the COREWT/MISC_RAMWT register setting. (See the “Misc Registers (MISC)” chapter for more information on the register.)

Note: The address range from 0x0 to 0xff (256 bytes) in IRAM is reserved for use as the debugging RAM area. Be sure to avoid accessing this area from the user program and the debugger. When using a debugger, specify 0x80 as the debugging RAM address.

```
c33 das 0x60000 0x80
```

3.2.2 Cache Memory

Area 0 also includes two 1K-byte RAMs, one of them is located at addresses 0x1f800 to 0x1fbff and can be used as an instruction cache, and another is located at addresses 0x1fc00 to 0x1ffff and can be used as a data cache. Each 1K-byte RAM is enabled as a cache memory by setting the cache controller. For more information on the caches, see the “Cache Controller (CCU)” chapter.

3.3 Areas 1 and 2 (Reserved for System)

Areas 1 and 2 are reserved for the system. Be sure to avoid accessing these areas from the user program and the debugger.

3.4 Area 3 (IVRAM, DSTRAM)

3.4.1 IVRAM

Area 3 includes a 20K-byte IVRAM (internal VRAM) for the LCD controller and graphics engine. It is located at addresses 0x90000 to 0x94fff.

The 20K-byte IVRAM can be relocated to Area 0 to use it as a general-purpose RAM by writing 0 to IVRAM_LOC/MISC_IRAM_LOC register. IVRAM_LOC must be set to 1 (default) when using IVRAM. (See the “Misc Registers (MISC)” chapter for more information on the register.)

IVRAM located in Area 3 is accessed in four or five cycles, according to the BUSWT/MISC_RAMWT register setting.

3.4.2 DSTRAM

Area 3 includes a 512-byte RAM (DSTRAM) located at addresses 0x80000 to 0x801ff.

DSTRAM is used to store the control table for the DMA controller. It can also be used as a general-purpose RAM.

When using the color look-up table function for the LCD, DSTRAM is configured as the look-up table memory (LUTRAM) by setting DSTRAM_CFG/MISC_IRAM_LOC register and LUTPASS/LCDC_DISPDMOD register. In this case, DSTRAM cannot be accessed from the CPU and DMAC. For more information on the look-up table and memory switching, see the “LCD Controller (LCDC)” chapter.

Note: When DSTRAM is switched to LUTRAM, locate the DMAC control table in IVRAM (Area 3) or an external RAM.

3.5 Area 6 (I/O Area)

Area 6 is allocated to the I/O area for the internal peripheral circuits.

For details on the internal peripheral circuits mapped to this area, see the descriptions of each peripheral module. For the list of control registers, see the “List of I/O Registers” section in Appendix.

Area 6 includes BBRAM (16 bytes) located at addresses 0x300b00 to 0x300b0f. BBRAM operates with a power source (RTC_{VDD}) separated from the system power. Thus, BBRAM can retain data even if the system power is off.

3.6 External Memory Area

Areas 4, 5, 7 to 10, 13 to 16, and 19 to 22 can be used for external memory and other external devices. Configure the SRAMC and/or SDRAMC according to the devices connected. Although the internal address and internal data buses of the S1C33L26 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 26 bits (A[25:0]) due to the limited number of pins available.

- Notes:**
- A NAND Flash can be connected to Area 9 (4M bytes) or Area 22 (2G bytes).
 - An SDRAM can be connected to Area 7 (2M bytes) or Area 19 (256M bytes).
 - The external VRAM used for the graphics engine and LCD controller can be connected to Areas 4 to 22.

3.7 Bus Masters and Accessible Memories

The table below lists the bus masters and the memories that can be accessed.

Table 3.7.1 Bus Masters and Accessible Memories

Memory	Bus master			
	CPU	DMAC	LCDC	GE
IRAM (Area 0)	✓	–	–	–
IVRAM (Area 0)	✓	–	–	–
IVRAM (Area 3)	✓	✓	✓	✓
DSTRAM (Area 3)	✓	✓	–	–
LUTRAM (LCDC module)	–	–	✓	–
I/O registers (Area 6)	✓	✓	–	*
External memory	✓	✓	✓	✓

✓: Can be accessed. –: Cannot be accessed.

*: The GE can only write data to the USIL transmit data buffer via the BLKCOPY command.

3.8 Memory Access Rate

(1) Internal RAM and external SRAM access rate

Table 3.8.1 Static Memory Access Cycle

Memory access cycle	Number of access cycles				
	Single byte	Single half-word	Single word	Successive address	Burst access
IRAM (12KB/Area 0) read	2	2	2	1	–
IRAM (12KB/Area 0) write	1	1	1	–	–
IRAM (20KB/Area 0) read	2 + W	2 + W	2 + W	1 or 1 + W (*1)	–
IRAM (20KB/Area 0) write	1	1	1	–	–
Cache RAM read	2	2	2	–	–
Cache RAM write	1	1	1	–	–
DSTRAM (512B/Area 3) read	4	4	4	–	–
DSTRAM (512B/Area 3) write	4	4	4	–	–
IVRAM (20KB/Area 3) read	4 + W	4 + W	4 + W	–	1 + W (*2)
IVRAM (20KB/Area 3) write	4	4	4	–	–

Memory access cycle	Number of access cycles				
	Single byte	Single half-word	Single word	Successive address	Burst access
Area 6 register (8-bit) read	6	9	15	–	–
Area 6 register (8-bit) write	3	6	12	–	–
Area 6 register (16-bit) read	6	6	9	–	–
Area 6 register (16-bit) write	3	3	6	–	–
Area 6 register (32-bit) read	6	6	6	–	–
Area 6 register (32-bit) write	3	3	3	–	–
External SRAM/ROM (8-bit) read/write	4 + S + W + H	6 + (S + W + H) × 2	8 + (S + W + H) × 4	–	(*3)
External SRAM/ROM (16-bit) read/write	4 + S + W + H	4 + S + W + H	6 + (S + W + H) × 2	–	(*3)

W: Number of WAIT cycles (0 min.) S: Number of SETUP cycles (1 min.) H: Number of HOLD cycles (1 min.)

(Note)

- *1: 1 cycle if there is an idle cycle inserted before starting a successive read. Otherwise, just reduce the number of single access cycle by 1.
- *2: The cache controller does not support IVRAM, so burst read will be issued by the LCDc when IVRAM is used as a VRAM or by the DMAC when used as a DMA control table. Add (1 + W) cycles to single word access cycles.
- *3: Burst access can be regarded as 4 or 8 single word access cycles. However, refer to item (2) below for the external SRAM/ROM access cycle when the SDRAMC is enabled and it is in self-refresh mode.

(2) External SRAM access rate while the SDRAM is in self-refresh status

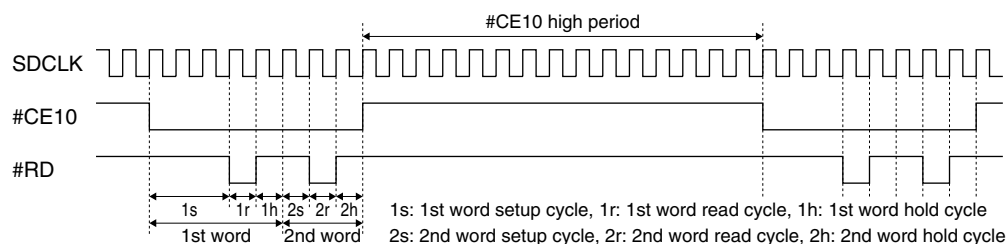
Table 3.8.2 lists the number of external SRAM access cycles when SDON/SDRAMC_INIT register is set to 1 (SDRAMC enabled) and SELEN/SDRAMC_REF register is set to 1 (self-refresh enabled).

When SDON is set to 0 (SDRAMC disabled) or SELEN is set to 0 (self-refresh disabled), the SRAM will be accessed according to the setup, hold, and wait cycle conditions set using the SRAMC register.

The S1C33L26 is designed under the assumption that the CPU is set in HALT status and the LCDc only is active while the SDRAM is placed into self-refresh mode.

The SDRAMC should be disabled by setting SDON to 0 if no SDRAM is used.

The T80NS[3:0]/SDRAMC_CFG register value (TRC, TRFC, TXSR) should be set to 0x0, 0x4, 0x8, or 0xc.



(Setting example: T80NS[3:0] = 0xe, CE10SETUP[1:0] = 0x0, CE10HOLD[1:0] = 0x0, CE10WAIT[3:0] = 0x0)

Figure 3.8.1 SRAM Access Cycle during SDRAM Self-Refresh

Table 3.8.2 SRAM Access Cycle during SDRAM Self-Refresh

(Condition: No other access sources, Unit: cycle)

SDRAMC T80NS [3:0]	SRAMC			#CE10 high period	1st word			2nd word		
	CExSETUP [1:0]	CExWAIT [3:0]	CExHOLD [1:0]		Setup cycle	Read cycle	Hold cycle	Setup cycle	Read cycle	Hold cycle
0x0	0x0	W	H	5	1	W + 1	H + 1	1	W + 1	H + 1
0x1	0x0	W	H	5	4 (*3)	W + 1	H + 1	1	W + 1	H + 1
0x2	0x0	W	H	5	3 (*3)	W + 1	H + 1	1	W + 1	H + 1
0x3	0x0	W	H	5	2 (*3)	W + 1	H + 1	1	W + 1	H + 1
0x4	0x0	W	H	5	1	W + 1	H + 1	1	W + 1	H + 1
0x5	0x0	W	H	6 (*1)	4 (*3)	W + 1	H + 1	1	W + 1	H + 1
0x6	0x0	W	H	7 (*1)	3 (*3)	W + 1	H + 1	1	W + 1	H + 1
0x7	0x0	W	H	8 (*1)	2 (*3)	W + 1	H + 1	1	W + 1	H + 1
0x8	0x0	W	H	9 (*1)	1	W + 1	H + 1	1	W + 1	H + 1
0x9	0x0	W	H	10 (*1)	4 (*3)	W + 1	H + 1	1	W + 1	H + 1
0xa	0x0	W	H	11 (*1)	3 (*3)	W + 1	H + 1	1	W + 1	H + 1

3 MEMORY MAP

SDRAMC	SRAMC			#CE10 high period	1st word			2nd word		
	T80NS [3:0]	CExSETUP [1:0]	CExWAIT [3:0]		CExHOLD [1:0]	Setup cycle	Read cycle	Hold cycle	Setup cycle	Read cycle
0xb	0x0	W	H	12 (*1)	2 (*3)	W + 1	H + 1	1	W + 1	H + 1
0xc	0x0	W	H	13 (*1)	1	W + 1	H + 1	1	W + 1	H + 1
0xd	0x0	W	H	14 (*1)	4 (*3)	W + 1	H + 1	1	W + 1	H + 1
0xe	0x0	W	H	15 (*1)	3 (*3)	W + 1	H + 1	1	W + 1	H + 1
0xf	0x0	W	H	16 (*1)	2 (*3)	W + 1	H + 1	1	W + 1	H + 1
0x0	0x1	W	H	5	2	W + 1	H + 1	2	W + 1	H + 1
0x1	0x1	W	H	5	1 (*2)	W + 1	H + 1	2	W + 1	H + 1
0x2	0x1	W	H	5	4 (*3)	W + 1	H + 1	2	W + 1	H + 1
0x3	0x1	W	H	5	3 (*3)	W + 1	H + 1	2	W + 1	H + 1
0x4	0x1	W	H	5	2	W + 1	H + 1	2	W + 1	H + 1
0x5	0x1	W	H	6 (*1)	1 (*2)	W + 1	H + 1	2	W + 1	H + 1
0x6	0x1	W	H	7 (*1)	4 (*3)	W + 1	H + 1	2	W + 1	H + 1
0x7	0x1	W	H	8 (*1)	3 (*3)	W + 1	H + 1	2	W + 1	H + 1
0x8	0x1	W	H	9 (*1)	2	W + 1	H + 1	2	W + 1	H + 1
0x9	0x1	W	H	10 (*1)	1 (*2)	W + 1	H + 1	2	W + 1	H + 1
0xa	0x1	W	H	11 (*1)	4 (*3)	W + 1	H + 1	2	W + 1	H + 1
0xb	0x1	W	H	12 (*1)	3 (*3)	W + 1	H + 1	2	W + 1	H + 1
0xc	0x1	W	H	13 (*1)	2	W + 1	H + 1	2	W + 1	H + 1
0xd	0x1	W	H	14 (*1)	1 (*2)	W + 1	H + 1	2	W + 1	H + 1
0xe	0x1	W	H	15 (*1)	4 (*3)	W + 1	H + 1	2	W + 1	H + 1
0xf	0x1	W	H	16 (*1)	3 (*3)	W + 1	H + 1	2	W + 1	H + 1
0x0	0x2	W	H	5	3	W + 1	H + 1	3	W + 1	H + 1
0x1	0x2	W	H	5	2 (*2)	W + 1	H + 1	3	W + 1	H + 1
0x2	0x2	W	H	5	1 (*2)	W + 1	H + 1	3	W + 1	H + 1
0x3	0x2	W	H	5	4 (*3)	W + 1	H + 1	3	W + 1	H + 1
0x4	0x2	W	H	5	3	W + 1	H + 1	3	W + 1	H + 1
0x5	0x2	W	H	6 (*1)	2 (*2)	W + 1	H + 1	3	W + 1	H + 1
0x6	0x2	W	H	7 (*1)	1 (*2)	W + 1	H + 1	3	W + 1	H + 1
0x7	0x2	W	H	8 (*1)	4 (*3)	W + 1	H + 1	3	W + 1	H + 1
0x8	0x2	W	H	9 (*1)	3	W + 1	H + 1	3	W + 1	H + 1
0x9	0x2	W	H	10 (*1)	2 (*2)	W + 1	H + 1	3	W + 1	H + 1
0xa	0x2	W	H	11 (*1)	1 (*2)	W + 1	H + 1	3	W + 1	H + 1
0xb	0x2	W	H	12 (*1)	4 (*3)	W + 1	H + 1	3	W + 1	H + 1
0xc	0x2	W	H	13 (*1)	3	W + 1	H + 1	3	W + 1	H + 1
0xd	0x2	W	H	14 (*1)	2 (*2)	W + 1	H + 1	3	W + 1	H + 1
0xe	0x2	W	H	15 (*1)	1 (*2)	W + 1	H + 1	3	W + 1	H + 1
0xf	0x2	W	H	16 (*1)	4 (*3)	W + 1	H + 1	3	W + 1	H + 1
0x0	0x3	W	H	5	4	W + 1	H + 1	4	W + 1	H + 1
0x1	0x3	W	H	5	3 (*2)	W + 1	H + 1	4	W + 1	H + 1
0x2	0x3	W	H	5	2 (*2)	W + 1	H + 1	4	W + 1	H + 1
0x3	0x3	W	H	5	1 (*2)	W + 1	H + 1	4	W + 1	H + 1
0x4	0x3	W	H	5	4	W + 1	H + 1	4	W + 1	H + 1
0x5	0x3	W	H	6 (*1)	3 (*2)	W + 1	H + 1	4	W + 1	H + 1
0x6	0x3	W	H	7 (*1)	2 (*2)	W + 1	H + 1	4	W + 1	H + 1
0x7	0x3	W	H	8 (*1)	1 (*2)	W + 1	H + 1	4	W + 1	H + 1
0x8	0x3	W	H	9 (*1)	4	W + 1	H + 1	4	W + 1	H + 1
0x9	0x3	W	H	10 (*1)	3 (*2)	W + 1	H + 1	4	W + 1	H + 1
0xa	0x3	W	H	11 (*1)	2 (*2)	W + 1	H + 1	4	W + 1	H + 1
0xb	0x3	W	H	12 (*1)	1 (*2)	W + 1	H + 1	4	W + 1	H + 1
0xc	0x3	W	H	13 (*1)	4	W + 1	H + 1	4	W + 1	H + 1
0xd	0x3	W	H	14 (*1)	3 (*2)	W + 1	H + 1	4	W + 1	H + 1
0xe	0x3	W	H	15 (*1)	2 (*2)	W + 1	H + 1	4	W + 1	H + 1
0xf	0x3	W	H	16 (*1)	1 (*2)	W + 1	H + 1	4	W + 1	H + 1

(Note)

- *1: Interval between #CE access cycles will be increased.
- *2: When SDON = 1 (SDRAMC enabled) and SELEN = 1 (self-refresh enabled), the SRAM setup cycle will be shorter than the CExSETUP[1:0] set value. Determine the setup time according to the characteristics of the device connected.
- *3: When SDON = 1 (SDRAMC enabled) and SELEN = 1 (self-refresh enabled), the SRAM setup cycle will be longer than the CExSETUP[1:0] set value. Determine the setup time according to the characteristics of the device connected.

(3) SDRAM access rate

Table 3.8.3 SDRAM Access Cycle

SDRAM access cycle	Number of access cycles			
	Byte/half-word	Word	4-word burst read	8-word burst read
CPU/DMAC/GE single random write	$5 + T24NS \times 2$	$6 + T24NS \times 2$	–	–
CPU/DMAC/GE single random read	$6 + T24NS \times 2 + CAS$	$6 + T24NS \times 2 + CAS$	–	–
CPU/DMAC/GE bulk read	$6 + CAS$	$6 + CAS$	–	–
Cache burst read (*1)	–	–	$11 + T24NS \times 2 + CAS$	–
DMAC burst read (*2)	–	–	$11 + T24NS \times 2 + CAS$	–
LCDC burst read (*3)	–	–	–	$19 + T24NS \times 2 + CAS$

T24NS: Value set in T24NS[1:0]/SDRAMC_CFG register CAS: CAS latency set in CAS[1:0]/SDRAMC_APP register

The values in the table do NOT take the following conditions into consideration:

- When the SDRAM is in auto-refresh or self-refresh status
- When SDRAM addresses across the 512-byte boundary are accessed

(Note)

- *1: When the CPU accesses an external SDRAM with caching enabled (CCU module)
- *2: When the DMA control table is located in an external SDRAM and the SDRAMC loads the control table during DMA transfer
- *3: When the LCDC loads display data stored in an external SDRAM (VRAM) to the FIFO
Note that the factor when the LCDC accesses the external SDRAM across the boundary between the main and sub-window areas (located in the same SDRAM) are not taken into consideration.

4 Power Supply

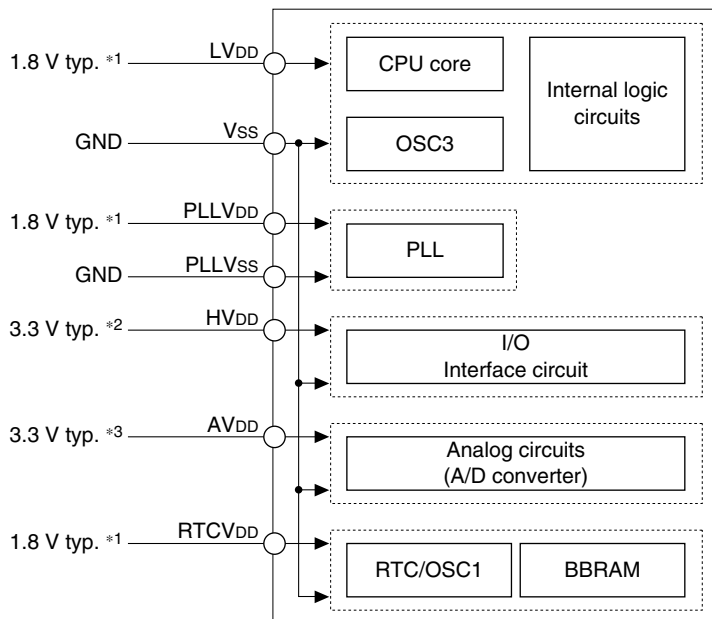
This section explains the operating voltage of the S1C33L26.

4.1 Power Supply Pins

The S1C33L26 has the power supply pins shown in Table 4.1.1.

Table 4.1.1 Power Supply Pins

No.	Pin name	I/O	Description	Pin No.			Power voltage
				TQFP15 128	TQFP24 144	PFBGA 180	
1	HV _{DD}	-	I/O power supply pin	1, 28, 58, 82, 109	1, 32, 65, 92, 121	C9, D4, D9, F11, G11, K4, L10, L11	3.3 V typ. (2.7 to 3.6 V or 3.0 to 3.6 V when the USB module is used.)
2	AV _{DD}	-	Analog power supply pin	38	43	M5	3.3 V typ. (2.7 to 3.6 V or 3.0 to 3.6 V)
3	LV _{DD}	-	Core power supply pin	14, 48, 70, 97, 118	16, 53, 78, 109, 132	C12, D6, D7, E4, F4, J11, L8, M8	1.8 V typ. (1.65 to 1.95 V or 1.7 to 1.9 V when a ce- ramic resonator is used.)
4	RTC _{VDD}	-	RTC/BBRAM power supply pin	33	37	N2	1.8 V typ. (= LV _{DD})
5	V _{SS}	-	Ground pin	8, 22, 36, 51, 65, 76, 91, 104, 113, 124	8, 24, 40, 56, 73, 86, 101, 116, 127, 138	B4, C10, D8, D10, E5, E6, E7, E8, E9, E10, F5, F10, G5, G10, H4, H5, H10, H11, J4, J5, J10, K5, K6, K7, K8, K9, K10, K11, L4, L5, L6, L7, M6, M13, N3, N8	GND
6	PLL _{VDD}	-	PLL power supply pin	47	52	M7	1.8 V typ. (= LV _{DD})
7	PLL _{VSS}	-	PLL power supply ground pin	45	50	P7	GND (= V _{SS})



*1: 1.65 to 1.95 V (1.7 to 1.9 V when a ceramic resonator is used.)

*2: 2.7 to 3.6 V (3.0 to 3.6 V when the USB module is used.)

*3: 2.7 to 3.6 V or 3.0 to 3.6 V

Figure 4.1.1 Power Supply System

4.2 Operating Voltage (LV_{DD})

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

LV_{DD} = 1.65 to 1.95 V (V_{SS} = GND)
or 1.7 to 1.9 V when a ceramic resonator is used

Note: The S1C33L26 packages have more than one LV_{DD} and V_{SS} pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.

4.3 Power Supply for PLL (PLL_{VDD}, PLL_{VSS})

The PLL power supply pins (PLL_{VDD} and PLL_{VSS}) are provided separately from the LV_{DD} and V_{SS} pins so that the digital circuits will not affect the PLL circuit. Supply the same voltage level as the LV_{DD} to the PLL_{VDD}.

PLL_{VDD} = LV_{DD}, PLL_{VSS} = 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.

4.4 Power Supply for RTC (RTC_{VDD})

The RTC power supply pin (RTC_{VDD}) is provided separately from the LV_{DD} pin in order to run the RTC and OSC1 oscillator at system power down. Supply the same voltage level as the LV_{DD} to the RTC_{VDD} pin.

RTC_{VDD} = LV_{DD} (V_{SS} = GND)

The RTC_{VDD} is also used for the battery-backup RAM (BDRAM).

4.5 I/O Interface Voltage (HV_{DD})

The HV_{DD} voltage is used for interfacing with external I/O signals. For the output interface of the S1C33L26, the HV_{DD} 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 LV_{DD}. The effective HV_{DD} voltage range is as follows:

HV_{DD} = 2.7 to 3.6 V (V_{SS} = GND)
or 3.0 to 3.6 V when the USB module is used

Note: The S1C33L26 packages have more than one HV_{DD} pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.

4.6 Power Supply for Analog Circuits (AV_{DD})

The analog power supply pin (AV_{DD}) is provided separately from other power supply pins so that the digital circuits will not affect the analog circuits (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 effective AV_{DD} voltage range is as follows:

AV_{DD} = 2.7 to 3.6 V (V_{SS} = GND)

Note: Be sure to supply a voltage within the range from 2.7 to 3.6 V to the AV_{DD} pin even if 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.

4.7 Precautions on Power Supply

Power-on sequence

To ensure that the device will operate normally, observe the timing requirements given below when turning the power on.

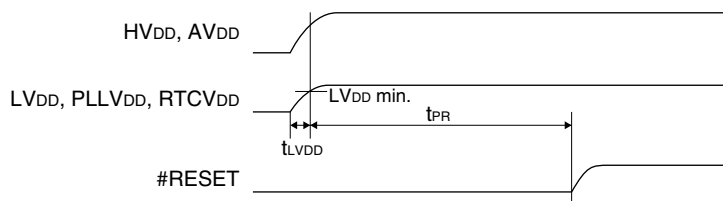


Figure 4.7.1 Power-On Sequence

- (1) t_{LVDD} : Elapsed time until the power supply stabilizes after power-on
Supply power in the following sequence.

Power-on: 1. LVDD, PLLVDD, (and RTCVDD)
2. HVDD, AVDD (May be applied with 1 above at the same time.)
3. Apply the input signal

* The RTCVDD can be always supplied to the chip to operate the RTC and BBRAM.

- (2) t_{PR} : Power-on-reset time
Keep the #RESET signal low for this period. See the “#RESET Pin” section in the “Reset and NMI” chapter for the power-on-reset time.

Power-off sequence

Shut off the power supply in the following sequence.

Power-off: 1. Turn off the input signal
2. HVDD, AVDD
3. LVDD, PLLVDD (and RTCVDD) (May be turned off with 2 above at the same time.)

Note: Be sure to avoid applying HVDD or AVDD for a duration of one second or more when the LVDD power is off, as a breakdown may occur in the device or the characteristics may be degraded due to flow-through current of the HVDD or AVDD.

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 LVDD and VSS and leading to breakage.

Latch-up occurs when the voltage applied to the input/output exceeds the rated value and a large current flows into the internal element, or when the voltage at the LVDD 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 LVDD and VSS once the device is in the latch-up condition. As this may result in heat generation or smoking, the following points must be taken into consideration:

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

5 Reset and NMI

5.1 Initial Reset

The S1C33L26 has two initial reset sources that initialize the internal circuits.

- (1) #RESET pin (external initial reset)
- (2) Watchdog timer (software selectable internal initial reset)

Figure 5.1.1 shows the configuration of the initial reset circuit.

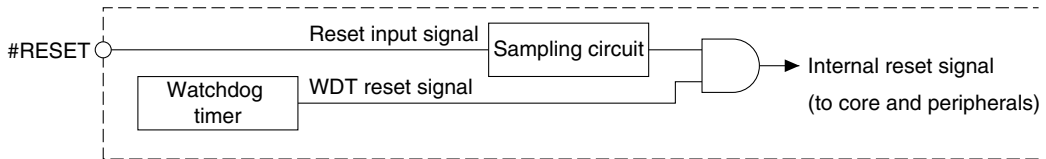


Figure 5.1.1 Configuration of Initial Reset Circuit

The CPU and peripheral circuits are initialized by the active signal from an initial reset source. When the reset signal is negated, the CPU starts reset handling. The reset handling reads the reset vector (reset handler start address) from the beginning of the vector table and starts executing the program (initial routine) beginning with the read address.

5.1.1 #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. And as shown in Figure 5.1.1.1, to assert the internal reset signal, low level 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 ensure that the chip is reset. Also the internal reset signal is negated when the default OSC3 oscillation stabilization wait time has elapsed after the #RESET pin goes high.

The S1C33L26 is reset by the low state (= 0) on the internal reset signal, and starts operating when the reset signal goes high (= 1).

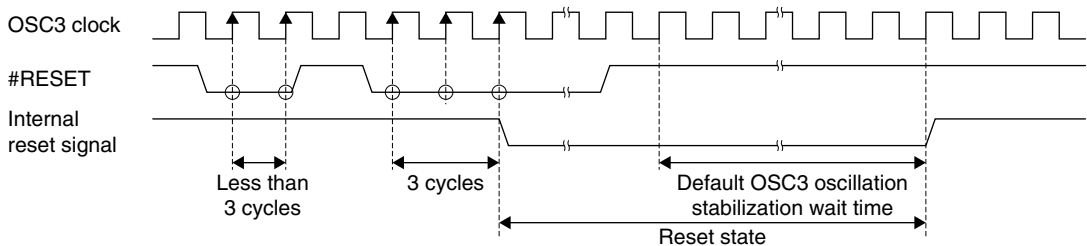


Figure 5.1.1.1 #RESET Sampling

5.1.2 Resetting by the Watchdog Timer

The S1C33L26 has a built-in watchdog timer to detect runaway of the CPU. The watchdog timer outputs a signal if it is not reset via software (due to CPU runaway) in the programmed cycles. The output signal can generate either NMI or reset. Write 1 to the RESEN/WDT_EN register to generate reset.

For details of the watchdog timer, see the “Watchdog Timer (WDT)” chapter.

- Notes:**
- When using the reset function of the watchdog timer, program the watchdog timer so that it will be reset within the programmed cycles to avoid occurrence of an unnecessary reset.
 - The reset function of the watchdog timer cannot be used for power-on reset as it must be enabled with software.

5.1.3 Initial Reset Sequence

Even if the #RESET pin input negates the reset signal after power is turned on, the CPU cannot boot up until the oscillation stabilization waiting time (128 / OSC3 clock frequency) has elapsed.

Figure 5.1.3.1 shows the operating sequence following cancellation of initial reset.

The CPU starts operating in synchronization with the OSC3 clock after reset state is canceled.

Note: The oscillation stabilization time described in this section does not include oscillation start time. Therefore the time interval until the CPU starts executing instructions after power is turned on or SLEEP mode is canceled may be longer than that indicated in the figure below.

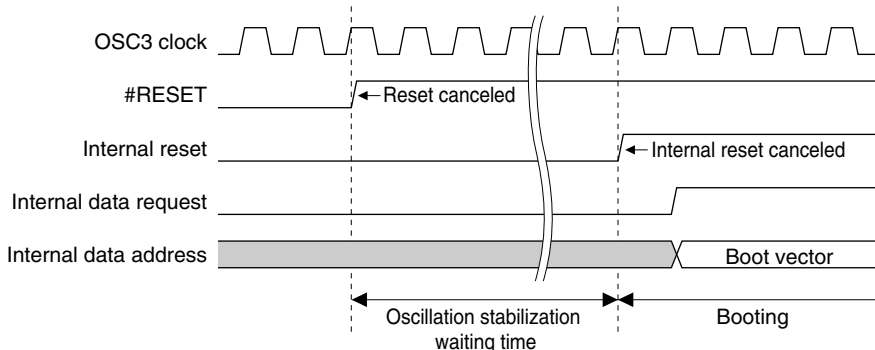


Figure 5.1.3.1 Operation Sequence Following Cancellation of Initial Reset

5.1.4 Initial Reset Status

The C33 PE Core and internal peripheral circuits are initialized while the internal reset signal is at kept 0. The following shows the reset status of the internal IC with the initial reset.

Table 5.1.4.1 Initial Reset Status

Item	Boot mode	Initial reset status
CPU - TTBR	NOR Flash/ external ROM boot	Initialized to 0x20000000
	SPI-EEPROM boot	
	PC RS232C boot	
	ICD debug	
CPU - PC	NOR Flash/ external ROM boot	The reset vector at address 0x20000000 is loaded to the PC.
	SPI-EEPROM boot	0x100 is loaded to the PC
	PC RS232C boot	
	ICD debug	0x20000 is loaded to the PC
CPU - PSR	-	Undefined
CPU - Other registers	-	Undefined
CPU - Operating clock	-	The CPU operates with OSC3 × 1/1 clock.
Oscillator circuits	-	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	-	Clocks are supplied to all the peripheral modules except LCDC and USB.
I/O pin state	-	I/O pins are initialized. (See the "Pin Functions" section in the "Overview" chapter.)
Other internal peripheral circuits	-	Initialized or undefined (See each I/O map.)

Note: The S1C33L26 does not support a hot reset feature that maintains I/O pin status and the TTBR value.

5.1.5 Precautions to be Taken during Initial Reset

Core CPU

When initially reset, all internal registers of the core CPU are undefined. 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 with hardware until data is written to the SP after an initial reset, to prevent erratic operation.

Internal RAM

The contents of internal RAM are undefined 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 negated, the CPU starts operating with the OSC3 clock. To prevent erratic operation due to an unstable 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 negated 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 unstable clock, the OSC1 clock should not be used only after this stabilization time elapsed.

I/O ports and I/O pins

Initial reset initializes the I/O port control and data registers, 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 undefined by initial reset. Therefore, setup of these registers with a program is required.

For the specific initial settings done on the peripheral circuits after an initial reset, see each I/O map or circuit descriptions.

5.2 NMI Input

The S1C33L26 has two NMI sources that generate NMI.

- (1) #NMI pin (external input)
- (2) Watchdog timer (software selectable)

Figure 5.2.1 shows the configuration of the NMI circuit.

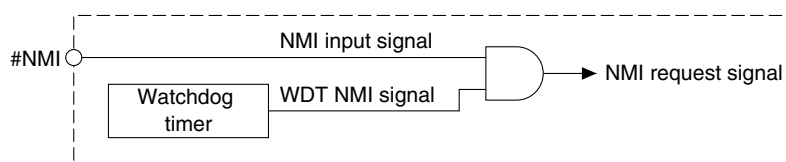


Figure 5.2.1 Configuration of NMI Circuit

The NMI signal, which is input from the #NMI pin or generated by the watchdog timer (WDT), generates a non-maskable interrupt to the C33 PE Core. This interrupt takes precedence over other interrupts and is unconditionally accepted by the C33 PE Core.

For details about NMI exception handling by the C33 PE Core, refer to the “S1C33 Family C33 PE Core Manual.”

5.2.1 #NMI Pin

The #NMI pin is used to generate a non-maskable interrupt to the C33 PE Core.

To generate an NMI (Non-Maskable Interrupt) to the C33 PE Core, the following two conditions must be met:

- (1) A negative edge is detected at the #NMI pin (the #NMI signal changes from a high to a low level).
- (2) The #NMI pin is maintained at a low level for three or more system clock cycles.

5.2.2 NMI by the Watchdog Timer

The S1C33L26 has a built-in watchdog timer to detect runaway of the CPU. The watchdog timer outputs a signal if it is not reset with software (due to CPU runaway) in the programmed cycles. The output signal can generate either NMI or reset. Write 1 to the NMIEN/WDT_EN register to generate NMI.

For details of the watchdog timer, see the “Watchdog Timer (WDT)” chapter.

6 Clock Management Unit (CMU)

6.1 CMU Module Overview

The CMU module controls the internal oscillators and the system clock.

The features of the CMU module are listed below.

- Generates the operating clocks with the built-in oscillators.
 - OSC3 oscillator circuit: 48 MHz (max.) crystal or ceramic oscillator circuit
Supports an external clock input.
 - OSC1 oscillator circuit: 32.768 kHz (typ.) crystal oscillator circuit
Supports an external clock input.
- Switches the system clock. The system clock source can be selected from OSC3, PLL, and OSC1 with software.
- Controls PLL and SSCG.
- Generates the system clock by dividing the source clock by 1 to 32.
- Generates the CPU core clock (CCLK) by dividing the system clock by 1 to 8.
- Controls the clock supply to the peripheral modules.
- Controls the clocks according to the standby mode (HALT, or SLEEP).
- Controls a clock output to external devices.

To reduce current consumption, control the clock in conjunction with processing and use standby mode. For more information on reducing current consumption, see “Power Saving” in the appendix chapter.

Figure 6.1.1 shows the clock system and CMU module configuration.

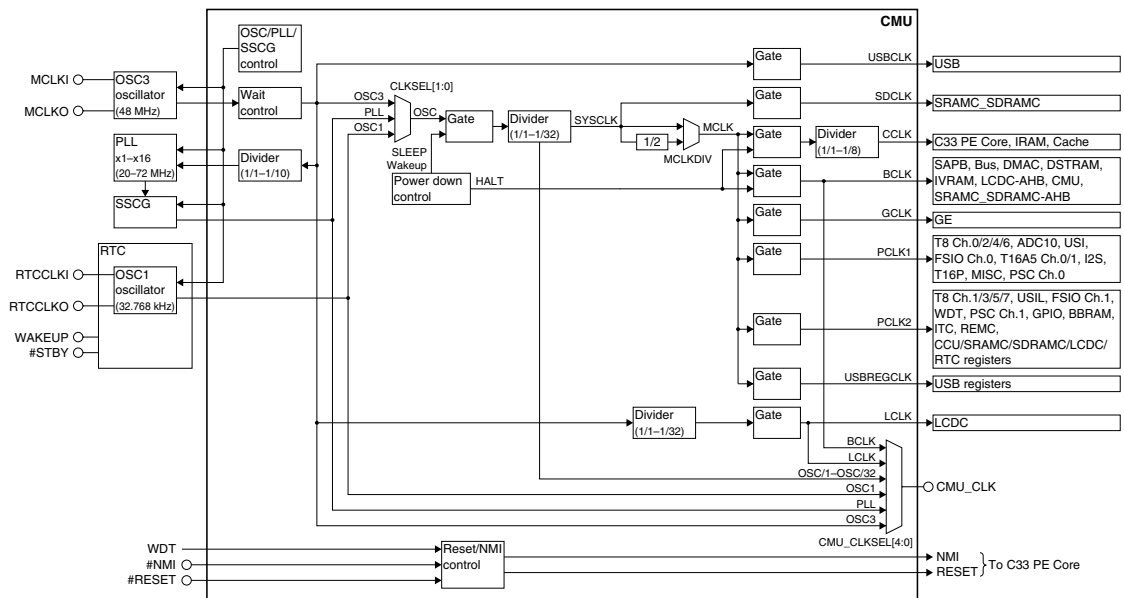


Figure 6.1.1 CMU Module Configuration

Note: The CMU control registers at addresses 0x300100–0x30010d are write-protected. Before the CMU control registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to CMUP[7:0]/CMU_PROTECT register. Note that since unnecessary rewrites to the CMU control registers could lead to erratic system operation, CMUP[7:0] should be set to other than 0x96 unless the CMU control registers must be rewritten.

6.2 CMU Pins

Table 6.2.1 lists the input/output pins for the CMU module.

Table 6.2.1 List of CMU Pins

Pin name	I/O	Qty	Function
RTCCLKI	I	1	OSC1 oscillator input pin Connect a crystal resonator (32.768 kHz), a feedback resistor, and a gate capacitor. Or input an external clock.
RTCCLKO	O	1	OSC1 oscillator output pin Connect a crystal resonator (32.768 kHz), a feedback resistor, a drain resistor, and a drain capacitor.
MCLKI	I	1	OSC3 oscillator input pin Connect a crystal or ceramic resonator (max. 48 MHz), a feedback resistor, and a gate capacitor. Or input an external clock.
MCLKO	O	1	OSC3 oscillator output pin Connect a crystal or ceramic resonator (max. 48 MHz), a feedback resistor, a drain resistor, and a drain capacitor.
CMU_CLK	O	1	CMU_CLK output pin Outputs the clock selected from OSC3, OSC1, PLL, OSC1–OSC/32, and LCLK.

The CMU output pin (CMU_CLK) is shared with an I/O port and are initially set as general purpose I/O port pin. The pin function must be switched using the port function select bit to use the general purpose I/O port pin as the CMU output pin. For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

6.3 Oscillators

6.3.1 OSC3 Oscillator Circuit

The OSC3 oscillator circuit generates the main clock for high-speed operation of the C33 PE Core and peripheral circuits.

Structure of the OSC3 oscillator circuit

The OSC3 oscillator circuit accommodates a crystal/ceramic oscillator and external clock input.

Figure 6.3.1.1 shows the structure of the OSC3 oscillator circuit.

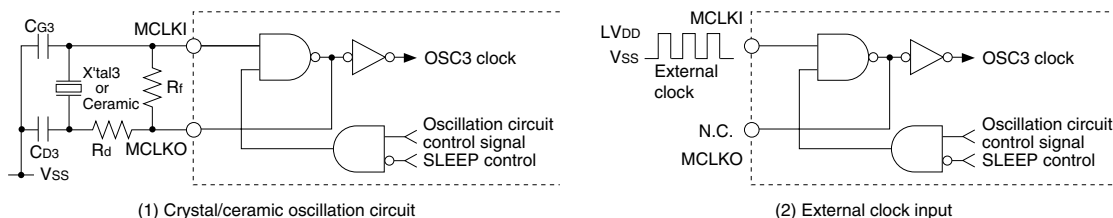


Figure 6.3.1.1 OSC3 Oscillator Circuit

For use as a crystal or ceramic oscillator circuit, connect a crystal (X'tal3) or ceramic resonator and a feedback resistor (Rf), two capacitors (CG3, CD3) and, if necessary, a drain resistor (Rd) to the MCLKI and MCLKO pins and Vss.

To use an external clock, leave the MCLKO pin open and input a LVDD-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: 5 MHz (min.) to 48 MHz (max.)
- 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.
- Before using a ceramic resonator, please be sure to contact Murata Manufacturing Co., Ltd. for further information on conditions of use for ceramic resonators.

For details of oscillation characteristics and external clock input characteristics, see “Electrical Characteristics.”

OSC3 oscillation on/off

The OSC3 oscillator circuit stops oscillating when OSC3EN/CMU_OSCCTL register is set to 0 and starts oscillating when set to 1. The OSC3 oscillator circuit stops oscillating even in SLEEP mode. After an initial reset, OSC3EN is set to 1 and the OSC3 oscillator circuit is activated.

Stabilization wait time at start of OSC3 oscillation

The OSC3 oscillator circuit incorporates an oscillation stabilization wait timer to prevent malfunctions due to unstable clock operations at the start of OSC3 oscillation—e.g., after an initial reset or waking from SLEEP mode when OSC3 or PLL is used as the system clock source. The OSC3 or PLL clock is not supplied to the system until the time set for this timer has elapsed. Use OSC3WT[3:0]/CMU_OSCCTL register to select one of 16 oscillation stabilization wait times.

Table 6.3.1.1 OSC3 Oscillation Stabilization Wait Time Settings

OSC3WT[3:0]	Oscillation stabilization wait time
0xf	128 cycles
0xe	256 cycles
0xd	512 cycles
0xc	1,024 cycles
0xb	2,048 cycles
0xa	4,096 cycles
0x9	8,192 cycles
0x8	16,384 cycles
0x7	32,768 cycles
0x6	65,536 cycles
0x5	131,072 cycles
0x4	262,144 cycles
0x3	524,288 cycles
0x2	1,048,576 cycles
0x1	2,097,152 cycles
0x0	4,194,304 cycles

(Default: 0xf)

This is set to 128 cycles (OSC3 clock) after an initial reset.

- Notes:**
- The OSC3 oscillation stabilization wait timer cannot be used when the OSC3 oscillator is turned on with software. Therefore, a software wait routine must be implemented.
 - Oscillation stability will vary, depending on the resonator and other external components. Carefully consider the OSC3 oscillation stabilization wait time before reducing the time. When waking from SLEEP mode if OSC3 or PLL is used as the system clock source, set the OSC3 oscillation stabilization wait time as follows:

$$\text{OSC3 oscillation stabilization wait time [cycle]} \geq \text{OSC3 oscillation start time [s (max.)]} \times \text{f}_{\text{SYSCLK}} [\text{Hz}]$$

f_{SYSCLK}: SYSCLK frequency when the clock source is OSC3 or PLL.

Example: When OSC3 oscillation start time (max.) = 10 ms and f_{SYSCLK} = 48 MHz

$$\text{OSC3 oscillation stabilization wait time} \geq 480,000 \text{ [cycles]}$$

OSC3WT[3:0] should be set to 0x3 (OSC3 oscillation stabilization wait time = 524,288 cycles).

6.3.2 OSC1 Oscillator Circuit

The S1C33L26 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.

Structure of the OSC1 oscillator circuit

The OSC1 oscillator circuit accommodates a crystal oscillator and external clock input. As for the RTC, RTCV_{DD} is used to supply power to this circuit.

6 CLOCK MANAGEMENT UNIT (CMU)

Figure 6.3.2.1 shows the structure of the OSC1 oscillator circuit.

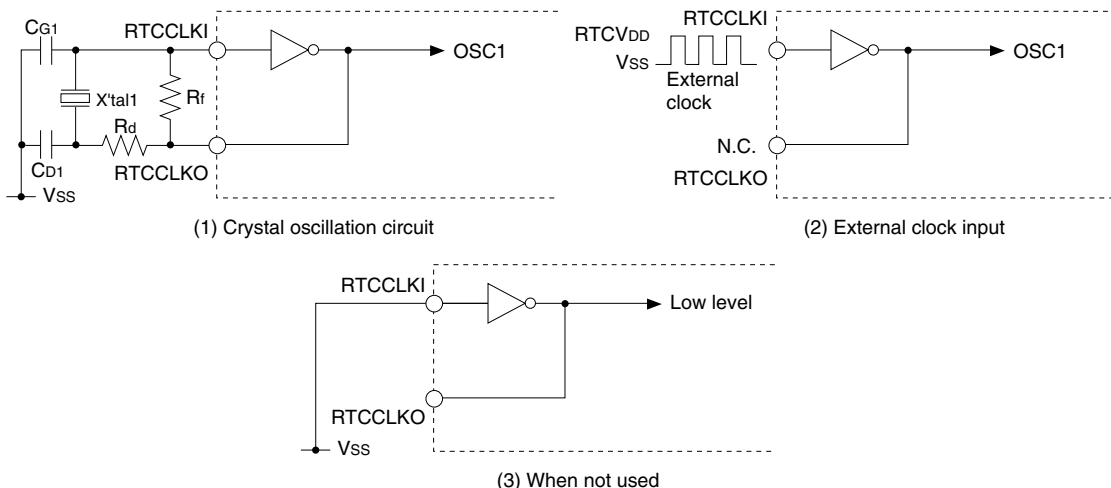


Figure 6.3.2.1 OSC1 Oscillator Circuit

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 RTCCLKI and RTCCLKO pins and Vss, as shown in the figure above.

To use an external clock, leave the RTCCLKO pin open and input an RTCVDD-level clock (whose duty cycle is 50%) to the RTCCLKI pin.

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 RTCCLKI pin to Vss and leave the RTCCLKO pin open.

OSC1 oscillation on/off

The OSC1 oscillator circuit stops oscillating when OSC1EN/CMU_OSCCTL register is set to 0 and starts oscillating when set to 1. After an initial reset, OSC1EN is set to 1 and the OSC1 oscillator circuit is activated. The OSC1 oscillator circuit does not stop oscillating in SLEEP mode.

Note: A finite time (see “Electrical Characteristics”) is required until oscillation stabilizes after the OSC1 oscillator starts oscillating. To prevent system malfunction, do not use the oscillator-derived clock until this oscillation stabilization time elapses.

6.4 PLL

The PLL multiplies the OSC3 clock frequency by a given value to generate a source clock for high-speed operation.

6.4.1 PLL On/Off Control

PLLPOWER/CMU_PLLCTL0 register can be used to turn the PLL on or off. Setting PLLPOWER to 1 initiates PLL operation. When initially reset, PLLPOWER is set to 0 (power-down mode), with the PLL turned off.

- Notes:**
- Immediately after the PLL is started by setting PLLPOWER to 1, an output clock stabilization wait time is required (see “Electric Characteristics”). After the PLL clock is stabilized, the clock source for the system can be switched over to the PLL.
 - Be sure to turn the PLL off before setting the CPU into SLEEP mode (before executing the slp instruction).

6.4.2 Selecting the PLL Input Clock

The PLL input clock can be selected from among 10 kinds of OSC3 divided clocks, OSC3/1 to OSC3/10, using PLLINDIV[3:0]/CMU_PLLINDIV register.

Table 6.4.2.1 PLL Input Clock (OSC3 Division Ratio) Selections

PLLINDIV[3:0]	Division ratio (OSC3/n)
0xf-0xa	1/8
0x9	1/10
0x8	1/9
0x7	1/8
0x6	1/7
0x5	1/6
0x4	1/5
0x3	1/4
0x2	1/3
0x1	1/2
0x0	1/1

(Default: 0x7)

- Notes:**
- The PLL input clock can only be selected when the PLL is turned off (PLLPOWR/CMU_PLLCTL0 register = 0) and the clock source is other than the PLL (CLKSEL[1:0]/CMU_OSCSEL register is not 0x2). 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.”

6.4.3 Setting the Frequency Multiplication Rate

The PLL frequency multiplication rate can be specified as shown in the table below using PLLN[3:0]/CMU_PLLCTL0 register.

Table 6.4.3.1 PLL Frequency Multiplication Rates

PLLN[3:0]	Multiplication rate
0xf	x16
0xe	x15
0xd	x14
0xc	x13
0xb	x12
0xa	x11
0x9	x10
0x8	x9
0x7	x8
0x6	x7
0x5	x6
0x4	x5
0x3	x4
0x2	x3
0x1	x2
0x0	x1

(Default: 0x0)

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 (PLLPOWR/CMU_PLLCTL0 register = 0) and the clock source is other than the PLL (CLKSEL[1:0]/CMU_OSCSEL register is not 0x2). If the frequency multiplication rate is changed while the system is operating with the PLL clock, the system may operate erratically.

6.4.4 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 using PLLV[1:0]/CMU_PLLCTL0 register. Lower value is better for low power consumption.

Table 6.4.4.1 W Value Settings

PLLV[1:0]	W
0x3	8
0x2	4
0x1	2
0x0	Setting prohibited

(Default: 0x1)

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) using PLLVC[3:0]/CMU_PLLCTL1 register.

Table 6.4.4.2 VC Value Settings

PLLVC[3:0]	f_{vco} [MHz]
0x8	$360 < f_{vco} \leq 400$
0x7	$320 < f_{vco} \leq 360$
0x6	$280 < f_{vco} \leq 320$
0x5	$240 < f_{vco} \leq 280$
0x4	$200 < f_{vco} \leq 240$
0x3	$160 < f_{vco} \leq 200$
0x2	$120 < f_{vco} \leq 160$
0x1	$100 \leq f_{vco} \leq 120$
Other	Setting prohibited

(Default: 0x1)

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]/CMU_PLLCTL1 register.

Table 6.4.4.3 RS Value Settings

PLLRS[3:0]	f_{REFCK} [MHz]
0xa	$5 \leq f_{REFCK} < 20$
0x8	$20 \leq f_{REFCK} \leq 150$
Other	Setting prohibited

(Default: 0x8)

LPF capacitance value (CS value)

Bits to set the LPF capacitance value (CS value) is provided in the CMU control registers, PLLCS[1:0]/CMU_PLLCTL2 register. However, do not alter the value of these bits, and leave them as initially set (0x0).

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]/CMU_PLLCTL2 register. However, do not alter the value of these bits, and leave them as initially set (0x10).

Table 6.4.4.4 PLL Setting Examples

OSC3 clock	PLL output clock	PLLINDIV[3:0]	PLLN[3:0]	PLLV[1:0]	PLLVC[3:0]	PLLRS[3:0]
6 MHz	72 MHz	1/1 (0x0)	x12 (0xb)	0x1	0x2	0xa
	60 MHz	1/1 (0x0)	x10 (0x9)	0x1	0x1	0xa
10 MHz	70 MHz	1/1 (0x0)	x7 (0x6)	0x1	0x2	0xa
	40 MHz	1/1 (0x0)	x4 (0x3)	0x2	0x2	0xa
20 MHz	60 MHz	1/1 (0x0)	x3 (0x2)	0x1	0x1	0x8
	40 MHz	1/1 (0x0)	x2 (0x1)	0x2	0x2	0x8
36 MHz	72 MHz	1/1 (0x0)	x2 (0x1)	0x1	0x2	0x8
48 MHz	72 MHz	1/8 (0x7)	x12 (0xb)	0x1	0x2	0xa
	60 MHz	1/8 (0x7)	x10 (0x9)	0x1	0x1	0xa

Note: The PLL can only be set up when the PLL is turned off (PLLPOWER/CMU_PLLCTL0 register = 0) and the clock source is other than the PLL (CLKSEL[1:0]/CMU_OSCSEL register is not 0x2). If settings are changed while the system is operating with the PLL clock, the system may operate erratically.

6.4.5 Power Supply for PLL

In order to prevent undesirable effects of noise, the PLLVDD and PLLVSS pins are provided, in addition to the core power supply, to feed power to the PLL. Make sure that the following voltages are supplied to the respective pins.

PLLVDD pin: Supply LVDD level voltage.

PLLVSS pin: Set to VSS level.

For pin assignments, see the “Pin Descriptions” section.

6.5 SSCG

The SSCG (Spread Spectrum Clock Generator) is a circuit used to reduce EMI (Electromagnetic Interference) noise by conducting spread spectrum (or SS modulation) on the PLL output clock signal. SSCG conducts SS modulation on the PLL output clock signal. Thus SS modulation contributes in reducing noise when the PLL output clock is selected as the system clock source since, in this case, the SS modulation is effective for all the operating clocks for the core and peripheral circuits (except the RTC that uses the OSC1 clock).

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).

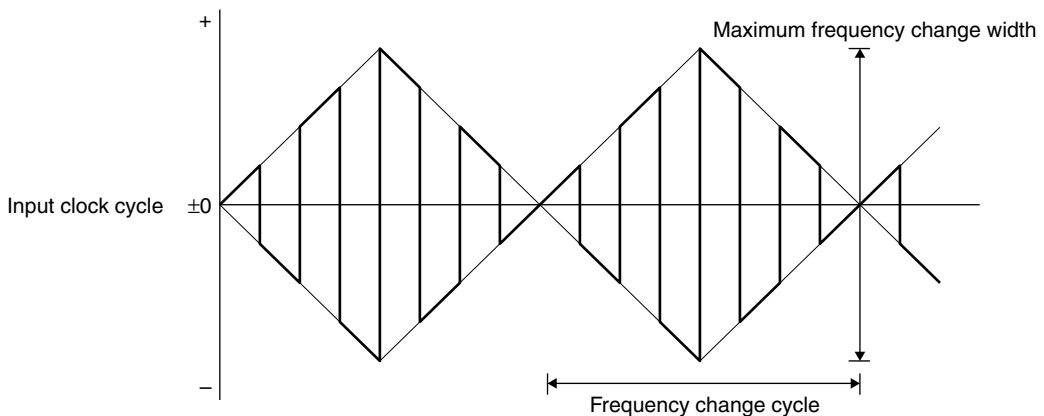


Figure 6.5.1 SS Modulation

6.5.1 SSCG On/Off Control

The SSCG can be turned on or off using SSMCON/CMU_SSCG0 register. Setting SSMCON to 1 causes the SSCG to start operating. When initially reset, SSMCON 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 for the PLL stabilization time to elapse at the minimum.
3. Turn the SSCG on.

To turn the SSCG off

1. Turn the SSCG off.
2. Turn the PLL off.

- SS modulation is conducted on the PLL output clock signal. SS modulation is not applicable to the signals other than the PLL clock signal. When the PLL output clock is not used for the system clock, turn the SSCG off.

6.5.2 SS Modulation Parameter Settings

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 the table below using SSMCIDT[3:0]/CMU_SSCG1 register. The maximum frequency change width will be about ±2% of the PLL output clock by the above setting.

Table 6.5.2.1 Maximum Frequency Change Width Settings

PLL output clock frequency f [MHz]	SSMCIDT[3:0]
$f \leq 19.8$	0xf
$19.8 < f \leq 21.2$	0xe
$21.2 < f \leq 22.5$	0xd
$22.5 < f \leq 24.2$	0xc
$24.2 < f \leq 25.9$	0xb
$25.9 < f \leq 28.4$	0xa
$28.4 < f \leq 30.8$	0x9
$30.8 < f \leq 34.2$	0x8
$34.2 < f \leq 37.8$	0x7
$37.8 < f \leq 43.1$	0x6
$43.1 < f \leq 48.9$	0x5
$48.9 < f \leq 58.5$	0x4
$58.5 < f \leq 69.7$	0x3
$69.7 < f \leq 90.0$	0x2
–	0x1
–	0x0

(Default: undefined)

SSMCITM[3:0]/CMU_SSCG1 register is used to set the frequency change cycle. However, always set it to 0x1.

- Notes:** • SSMCIDT[3:0] must be set according to the PLL output clock frequency as shown in Table 6.5.2.1. 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] and SSMCITM[3:0] 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.

6.6 System Clock Settings

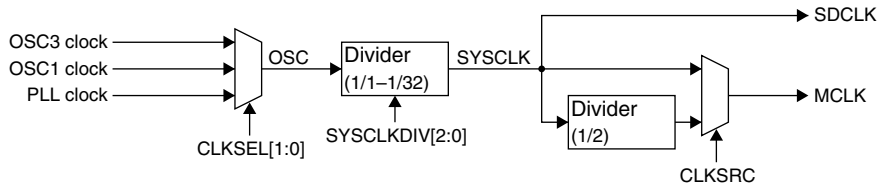


Figure 6.6.1 System Clock Control Circuit

6.6.1 System Clock Source Selection

The system clock source can be selected from OSC3, OSC1, or PLL using CLKSEL[1:0]/CMU_OSCSEL register.

Table 6.6.1.1 System Clock Source Selections

CLKSEL[1:0]	Clock source
0x3	Reserved
0x2	PLL
0x1	OSC1
0x0	OSC3

(Default: 0x0)

The following shows system clock switching procedures:

Switching the system clock to OSC1 from OSC3

1. Turn the OSC1 oscillator on if it is off. (OSC1EN = 1)
2. Wait until the OSC1 oscillation is stabilized.
3. Stop the peripheral circuits being currently operated.
4. Select the OSC1 clock as the system clock. (CLKSEL[1:0] = 0x1)
5. Check if CLKSEL[1:0] is set to 0x1 to confirm that the system clock has been switched to OSC1.
6. Turn the OSC3 oscillator off to reduce current consumption if the CMU_CLK output circuit has not used the OSC3 clock. (OSC3EN = 0)

Switching the system clock to PLL from OSC3

1. Configure the PLL input clock and the PLL parameters such as the multiplication rate before activating the PLL.
2. Enable the PLL. (PLLPOWR = 1)
3. Wait until the PLL operation is stabilized.
4. Stop the peripheral circuits being currently operated except the RTC.
5. Select the PLL clock as the system clock. (CLKSEL[1:0] = 0x2)
6. Check if CLKSEL[1:0] is set to 0x2 to confirm that the system clock has been switched to PLL.

Switching the system clock to OSC3 from OSC1

1. Turn the OSC3 oscillator on if it is off. (OSC3EN = 1)
2. Wait until the OSC3 oscillation is stabilized.
3. Stop the peripheral circuits being currently operated except the RTC.
4. Select the OSC3 clock as the system clock. (CLKSEL[1:0] = 0x0)
5. Check if CLKSEL[1:0] is set to 0x0 to confirm that the system clock has been switched to OSC3.

Switching the system clock to PLL from OSC1

1. Switch the system clock to OSC3 from OSC1 by following the procedure shown above.
2. Switch the system clock to PLL from OSC3 by following the procedure shown above.

Switching the system clock to OSC3 from PLL

1. Stop the peripheral circuits being currently operated except the RTC.
2. Select the OSC3 clock as the system clock. (CLKSEL[1:0] = 0x0)
3. Check if CLKSEL[1:0] is set to 0x0 to confirm that the system clock has been switched to OSC3.
4. Disable the PLL to reduce current consumption if the CMU_CLK output circuit has not used the PLL clock. (PLLPOWR = 0)

Switching the system clock to OSC1 from PLL

1. Switch the system clock to OSC3 from PLL by following the procedure shown above.
2. Switch the system clock to OSC1 from OSC3 by following the procedure shown above.

Note: Do not select the system clock from deactivated clock sources. It will cause the system to hang as the CMU does not include a protection mechanism against such system clock selection.

6.6.2 System Clock Frequency Setting

The source clock frequency can be divided by 1 to 32 to generate the system clock using SYSCLKDIV[2:0]/CMU_SYSCLKDIV register. Setting the system clock to the lowest frequency possible according to the processing can reduce current consumption.

Table 6.6.2.1 System Clock Division Ratio

SYSCLKDIV[2:0]	Division ratio (OSC/n)
0x7-0x6	1/1
0x5	1/32
0x4	1/16
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

6.6.3 Main System Clock (MCLK) Setting

The MCLK clock is the main system clock for the C33 PE Core and internal modules. It is used as CCLK, BCLK, GCLK, PCLK1, and PCLK2.

Either SYSCLK (configured with CLKSEL[1:0] and SYSCLKDIV[2:0]) or SYSCLK/2 can be selected as MCLK using MCLKDIV/CMU_SYSCLKDIV register.

Table 6.6.3.1 MCLK (SYSCLK Division Ratio) Selections

MCLKDIV	MCLK (SYSCLK/n)
1	1/2
0	1/1

(Default: 0x0)

When using the SDRAMC in double frequency mode (MCLK : SDCLK = 1 : 2), MCLK should be set to SYSCLK/2 (SYSCLK 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.

6.7 Clock Supply Control

To reduce current consumption on the chip, the CMU provides some gate circuits to disable clock supply.

6.7.1 Core Clock (CCLK)

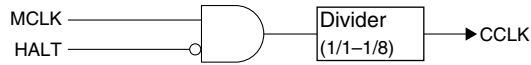


Figure 6.7.1.1 CCLK Control Circuit

The CCLK clock is the C33 PE Core operating clock.

In normal mode, CCLK is always supplied to the C33 PE Core.

When the C33 PE Core executes the halt or slp instruction, the CMU stops supplying the clock to the C33 PE Core and the C33 PE Core enters a standby (HALT or SLEEP) mode. The CMU resumes the clock supply to the C33 PE Core when the standby mode is canceled by occurrence of an interrupt.

The CMU module includes a divider to slow down CCLK. To reduce current consumption, operate the C33 PE Core with the slowest possible clock speed.

CLK_DOWN[1:0]/CCU_CCLKDV register is used to select the clock division ratio. For more information on CLK_DOWN[1:0], see the “Cache Controller (CCU)” chapter.

Table 6.7.1.1 Core Clock Division Ratio Selection

CLK_DOWN[1:0]	Division ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

6.7.2 Bus Clock (BCLK)

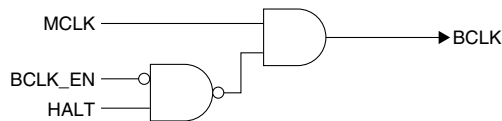


Figure 6.7.2.1 BCLK Control Circuit

The BCLK clock is used to operate the modules listed below.

- IVRAM (Area 3)
- DSTRAM (Area 3)
- SRAM controller (SRAMC)
- SDRAM controller (SDRAMC)
- DMA controller (DMAC)
- LCD controller (LCDC) bus interface
- Clock management unit (CMU) registers
- Bus arbiters

BCLK is required for bus and memory operations, therefore, it is always supplied to the modules listed above in normal mode.

However, the BCLK supply in HALT mode can be disabled using BCLK_EN/CMU_CLKCTL register if the LCDC and DMA do not need bus operations.

To stop BCLK in HALT mode, set BCLK_EN to 0. The CMU stops supplying BCLK when the halt instruction is executed. The CMU resumes the clock supply when the HALT mode is canceled.

To supply BCLK in HALT mode, set BCLK_EN to 1 (default). The modules listed above operates even in HALT mode.

BCLK stops in SLEEP mode (when the slp instruction is executed) regardless of the BCLK_EN set value.

6.7.3 Peripheral Module Clocks (PCLK1, PCLK2)

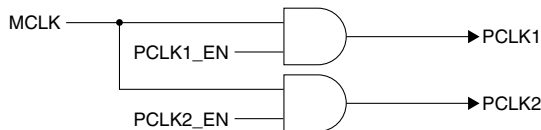


Figure 6.7.3.1 PCLK Control Circuit

The PCLK1 and PCLK2 clocks are used to operate the modules listed below.

Table 6.7.3.1 Peripheral Modules and Operating Clocks

Clock	Clock enable bit	Peripheral modules
PCLK1	PCLK1_EN/CMU_CLKCTL register	<ul style="list-style-type: none"> • Prescaler (PSC Ch.0) • 8-bit programmable timer Ch.0, 2, 4, 6 (T8 Ch.0, 2, 4, 6) • 16-bit PWM timer Ch.0, 1 (T16A5 Ch.0, 1) • 16-bit audio PWM timer (T16P) • Universal serial interface (USI) • Serial interface Ch.0 (FSIO Ch.0) • A/D converter (ADC10) • I²S (I2S) • Misc registers (MISC)
PCLK2	PCLK2_EN/CMU_CLKCTL register	<ul style="list-style-type: none"> • Prescaler (PSC Ch.1) • 8-bit programmable timer Ch.1, 3, 5, 7 (T8 Ch.1, 3, 5, 7) • Universal serial interface with LCD interface (USIL) • Serial interface Ch.1 (FSIO Ch.1) • Watchdog timer (WDT) • Remote controller (REMC) • Interrupt controller (ITC) • I/O ports (GPIO) • BBRAM • Cache controller (CCU) registers • Real-time clock (RTC) registers • SRAM controller (SRAMC) registers • SDRAM controller (SDRAMC) registers • LCD controller (LCDC) registers

The peripheral module clock (PCLK1, PCLK2) supply can be controlled using the clock enable bit (PCLK1_EN, PCLK2_EN).

The default setting of the clock enable bit is 1, which enables the clock supply. Disable the clock supply by setting the clock enable bit to 0 to reduce current consumption unless all the modules that use the clock need to be running. The clock is supplied even in HALT mode when the clock enable bit is set to 1. To stop the modules in HALT mode, set the clock enable bit to 0 before executing the halt instruction.

In SLEEP mode (when the slp instruction is executed), these clocks stop even if the clock enable bit is set to 1.

6.7.4 GE Module Clock (GCLK)

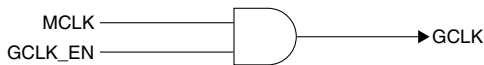


Figure 6.7.4.1 GCLK Control Circuit

The GCLK clock is the GE module operating clock. GCLK_EN/CMU_CLKCTL register is used for clock supply control. The default setting of GCLK_EN is 1, which enables the clock supply. Disable the clock supply by setting GCLK_EN to 0 to reduce current consumption when the GE functions are not used.

GCLK is supplied even in HALT mode when GCLK_EN is set to 1. To stop the GE module in HALT mode, set GCLK_EN to 0 before executing the halt instruction.

In SLEEP mode (when the slp instruction is executed), GCLK stops even if GCLK_EN is set to 1.

6.7.5 LCDC Module Clock (LCLK)

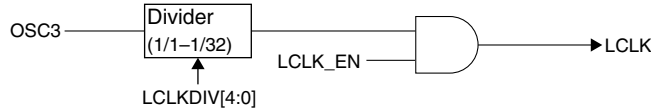


Figure 6.7.5.1 LCLK Control Circuit

The LCLK clock is generated by dividing the OSC3 clock and is supplied the LCD controller (LCDC). The frequency divider generates 32 kinds of clocks from OSC3/1 to OSC3/32. Select a division ratio according to the frame rate using LCLKDIV[4:0]/CMU_LCLKDIV register.

$$\text{Frame rate} = \frac{f_{\text{LCLK}}}{\text{HT} \times \text{VT}} \text{ [Hz]}$$

f_{LCLK} : LCLK frequency

HT: Horizontal total period (horizontal panel size + horizontal non-display period) [pixels]

VT: Vertical total period (vertical panel size + vertical non-display period) [lines]

Table 6.7.5.1 LCDC Clock (OSC3 Division Ratio) Selections

LCLKDIV[4:0]	Division ratio (OSC3/n)	LCLKDIV[4:0]	Division ratio (OSC3/n)
0x1f	1/32	0xf	1/16
0x1e	1/31	0xe	1/15
0x1d	1/30	0xd	1/14
0x1c	1/29	0xc	1/13
0x1b	1/28	0xb	1/12
0x1a	1/27	0xa	1/11
0x19	1/26	0x9	1/10
0x18	1/25	0x8	1/9
0x17	1/24	0x7	1/8
0x16	1/23	0x6	1/7
0x15	1/22	0x5	1/6
0x14	1/21	0x4	1/5
0x13	1/20	0x3	1/4
0x12	1/19	0x2	1/3
0x11	1/18	0x1	1/2
0x10	1/17	0x0	1/1

(Default: 0x7)

LCLK_EN/CMU_CLKCTL register is used for clock supply control (default: off). Before using the LCDC, set LCLK_EN to 1. Note that PCLK2 is required to set the LCDC registers.

In HALT mode, LCLK does not stop if LCLK_EN is set to 1. To stop supplying the clock in HALT mode, LCLK_EN should be set to 0 before executing the halt instruction.

In SLEEP mode (when the slp instruction is executed), LCLK stops even if LCLK_EN is set to 1.

Note: Disable LCLK supply (LCLK_EN = 0) when changing the clock division ratio using LCLKDIV[4:0] or before executing the slp instruction.

6.7.6 SRAMC and SDRAMC Clock (SDCLK)

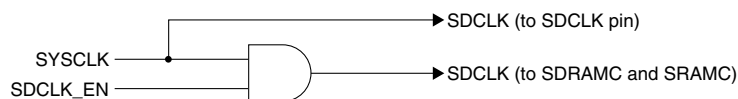


Figure 6.7.6.1 SDCLK Control Circuit

The SDCLK clock is used for the SRAMC and SDRAMC. SYSCLK is supplied as SDCLK. SDCLK_EN/CMU_CLKCTL register is used for clock supply control. The default setting of SDCLK_EN is 1, which enables the clock supply. Disable the clock supply by setting SDCLK_EN to 0 to reduce current consumption when the external bus devices (e.g., SRAM and SDRAM) are not used. When using the SDRAMC in double frequency mode (MCLK : SDCLK = 1 : 2), the SDRAMC operates with SDCLK (Max. 72 MHz) configured to double the frequency of MCLK (Max. 36 MHz), while the SRAMC operates on the same frequency as MCLK.

6 CLOCK MANAGEMENT UNIT (CMU)

SDCLK is supplied even in HALT mode when SDCLK_EN is set to 1. To stop the clock supply in HALT mode, set SDCLK_EN to 0 before executing the halt instruction.

In SLEEP mode (when the slp instruction is executed), SDCLK stops even if SDCLK_EN is set to 1.

SYSCLK is directly fed to the SDCLK pin used to connecting an external SDRAM. In normal or HALT mode, the SDCLK pin always outputs the clock. In SLEEP mode, the SDCLK pin stops supplying the clock.

6.7.7 USB Clocks (USBCLK, USBREGCLK)

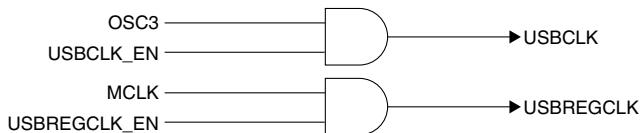


Figure 6.7.7.1 USBCLK/USBREGCLK Control Circuit

The USBCLK clock is the USB operating clock supplied to the USB function controller. Use a 48 MHz ceramic resonator for the OSC3 oscillator circuit when using the USB function. USBCLK_EN/CMU_CLKCTL register is used for clock supply control. The default setting of USBCLK_EN is 0, which disables the clock supply. Enable the clock supply by setting USBCLK_EN to 1 before the USB function controller can be used.

USBCLK is supplied even in HALT mode when USBCLK_EN is set to 1. To stop the clock supply in HALT mode, set USBCLK_EN to 0 before executing the halt instruction.

In SLEEP mode (when the slp instruction is executed), USBCLK stops even if USBCLK_EN is set to 1.

The USBREGCLK clock is the clock for accessing the USB registers. USBREGCLK_EN/CMU_CLKCTL register is used for clock supply control. The default setting of USBREGCLK_EN is 0, which disables the clock supply. Enable the clock supply by setting USBREGCLK_EN to 1 before the USB registers can be accessed.

USBREGCLK is supplied even in HALT mode when USBREGCLK_EN is set to 1. To stop the clock supply in HALT mode, set USBREGCLK_EN to 0 before executing the halt instruction.

In SLEEP mode (when the slp instruction is executed), USBREGCLK stops even if USBREGCLK_EN is set to 1.

Note: When accessing the USB registers, the USBCLK clock must be supplied to the USB function controller in addition to USBREGCLK.

6.8 Clock External Output (CMU_CLK)

An internally generated clock can be output from the CMU_CLK pin to external devices.

The output clock can be selected from among 11 clocks using CMU_CLKSEL[4:0]/CMU_CMUCLK register.

Table 6.8.1 CMU_CLK Selections

CMU_CLKSEL[4:0]	CMU_CLK
0xf-0xb	Reserved
0xa	OSC/32
0x9	OSC/16
0x8	OSC/8
0x7	OSC/4
0x6	OSC/2
0x5	OSC/1
0x4	LCLK
0x3	BCLK
0x2	PLL
0x1	OSC1
0x0	OSC3

(Default: 0x0)

CMU_CLK can be selected at any time. However, switching over the clocks creates hazards.

Note: Settings other than those listed in Table 6.8.1 are reserved for testing. Do not set undescribed values to CMU_CLKSEL[4:0] as undesired clocks may output.

6.9 Standby Modes

The S1C33L26 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.

6.9.1 HALT Mode

The CPU suspends program execution upon executing the halt instruction and enters HALT mode. 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.

In HALT mode, CCLK stops so the CPU, CCU, and IRAM stop operating. Furthermore, BCLK (bus-related modules) can be stopped in HALT mode (after the halt instruction is executed) by setting BCLK_EN/CMU_CLKCTL register to 0 (see Sections 6.7.2 for BCLK). 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. When an interrupt is used to cancel HALT mode, the C33 PE Core uses the interrupt signal sent from the interrupt controller (ITC). Therefore, the interrupts used to cancel HALT mode must be enabled in the interrupt source modules. The C33 PE Core can restart from HALT mode even if the PSR is set to disable interrupts.

When the IE (interrupt enable) bit in the PSR is set to 1 (enabled), the C33 PE Core executes the interrupt handler routine after HALT mode is canceled. When the IE bit is set to 0 (disabled), an interrupt does not occur and the C33 PE Core resumes execution from the instruction that follows the halt instruction.

The #NMI signal releases the CPU from HALT mode when it goes low level.

Note: The PCLK2 clock must be enable before setting the S1C33L26 into HALT mode, as it is required for ITC module when the C33 PE Core wakes up from HALT mode.

6.9.2 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 clocks. Therefore, all peripheral modules (except for the OSC1 oscillator circuit and RTC) stop operating.

The CPU is reawaken from SLEEP mode by initial reset, an RTC interrupt, an NMI, or other interrupt from an external device (port input interrupt).

The C33 PE Core can restart from SLEEP mode even if the PSR is set to disable interrupts. When the IE (interrupt enable) bit in the PSR is set to 0 (disabled), an interrupt does not occur and the C33 PE Core resumes execution from the instruction that follows the slp instruction. When the IE bit is set to 1 (enabled), the C33 PE Core executes the interrupt handler routine after SLEEP mode is canceled.

The #NMI signal releases the CPU from SLEEP mode when it goes low level.

Notes:

- In SLEEP mode, there is a time lag between inputting an interrupt signal for wake-up and starting the clock supply to the interrupt source module, so a delay will occur until the interrupt flag is set. Therefore, no interrupt will occur if the interrupt signal is negated before the clock is supplied, as the interrupt flag is not set.

Furthermore, additional time is needed for the C33 PE Core to accept the interrupt request from the ITC, the C33 PE Core may execute a few instructions that follow the slp instruction before it executes the interrupt handler routine.

When a level trigger port input interrupt is used to wake up the C33 PE Core from SLEEP mode, assert the input signal until the clock supply has started. Edge trigger port input interrupts can also used to cancel SLEEP mode. The active signal edge will be automatically converted into an active level signal by the GPIO module and it keeps on active until the clock supply has started.

The same problem may occur when the CPU wakes up from SLEEP mode by an NMI. No interrupt will occur if the #NMI signal is negated before the clock is supplied.

- Before setting the S1C33L26 into SLEEP mode, the clock supply for the USB and LCDG must be disabled.
- Be sure to disable the instruction and data caches before executing the halt or slp instruction.

6.10 Control Register Details

Table 6.10.1 List of CMU Registers

Address	Register name		Function
0x300100	CMU_OSCSEL	Clock Source Select Register	Select system clock source
0x300101	CMU_OSCCTL	Oscillation Control Register	Control oscillators
0x300103	CMU_LCLKDIV	LCDC Clock Division Ratio Select Register	Set LCLK frequency
0x300104	CMU_CLKCTL	Clock Control Register	Control clock supply to peripheral/bus modules
0x300105	CMU_SYCLKDIV	System Clock Division Ratio Select Register	Set system clock frequency
0x300106	CMU_CMUCLK	CMU_CLK Select Register	Select CMU_CLK output clock
0x300107	CMU_PLLINDIV	PLL Input Clock Division Ratio Select Register	Set PLL input clock frequency
0x300108	CMU_PLLCTL0	PLL Control Register 0	Set PLL multiplication rate and enable PLL
0x300109	CMU_PLLCTL1	PLL Control Register 1	Set PLL parameters
0x30010a	CMU_PLLCTL2	PLL Control Register 2	
0x30010c	CMU_SSCG0	SSCG Macro Control Register 0	Enable SSCG
0x30010d	CMU_SSCG1	SSCG Macro Control Register 1	Set SSCG parameters
0x300110	CMU_PROTECT	CMU Write Protect Register	Enable/disable CMU register write protection

The CMU module registers are described in detail below. These are 8-bit registers.

- Notes:**
- When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.
 - The CMU control registers at addresses 0x300100–0x30010d are write-protected. Before the CMU control registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to CMUP[7:0]/CMU_PROTECT register. Note that since unnecessary re-writes to the CMU control registers could lead to erratic system operation, CMUP[7:0] should be set to other than 0x96 unless the CMU control registers must be rewritten.

Clock Source Select Register (CMU_OSCSEL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Clock Source Select Register (CMU_OSCSEL)	0x300100 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.	
		D1–0	CLKSEL[1:0]	System clock source select	CLKSEL[1:0] Clock source	0x0	R/W	Write-protected	
					0x3	Not allowed			
					0x2	PLL			
					0x1	OSC1			
				0x0	OSC3				

D[7:2] Reserved

D[1:0] CLKSEL[1:0]: System Clock Source Select Bits

Selects the system clock source.

Table 6.10.2 System Clock Source Selections

CLKSEL[1:0]	Clock source
0x3	Reserved
0x2	PLL
0x1	OSC1
0x0	OSC3

(Default: 0x0)

Note: Do not select the system clock from deactivated clock sources. It will cause the system to hang as the CMU does not include a protection mechanism against such system clock selection.

Oscillation Control Register (CMU_OSCCTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Oscillation Control Register (CMU_OSCCTL)	0x300101 (8 bits)	D7-4	OSC3WT[3:0]	OSC3 wait cycle select	OSC3WT[3:0]	Wait cycle	0xf	R/W	Write-protected
					0xf	128 cycles			
					0xe	256 cycles			
					0xd	512 cycles			
					0xc	1,024 cycles			
					0xb	2,048 cycles			
					0xa	4,096 cycles			
					0x9	8,192 cycles			
					0x8	16,384 cycles			
					0x7	32,768 cycles			
0x6	65,536 cycles								
0x5	131,072 cycles								
0x4	262,144 cycles								
0x3	524,288 cycles								
0x2	1,048,576 cycles								
0x1	2,097,152 cycles								
0x0	4,194,304 cycles								
D3-2	–	reserved	–	–	–	–	0 when being read.		
D1	OSC1EN	OSC1 enable	1	Enable	0	Disable	1	R/W	Write-protected
D0	OSC3EN	OSC3 enable	1	Enable	0	Disable	1	R/W	

D[7:4] OSC3WT[3:0]: OSC3 Wait Cycle Select Bits

An oscillation stabilization wait timer is set to prevent malfunctions due to unstable clock operation at the start of OSC3 oscillation. The OSC3 or PLL clock is not supplied to the system immediately after OSC3 oscillation starts—e.g., after an initial reset or when waking from SLEEP—until the time set here has elapsed.

Table 6.10.3 OSC3 Oscillation Stabilization Wait Time Settings

OSC3WT[3:0]	Oscillation stabilization wait time
0xf	128 cycles
0xe	256 cycles
0xd	512 cycles
0xc	1,024 cycles
0xb	2,048 cycles
0xa	4,096 cycles
0x9	8,192 cycles
0x8	16,384 cycles
0x7	32,768 cycles
0x6	65,536 cycles
0x5	131,072 cycles
0x4	262,144 cycles
0x3	524,288 cycles
0x2	1,048,576 cycles
0x1	2,097,152 cycles
0x0	4,194,304 cycles

(Default: 0xf)

This is set to 128 cycles (OSC3 clock) after an initial reset.

- Notes:**
- The OSC3 oscillation stabilization wait timer cannot be used when the OSC3 oscillator is turned on with software. Therefore, a software wait routine must be implemented.
 - Oscillation stability will vary, depending on the resonator and other external components. Carefully consider the OSC3 oscillation stabilization wait time before reducing the time. When waking from SLEEP mode if OSC3 or PLL is used as the system clock source, set the OSC3 oscillation stabilization wait time as follows:

$$\text{OSC3 oscillation stabilization wait time [cycle]} \geq \text{OSC3 oscillation start time [s] (max.)} \times \text{fsysCLK [Hz]}$$

fsysCLK: SYSCLK frequency when the clock source is OSC3 or PLL.

Example: When OSC3 oscillation start time (max.) = 10 ms and fsysCLK = 48 MHz

$$\text{OSC3 oscillation stabilization wait time} \geq 480,000 \text{ [cycles]}$$

OSC3WT[3:0] should be set to 0x3 (OSC3 oscillation stabilization wait time = 524,288 cycles).

6 CLOCK MANAGEMENT UNIT (CMU)

D[3:2] Reserved

D1 OSC1EN: OSC1 Enable Bit

Enables or disables OSC1 oscillator operations.

1 (R/W): Enabled (on) (default)

0 (R/W): Disabled (off)

Note: Do not stop the OSC1 oscillator if the OSC1 clock is being used as the system clock.

D0 OSC3EN: OSC3 Enable Bit

Enables or disables OSC3 oscillator operations.

1 (R/W): Enabled (on) (default)

0 (R/W): Disabled (off)

Note: The OSC3 oscillator cannot be stopped if the OSC3 or PLL clock is being used as the system clock.

LCDC Clock Division Ratio Select Register (CMU_LCLKDIV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
LCDC Clock Division Ratio Select Register (CMU_LCLKDIV)	0x300103 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.		
		D4-0	LCLKDIV[4:0]	LCDC clock division ratio select	LCLKDIV[4:0] Division ratio	0x7	R/W	Clock source = OSC3 Write-protected		
							0x1f		1/32	
							0x1e		1/31	
							0x1d		1/30	
							0x1c		1/29	
							0x1b		1/28	
							0x1a		1/27	
							0x19		1/26	
							0x18		1/25	
							0x17		1/24	
							0x16		1/23	
							0x15		1/22	
							0x14		1/21	
							0x13		1/20	
							0x12		1/19	
							0x11		1/18	
							0x10		1/17	
							0xf		1/16	
							0xe		1/15	
							0xd		1/14	
							0xc		1/13	
							0xb		1/12	
							0xa		1/11	
							0x9		1/10	
							0x8		1/9	
							0x7		1/8	
							0x6		1/7	
							0x5		1/6	
							0x4		1/5	
							0x3		1/4	
							0x2		1/3	
					0x1		1/2			
					0x0		1/1			

D[7:5] Reserved

D[4:0] LCLKDIV[4:0]: LCDC Clock Division Ratio Select Bits

Selects the LCDC clock (LCLK) from among 32 kinds of OSC3 division clocks. Select a clock according to the frame rate.

$$\text{Frame rate} = \frac{f_{\text{LCLK}}}{\text{HT} \times \text{VT}} \text{ [Hz]}$$

f_{LCLK} : LCLK frequency

HT: Horizontal total period (horizontal panel size + horizontal non-display period) [pixels]

VT: Vertical total period (vertical panel size + vertical non-display period) [lines]

Table 6.10.4 LCDC Clock (OSC3 Division Ratio) Selections

LCLKDIV[4:0]	Division ratio (OSC3/n)	LCLKDIV[4:0]	Division ratio (OSC3/n)
0x1f	1/32	0xf	1/16
0x1e	1/31	0xe	1/15
0x1d	1/30	0xd	1/14
0x1c	1/29	0xc	1/13
0x1b	1/28	0xb	1/12
0x1a	1/27	0xa	1/11
0x19	1/26	0x9	1/10
0x18	1/25	0x8	1/9
0x17	1/24	0x7	1/8
0x16	1/23	0x6	1/7
0x15	1/22	0x5	1/6
0x14	1/21	0x4	1/5
0x13	1/20	0x3	1/4
0x12	1/19	0x2	1/3
0x11	1/18	0x1	1/2
0x10	1/17	0x0	1/1

(Default: 0x7)

Clock Control Register (CMU_CLKCTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
Clock Control Register (CMU_CLKCTL)	0x300104 (8 bits)	D7	USBREGCLK_EN	USB I/O register clock enable	1	Enable	0	Disable	0	R/W	Write-protected
		D6	LCLK_EN	LCDC clock enable	1	Enable	0	Disable	0	R/W	
		D5	USBCLK_EN	USB clock enable	1	Enable	0	Disable	0	R/W	
		D4	SDCLK_EN	SDCLK clock enable	1	Enable	0	Disable	1	R/W	
		D3	BCLK_EN	BCLK clock enable (in HALT)	1	Enable	0	Disable	1	R/W	
		D2	PCLK2_EN	PCLK2 clock enable	1	Enable	0	Disable	1	R/W	
		D1	PCLK1_EN	PCLK1 clock enable	1	Enable	0	Disable	1	R/W	
		D0	GCLK_EN	GCLK clock enable	1	Enable	0	Disable	1	R/W	

D7 USBREGCLK_EN: USB I/O Register Clock Enable Bit

Enables or disables the USBREGCLK clock supply to the USB function controller.

1 (R/W): Enabled (on)

0 (R/W): Disabled (off) (default)

The USBREGCLK_EN default setting is 0, which stops the clock supply. Setting USBREGCLK_EN to 1 supplies USBREGCLK to the USB function controller. If no USB register access is required, stop the clock supply to reduce current consumption.

D6 LCLK_EN: LCLK Clock Enable Bit

Enables or disables the LCLK clock supply to the LCD controller.

1 (R/W): Enabled (on)

0 (R/W): Disabled (off) (default)

The LCLK_EN default setting is 0, which stops the clock supply. Setting LCLK_EN to 1 supplies LCLK to the LCD controller. If no LCD display is required, stop the clock supply to reduce current consumption.

D5 USBCLK_EN: USB Clock Enable Bit

Enables or disables the USBCLK clock supply to the USB function controller.

1 (R/W): Enabled (on)

0 (R/W): Disabled (off) (default)

The USBCLK_EN default setting is 0, which stops the clock supply. Setting USBCLK_EN to 1 supplies USBCLK to the USB function controller. If no USB operation is required, stop the clock supply to reduce current consumption.

D4 SDCLK_EN: SDCLK Clock Enable Bit

Enables or disables the SDCLK clock supply to the SRAMC and SDRAMC.

1 (R/W): Enabled (on) (default)

0 (R/W): Disabled (off)

The SDCLK_EN default setting is 1, which enables the clock supply. If the SRAMC and SDRAMC modules can be stopped, disable the clock supply by setting SDCLK_EN to 0 to reduce current consumption.

Note: When using the SDRAMC in double frequency mode (MCLK : SDCLK = 1 : 2), the SDRAMC operates with SDCLK configured to double the frequency of MCLK, while the SRAMC operates on the same frequency as MCLK.

D3 BCLK_EN: BCLK Clock Enable (in HALT) Bit

Enables or disables the BCLK clock supply in HALT mode.

1 (R/W): Enabled (on) (default)

0 (R/W): Disabled (off)

The BCLK clock is used to operate the modules listed below.

- IVRAM (Area 3)
- DSTRAM (Area 3)
- SRAM controller (SRAMC)
- SDRAM controller (SDRAMC)
- DMA controller (DMAC)
- LCD controller (LCDC) bus interface
- Clock management unit (CMU) registers
- Bus arbiters

BCLK is required for bus and memory operations, therefore, it is always supplied to the modules listed above in normal mode. However, the BCLK supply in HALT mode can be disabled to reduce current consumption by setting BCLK_EN to 0 if the LCDC and DMA do not need bus operations.

D2 PCLK2_EN: PCLK2 Clock Enable Bit

Enables or disables the PCLK2 clock supply.

1 (R/W): Enabled (on) (default)

0 (R/W): Disabled (off)

The PCLK2 clock is used to operate the modules listed below.

- Prescaler (PSC Ch.1)
- 8-bit programmable timer Ch.1, 3, 5, 7 (T8 Ch.1, 3, 5, 7)
- Universal serial interface with LCD interface (USIL)
- Serial interface Ch.1 (FSIO Ch.1)
- Watchdog timer (WDT)
- Remote controller (REMC)
- Interrupt controller (ITC)
- I/O ports (GPIO)
- BBRAM
- Cache controller (CCU) registers
- Real-time clock (RTC) registers
- SRAM controller (SRAMC) registers
- SDRAM controller (SDRAMC) registers
- LCD controller (LCDC) registers

The PCLK2_EN default setting is 1, which enables the clock supply. If all the modules listed above can be stopped, disable the clock supply by setting PCLK2_EN to 0 to reduce current consumption.

D1 PCLK1_EN: PCLK1 Clock Enable Bit

Enables or disables the PCLK1 clock supply.

1 (R/W): Enabled (on) (default)

0 (R/W): Disabled (off)

The PCLK1 clock is used to operate the modules listed below.

- Prescaler (PSC Ch.0)
- 8-bit programmable timer Ch.0, 2, 4, 6 (T8 Ch.0, 2, 4, 6)
- 16-bit PWM timer Ch.0, 1 (T16A5 Ch.0, 1)
- 16-bit audio PWM timer (T16P)
- Universal serial interface (USI)
- Serial interface Ch.0 (FSIO Ch.0)
- A/D converter (ADC10)
- I²S (I2S)
- Misc registers (MISC)

The PCLK1_EN default setting is 1, which enables the clock supply. If all the modules listed above can be stopped, disable the clock supply by setting PCLK1_EN to 0 to reduce current consumption.

D0 GCLK_EN: GCLK Clock Enable Bit

Enables or disables the GCLK clock supply to the GE module.

1 (R/W): Enabled (on) (default)

0 (R/W): Disabled (off)

The GCLK_EN default setting is 1, which enables the clock supply. If the GE module can be stopped, disable the clock supply by setting GCLK_EN to 0 to reduce current consumption.

System Clock Division Ratio Select Register (CMU_SYCLKDIV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
System Clock Division Ratio Select Register (CMU_SYCLKDIV)	0x300105 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	MCLKDIV	MCLK clock divider select	1 1/2 0 1/1	0	R/W	Write-protected
		D3	–	reserved	–	–	–	0 when being read.
		D2-0	SYSCLKDIV[2:0]	System clock division ratio select	SYSCLKDIV[2:0] Division ratio	0x0	R/W	Clock source = OSC (OSC3, PLL, or OSC1) Write-protected
					0x7-0x6	1/1		
					0x5	1/32		
					0x4	1/16		
					0x3	1/8		
					0x2	1/4		
					0x1	1/2		
					0x0	1/1		

D[7:5] Reserved

D4 MCLKDIV: MCLK Clock Divider Select Bit

Selects the main system clock.

1 (R/W): SYSCLK/2

0 (R/W): SYSCLK/1 (default)

MCLK is the main system clock for S1C33L26. This bit selects either SYSCLK (selected with SYSCLKDIV[2:0]) or its halved clock.

When using the SDRAMC in double frequency mode (MCLK : SDCLK = 1 : 2), MCLK should be set to SYSCLK/2 (SYSCLK is used for the SDRAM clock).

D3 Reserved

D[2:0] SYSCLKDIV[2:0]: System Clock Division Ratio Select Bits

Selects a division ratio to set the system clock frequency. To reduce current consumption, operate the C33 PE Core and peripheral modules using the slowest possible clock speed.

6 CLOCK MANAGEMENT UNIT (CMU)

Table 6.10.5 System Clock Division Ratio

SYSLCKDIV[2:0]	Division ratio (OSC/n)
0x7–0x6	1/1
0x5	1/32
0x4	1/16
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

CMU_CLK Select Register (CMU_CMUCLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
CMU_CLK Select Register (CMU_CMUCLK)	0x300106 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.	
		D4–0	CMU_CLKSEL[4:0]	CMU_CLK select	CMU_CLKSEL[4:0]	CMU_CLK	0x0	R/W	OSC: system clock (OSC3, PLL, OSC1) Write-protected
					0xf–0xb	reserved			
					0xa	OSC/32			
					0x9	OSC/16			
					0x8	OSC/8			
					0x7	OSC/4			
					0x6	OSC/2			
					0x5	OSC/1			
					0x4	LCLK			
					0x3	BCLK			
					0x2	PLL			
					0x1	OSC1			
			0x0	OSC3					

D[7:5] Reserved

D[4:0] CMU_CLKSEL[4:0]: CMU_CLK Select Bits

Selects an internally generated clock to be output from the CMU_CLK pin to external devices.

Table 6.10.6 CMU_CLK Selections

CMU_CLKSEL[4:0]	CMU_CLK
0xf–0xb	Reserved
0xa	OSC/32
0x9	OSC/16
0x8	OSC/8
0x7	OSC/4
0x6	OSC/2
0x5	OSC/1
0x4	LCLK
0x3	BCLK
0x2	PLL
0x1	OSC1
0x0	OSC3

(Default: 0x0)

CMU_CLK can be selected at any time. However, switching over the clocks creates hazards.

Note: Settings other than those listed in Table 6.10.6 are reserved for testing. Do not set undescribed values to CMU_CLKSEL[4:0] as undesired clocks may output.

PLL Input Clock Division Ratio Select Register (CMU_PLLINDIV)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PLL Input Clock Division Ratio Select Register (CMU_PLLINDIV)	0x300107 (8 bits)	D7-4	–	reserved	–		–	–	0 when being read.
		D3-0	PLLINDIV [3:0]	PLL input clock division ratio select	PLLINDIV[3:0]	Division ratio	0x7	R/W	Clock source = OSC3 Write-protected
					0xf-0xa	1/8			
					0x9	1/10			
					0x8	1/9			
					0x7	1/8			
					0x6	1/7			
					0x5	1/6			
					0x4	1/5			
					0x3	1/4			

D[7:4] Reserved

D[3:0] PLLINDIV[3:0]: PLL Input Clock Division Ratio Select Bits

Selects the PLL input clock (OSC3 division ratio).

Table 6.10.7 PLL Input Clock (OSC3 Division Ratio) Selections

PLLINDIV[3:0]	Division ratio (OSC3/n)
0xf-0xa	1/8
0x9	1/10
0x8	1/9
0x7	1/8
0x6	1/7
0x5	1/6
0x4	1/5
0x3	1/4
0x2	1/3
0x1	1/2
0x0	1/1

(Default: 0x7)

- Notes:**
- The PLL input clock can only be selected when the PLL is turned off (PLLPOWER/CMU_PLLCTL0 register = 0) and the clock source is other than the PLL (CLKSEL[1:0]/CMU_OSCSEL register is not 0x2). 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.”

PLL Control Register 0 (CMU_PLLCTL0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PLL Control Register 0 (CMU_PLLCTL0)	0x300108 (8 bits)	D7-4	PLLN[3:0]	PLL multiplication rate setup	PLLN[3:0]	Multiplication rate	0x0	R/W	Write-protected
					0xf	x16			
					0xe	x15			
					0xd	x14			
					0xc	x13			
					0xb	x12			
					0xa	x11			
					0x9	x10			
					0x8	x9			
					0x7	x8			
0x6	x7								
0x5	x6								
0x4	x5								
0x3	x4								
0x2	x3								
0x1	x2								
0x0	x1								
		D3-2	PLLV[1:0]	PLL V-divider setup	PLLV[1:0]	W	0x1	R/W	
					0x3	8			
					0x2	4			
					0x1	2			
					0x0	Not allowed			
		D1	-	reserved	-	-	-	-	0 when being read.
		D0	PLLPOWR	PLL enable	1 Enable 0 Disable	0	R/W	Write-protected	

Note: Make sure that the PLL is turned off (PLLPOWR = 0) before altering D[7:4] in this register.

D[7:4] PLLN[3:0]: PLL Multiplication Rate Setup Bits

Sets the frequency multiplication rate of the PLL.

Table 6.10.8 PLL Frequency Multiplication Rates

PLLN[3:0]	Multiplication rate
0xf	x16
0xe	x15
0xd	x14
0xc	x13
0xb	x12
0xa	x11
0x9	x10
0x8	x9
0x7	x8
0x6	x7
0x5	x6
0x4	x5
0x3	x4
0x2	x3
0x1	x2
0x0	x1

(Default: 0x0)

PLL output clock frequency = PLL input clock frequency × multiplication rate

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 6.10.9 W Value Settings

PLLV[1:0]	W
0x3	8
0x2	4
0x1	2
0x0	Setting prohibited

(Default: 0x1)

D1 Reserved**D0** **PLLPOWR: PLL Enable Bit**

Turns the PLL on or off.

1 (R/W): On

0 (R/W): Off (Default)

Up to 200 μ s is required before the PLL output clock stabilizes after PLLPOWR is set to 1. Specify this wait time in the program before changing the system clock source to the PLL.

When not using the PLL, turn the PLL off (power-down mode) to reduce current consumption.

PLL Control Register 1 (CMU_PLLCTL1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PLL Control Register 1 (CMU_PLLCTL1)	0x300109 (8 bits)	D7-4	PLLVC[3:0]	PLL VCO Kv setup	PLLVC[3:0]	f _{vco} [MHz]	0x1	R/W	Write-protected
					0x8	360 < f _{vco} ≤ 400			
					0x7	320 < f _{vco} ≤ 360			
					0x6	280 < f _{vco} ≤ 320			
					0x5	240 < f _{vco} ≤ 280			
					0x4	200 < f _{vco} ≤ 240			
					0x3	160 < f _{vco} ≤ 200			
					0x2	120 < f _{vco} ≤ 160			
					0x1	100 ≤ f _{vco} ≤ 120			
					Other	Not allowed			
D3-0	PLLRS[3:0]	PLL LPF resistance setup	PLLRS[3:0]	f _{REFCK} [MHz]	0x8	R/W			
			0xa	5 ≤ f _{REFCK} < 20					
			0x8	20 ≤ f _{REFCK} ≤ 150					
			Other	Not allowed					

Note: Make sure that the PLL is turned off (PLLPOWR/CMU_PLLCTL0 register = 0) before altering this register.

D[7:4] **PLLVC[3:0]: PLL VCO Kv Setup Bits**

Sets the VCO Kv circuit constant (VC value) according to the range of f_{vco} frequencies obtained by <Output clock frequency × W>.

Table 6.10.10 VC Value Settings

PLLVC[3:0]	f _{vco} [MHz]
0x8	360 < f _{vco} ≤ 400
0x7	320 < f _{vco} ≤ 360
0x6	280 < f _{vco} ≤ 320
0x5	240 < f _{vco} ≤ 280
0x4	200 < f _{vco} ≤ 240
0x3	160 < f _{vco} ≤ 200
0x2	120 < f _{vco} ≤ 160
0x1	100 ≤ f _{vco} ≤ 120
Other	Setting prohibited

(Default: 0x1)

D[3:0] **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 6.10.11 RS Value Settings

PLLRS[3:0]	f _{REFCK} [MHz]
0xa	5 ≤ f _{REFCK} < 20
0x8	20 ≤ f _{REFCK} ≤ 150
Other	Setting prohibited

(Default: 0x8)

PLL Control Register 2 (CMU_PLLCTL2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PLL Control Register 2 (CMU_PLLCTL2)	0x30010a (8 bits)	D7–6	PLLCS[1:0]	PLL LPF capacitance	0x0	0x0	R	
		D5	PLLBYP	PLL bypass mode	0	0	R	
		D4–0	PLLCP[4:0]	PLL charge pump current	0x10	0x10	R	

D[7:6] PLLCS[1:0]: PLL LPF Capacitance Bits

Indicates the LPF capacitance value (CS value). (Default: 0x0)

D5 PLLBYP: PLL Bypass Mode Bit

Indicates the mode when the PLL is bypassed. (Default: 0)

D[4:0] PLLCP[4:0]: PLL Charge Pump Current Bits

Indicates the charge pump current value (CP value). (Default: 0x10)

SSCG Macro Control Register 0 (CMU_SSCG0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SSCG Macro Control Register 0 (CMU_SSCG0)	0x30010c (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SSMCON	SSCG enable	1 Enable 0 Disable	0	R/W	Write-protected

D[7:1] Reserved

D0 SSMCON: SSCG Enable 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, allowing the clock to bypasses the SSCG.

SSCG Macro Control Register 1 (CMU_SSCG1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SSCG Macro Control Register 1 (CMU_SSCG1)	0x30010d (8 bits)	D7–4	SSMCITM [3:0]	SSCG interval timer (ITM) setting	0x0 to 0xf	X	R/W	Write-protected
		D3–0	SSMCIDT [3:0]	SSCG maximum frequency change width setting	0x0 to 0xf	X	R/W	

Note: When the PLL is off, the initial values and the written values cannot be read correctly from this register since the source clock is not supplied to the SSCG (different values are read out). The correct values can be read out when the PLL is turned on.

D[7:4] SSMCITM[3:0]: SSCG Interval Timer Setting Bits

Sets the frequency change cycle in SS modulation of the SSCG. (See Section 6.5, “SSCG.”)

Always set these bits to 0x1. (Default: undefined)

D[3:0] SSMCIDT[3:0]: SSCG Maximum Frequency Change Width Setting Bits

Sets the maximum frequency change width in SS modulation of the SSCG. (See Section 6.5, “SSCG.”)

Table 6.10.12 Maximum Frequency Change Width Settings

PLL output clock frequency f [MHz]	SSMCIDT[3:0]
$f \leq 19.8$	0xf
$19.8 < f \leq 21.2$	0xe
$21.2 < f \leq 22.5$	0xd
$22.5 < f \leq 24.2$	0xc
$24.2 < f \leq 25.9$	0xb
$25.9 < f \leq 28.4$	0xa
$28.4 < f \leq 30.8$	0x9
$30.8 < f \leq 34.2$	0x8
$34.2 < f \leq 37.8$	0x7
$37.8 < f \leq 43.1$	0x6
$43.1 < f \leq 48.9$	0x5
$48.9 < f \leq 58.5$	0x4
$58.5 < f \leq 69.7$	0x3
$69.7 < f \leq 90.0$	0x2
–	0x1
–	0x0

(Default: undefined)

Note: SSMCIDT[3:0] must be set according to the PLL output clock frequency as shown in Table 6.10.12. Using the SSCG with an improper setting may cause a malfunction of the IC.

CMU Write Protect Register (CMU_PROTECT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CMU Write Protect Register (CMU_PROTECT)	0x300110 (8 bits)	D7–0	CMUP[7:0]	CMU register write-protect flag	Writing 10010110 (0x96) removes the write protection of the CMU registers (0x300100–0x30010d). Writing another value set the write protection.	0x0	R/W	

D[7:0] CMUP[7:0]: CMU Register Write-Protect Flag Bits

Enables or disables write protection of the CMU control registers (0x300100–0x30010d).

0x96 (R/W): Disable write protection

Other than 0x96 (R/W): Write-protect the register (default: 0x0)

Before altering any CMU control register, write data 0x96 to CMUP[7:0] to disable write protection. If CMUP[7:0] is set to other than 0x96, even if an attempt is made to alter any CMU control register by executing a write instruction, the content of the register will not be altered even though the instruction may have been executed without a problem. Once CMUP[7:0] is set to 0x96, the CMU control registers can be rewritten any number of times until being reset to other than 0x96. When rewriting the CMU control registers has finished, CMUP[7:0] should be set to other than 0x96 to prevent accidental writing to the CMU registers.

7 Prescaler (PSC)

7.1 PSC Module Overview

The S1C33L26 incorporates a prescaler (PSC) module to generate clocks for timer and serial interface operations. The PSC module consists of two frequency dividers (PSC Ch.0 and PSC Ch.1) that generate 15 different frequencies by dividing the PCLK1 and PCLK2 clock supplied from the clock management unit (CMU) into 1/1 to 1/16K. The peripheral modules to which the clock is supplied include clock-select registers enabling selection of one as a count or operation clock.

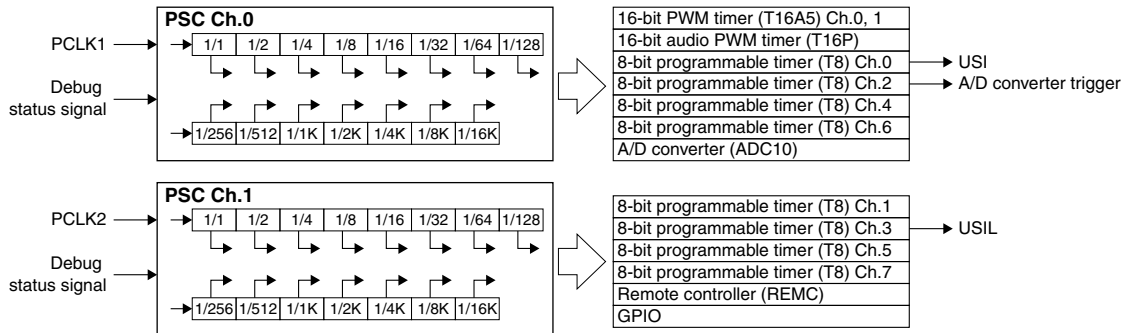


Figure 7.1.1 Prescaler Configuration

PSC Ch.0 and PSC Ch.1 are controlled by PRUN/PSC_CTL register. To operate the prescalers, write 1 to PRUN. Writing 0 to PRUN stops the prescalers. Stopping the prescalers while the timer and interface modules are halted enables the current consumption to be reduced. The prescalers are stopped at initial reset.

Note: PCLK1 and PCLK2 must be supplied from the CMU to use the PSC Ch.0 and PSC Ch.1, respectively.

The PSC module features another control bit, PRUND/PSC_CTL register, which specifies prescaler operations in debug mode. Setting PRUND to 1 operates the prescalers in debug mode. Setting it to 0 stops the prescalers when the C33 PE Core enters debug mode. Set PRUND to 1 when operating the timer and interface modules during debugging.

7.2 Control Register Details

Table 7.2.1 PSC Register

Address	Register name	Function
0x300e00	PSC_CTL PSC Control Register	Control prescaler

The prescaler register is an 8-bit register.

Note: When data is written to the register, the “Reserved” bits must always be written as 0 and not 1.

PSC Control Register (PSC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PSC Control Register (PSC_CTL)	0x300e00 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	PRUND	Prescaler run/stop in debug mode	1 Run 0 Stop	0	R/W	
		D0	PRUN	Prescaler run/stop control	1 Run 0 Stop	0	R/W	

D[7:2] Reserved

7 PRESCALER (PSC)

D1 **PRUND: Prescaler Run/Stop in Debug Mode Bit**

Selects prescaler operations in debug mode.

1 (R/W): Run

0 (R/W): Stop (default)

Setting PRUND to 1 operates PSC Ch.0 and PSC Ch.1 even in debug mode. Setting it to 0 stops PSC Ch.0 and PSC Ch.1 when the C33 PE Core enters debug mode. Set PRUND to 1 to use the modules listed below during debugging.

D0 **PRUN: Prescaler Run/Stop Control Bit**

Starts or stops the prescaler.

1 (R/W): Start operation

0 (R/W): Stop (default)

Write 1 to PRUN to operate PSC Ch.0 and PSC Ch.1. Write 0 to PRUN to stop PSC Ch.0 and PSC Ch.1. To reduce current consumption, stop PSC Ch.0 and PSC Ch.1 if the modules listed below are already stopped.

Modules that use PSC Ch.0 output clocks

- 16-bit PWM timer (T16A5) Ch.0, 1
- 16-bit audio PWM timer (T16P)
- 8-bit programmable timer (T8) Ch.0 (clock source for USI)
- 8-bit programmable timer (T8) Ch.2 (trigger source for A/D converter)
- 8-bit programmable timer (T8) Ch.4
- 8-bit programmable timer (T8) Ch.6
- A/D converter (ADC10)

Modules that use PSC Ch.1 output clocks

- 8-bit programmable timer (T8) Ch.1
- 8-bit programmable timer (T8) Ch.3 (clock source for USIL)
- 8-bit programmable timer (T8) Ch.5
- 8-bit programmable timer (T8) Ch.7
- Remote controller (REMC)
- I/O ports (GPIO)

8 Real-Time Clock (RTC)

8.1 RTC Module Overview

The S1C33L26 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.
- Includes read buffers to prevent carry over at reading.
- 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 1/512 second, 1/256 second, 1/128 second, 1/64 second, 1 second, 1 minute, or 1 hour. (Level interrupt mode)
- Independent power supply, so that the RTC can continue operating even when system power is turned off.
- A built-in OSC1 oscillator circuit (crystal oscillator or external clock input) that generates a 32.768-kHz (typ.) operating clock. (See the “Clock Management Unit (CMU)” chapter.)
- Provides the #STBY and WAKEUP pins to control the system power supply.

Figure 8.1.1 shows a block diagram of the RTC.

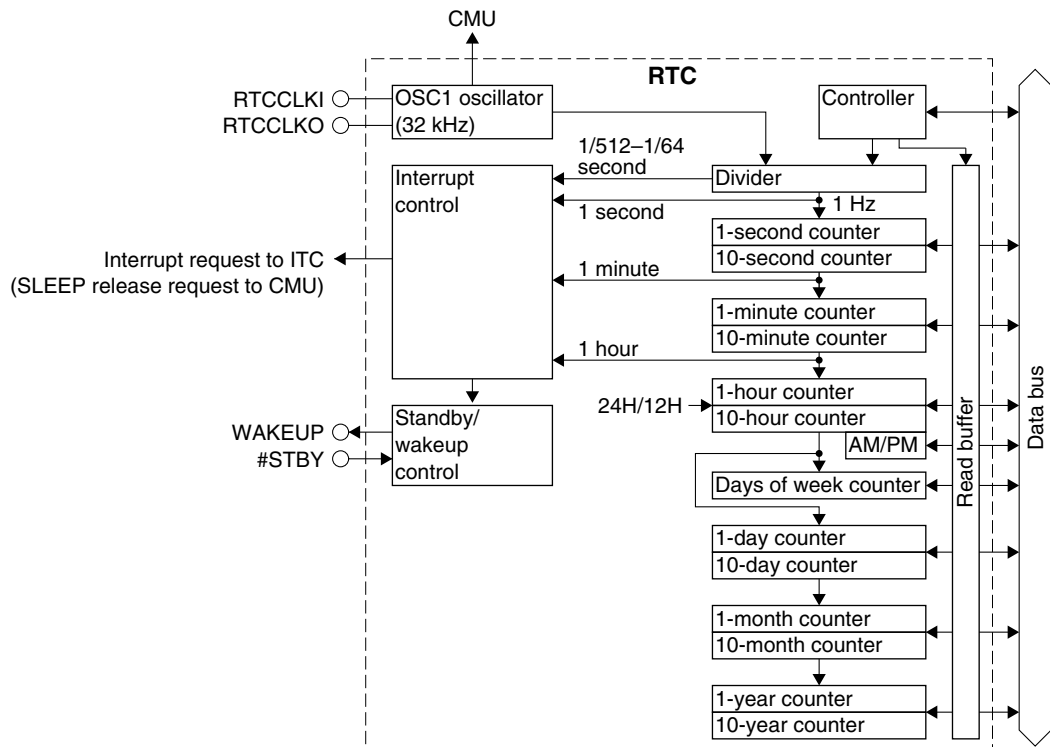


Figure 8.1.1 RTC Block Diagram

8.2 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]/RTC_SEC register.

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]/RTC_SEC register.

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]/RTC_MIN register.

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]/RTC_MIN register.

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]/RTC_HOUR register.

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]/RTC_HOUR register.

When 12-hour mode is selected, RTCAP/RTC_HOUR register 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 is fixed to 0.

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]/RTC_DAY register.

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]/RTC_DAY register.

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]/RTC_MONTH register.

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/RTC_MONTH register.

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]/RTC_YEAR register.

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]/RTC_YEAR register.

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]/RTC_WEEK register.

The correspondence between the counter values and days of the week can be set in a program as desired. Table 8.2.1 lists the basic correspondence.

Table 8.2.1 Correspondence between Counter Values and Days of the Week

RTCWK[2:0]	Days of the week
0x6	Saturday
0x5	Friday
0x4	Thursday
0x3	Wednesday
0x2	Tuesday
0x1	Monday
0x0	Sunday

(Default: indeterminate)

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 8.3.2, "RTC Initial Sequence."

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.

8.3 RTC Control

8.3.1 Operating Clock Control

Counter clock

The RTC is clocked by the 32.768-kHz (typ.) OSC1 clock. The OSC1 clock is always supplied from the OSC1 oscillator circuit (even in HALT/SLEEP mode).

Register clock

The PCLK2 clock is used for accessing the RTC control registers. To setup the registers, this clock is required. After the registers are set up, the clock supply can be stopped by setting the CMU.

Setting the wait cycles for accessing the RTC module

In order to access the RTC registers properly even if the system operates with a high-speed clock, a wait cycle can be inserted into the RTC access cycle. The number of MCLK cycles to be inserted as a wait cycle can be specified using RTCWT[2:0]/MISC_RTCWT register.

Table 8.3.1.1 Number of Wait Cycles during RTC Access

RTCWT[2:0]	Number of wait cycles
0x7	7 cycles
0x6	6 cycles
0x5	5 cycles
0x4	4 cycles
0x3	3 cycles
0x2	2 cycles
0x1	1 cycle
0x0	0 cycles (cannot be set)

(Default: 0x7)

The S1C33L26 is able to operate with $RTCWT[2:0] \geq 1$.

8.3.2 RTC Initial Sequence

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. Software reset
Write 1 to RTCRST/RTC_CNTL0 register and then write 0 to reset the RTC.
4. Confirming accessibility status of the RTC
See Section 8.3.5, "Counter Hold and Busy Flag."
5. Disabling the divider
Write 1 to RTCSTP/RTC_CNTL0 register to stop the divider in the RTC module.
6. Setting the RTC interrupt
Set the RTC_INTMODE register.
Be sure to set RTCIMD to 1 (level sense).
7. Setting the date and time
Set the RTC_SEC, RTC_MIN, RTC_HOUR, RTC_DAY, RTC_MONTH, RTC_YEAR, and RTC_WEEK registers. Then, write 0 to RTCHLD/RTC_CNTL1 register to release the 1-second, 10-second, 1-minute, 10-minute, 1-hour, 10-hour, 1-day, 10-day, 1-month, 10-month, 1-year, 10-year, and days of week counters from hold status.

8. Starting the divider

Write 0 to RTCSTP/RTC_CNTL0 register to run the divider in the RTC module.

8.3.3 12/24-hour Mode and Counter Settings

12-hour/24-hour mode selection

Whether to use the time clock in 12-hour or 24-hour mode can be selected using RTC24H/RTC_CNTL0 register.

RTC24H = 1: 24-hour mode

RTC24H = 0: 12-hour mode

The count range of hour counters changes with this selection.

Basically, this setting should be changed while the counters are idle. RTC24H 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 may corrupt count data for the hours, days, months, years or days of the week.

Therefore, once RTC24H 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/RTC_HOUR register that indicates A.M. or P.M. is enabled.

RTCAP = 0: A.M.

RTCAP = 1: P.M.

For 24-hour mode, RTCAP is fixed to 0.

When setting the time of day, write either of the values above to this bit to specify A.M. or P.M.

Counter settings

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 8.3.5, “Counter Hold and Busy Flag.”

8.3.4 Start/Stop and Software Reset

Starting and stopping divider

The RTC starts counting when RTCSTP/RTC_CNTL0 register is set to 0, and stops counting when this bit is set to 1.

The RTC is started/stopped by writing data to RTCSTP 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.

Software reset

RTCRST/RTC_CNTL0 register is the software reset bit used to reset the items shown below.

- Divider (32 kHz to 2 Hz bits)
- Interrupt request signal
- WAKEUP signal
- Some register bits (see Section 8.6 for the control bits and their initial values.)

To perform software reset, write 1 to RTCRST and then write back to 0.

The registers initialized by software reset must be re-programmed after releasing from reset status.

The divider bits above are cleared 0. The output signals above become inactive while RTCRST is set to 1 and are enabled to be output again after RTCRST is set to 0.

8.3.5 Counter Hold and Busy Flag

If 1 is carried over when writing the counters, the counter value may be corrupted. Therefore, whether counters are in a carry (busy) state should be checked before writing data to the count registers. For this purpose, control bits RTCBSY/RTC_CNTL1 register and RTCHLD/RTC_CNTL1 register are provided.

RTCBSY is a read-only flag indicating that carry is taking place. RTCBSY is set to 1 when carry is taking place; otherwise, it is 0. RTCBSY should be confirmed as being 0 before accessing the counters to ensure that the correct value will be set.

Writing 1 to RTCHLD suspends the counter operations. Note, however, that writing 1 to RTCHLD is ignored if RTCBSY is set to 1.

RTCBSY = 0 (RTC accessible)

When a value of 0 is read from RTCBSY after writing 1 to RTCHLD, it means that carry is not taking place. In this state, counter data can be written to.

After 1 is written to RTCHLD, the counters stop operating. So RTCBSY is fixed at 0, as carry will not take place. In this case, the counter hold function is also actuated, with a carry over of 1 to the 1-second counter disabled in hardware. The divider (counter for less than one second) continues operating.

Write data to the counter registers.

After writing data, reset RTCHLD to 0.

If 1 is being carried over when data is being written to a counter in the hold state, 1 second is automatically added to correct the counter values when RTCHLD is reset to 0. This correction is only effective for 1 second and no correction is conducted on the carry encountered in the second time and on. In this case, the timekeeping data gets out of order. Therefore, be sure to reset RTCHLD to 0 as soon as possible after completing the required write operation.

RTCBSY = 1 (RTC is busy)

When a value of 1 is read from RTCBSY after writing 1 to RTCHLD, it means that carry is taking place.

In this case, writing 1 to RTCHLD is ignored and RTCHLD retains 0.

A period of 4 ms per second is required for a carry over of 1 to the counters. In this case, [A] repeat writing 1 to RTCHLD and checking RTCBSY or [B] write 1 to RTCHLD and check RTCBSY after waiting for 4 ms.

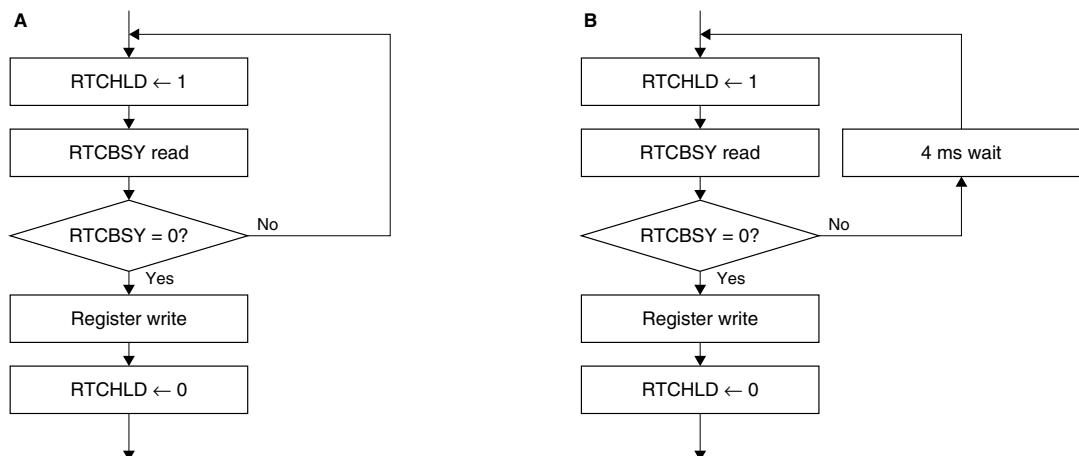


Figure 8.3.5.1 Procedure for Checking whether the RTC is Busy

8.3.6 30-second Correction

The description “30-second correction” means resetting the seconds to 0 and adding 1 to the minutes when seconds of the time clock are in the range of 30 to 59 seconds. When in the range of 0 to 29 seconds, the RTC resets the seconds to 0 but it does not change the minutes. 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/RTC_CNTL0 register.

Writing 1 to RTCADJ 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 is set to 1, it remains set for the 4-ms period required for this processing, then automatically returns to 0. To check whether the 30-second correction processing has completed or not, [A] repeat checking RTCADJ or [B] check RTCADJ after waiting for 4 ms.

Accessing the counters while RTCADJ = 1 is prohibited. Writing 0 to RTCADJ and writing 1 to RTCRST are also prohibited, because it would cause the RTC to operate erratically.

Writing 1 to RTCADJ when RTCBSY is 1 may corrupt the counter values. Always make sure that RTCBSY is set to 0 before writing 1 to RTCADJ.

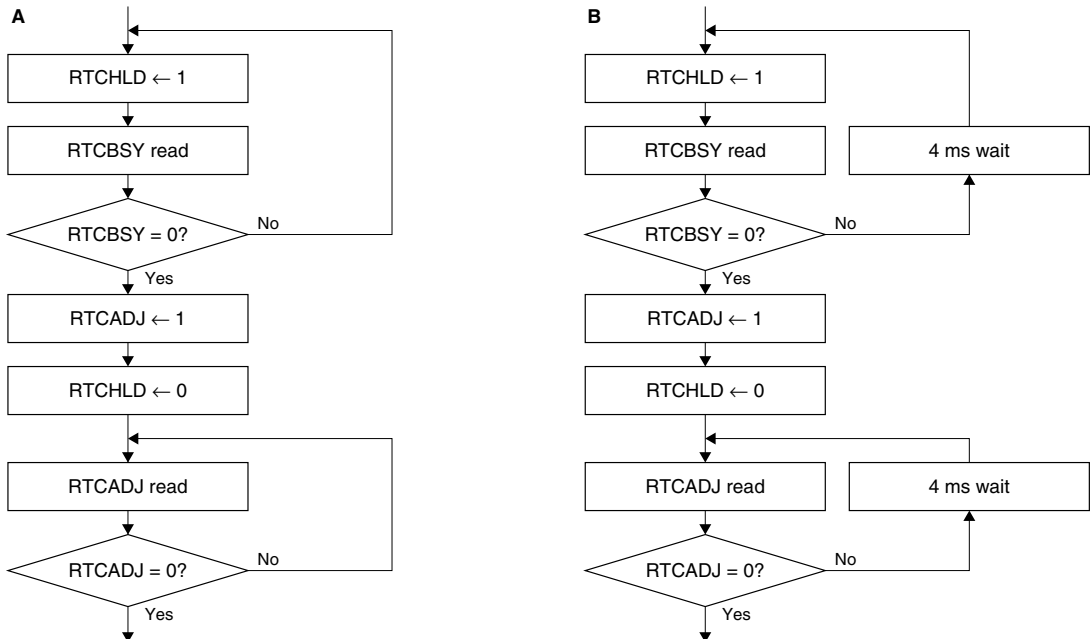


Figure 8.3.6.1 Procedure for Executing 30-second Correction

8.3.7 Counter Read

In order to prevent carry over during reading counters, the RTC includes a read buffer to hold counter data.

Before reading counter data, set RTCRDHLD/RTC_CNTL1 register to 1 to load the current counter data to the read buffer.

While RTCRDHLD is set to 1, the buffered data is read out from the counter registers. Be sure to reset RTCRDHLD to 0 after the buffered data is read out. This operation does not affect the counters. The counters keep counting while RTCRDHLD is set to 1.

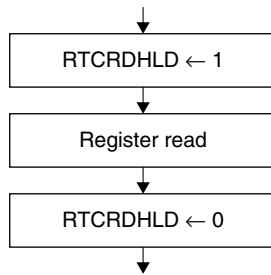


Figure 8.3.7.1 Procedure for Reading Counters

8.4 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 cancel SLEEP mode.

This section describes the internal interrupt control function of the RTC. To generate interrupts to the CPU, the interrupt controller (ITC) must also be set up. For details on how to control the ITC, see the “Interrupt Controller (ITC)” chapter. For details on how to cancel SLEEP mode using an interrupt, see the “Clock Management Unit (CMU)” chapter.

Interrupt cycle setting

The interrupt cycle (in which the RTC outputs interrupt requests at specific intervals) can be selected from seven choices listed in Table 8.4.1 by using RTCT[2:0]/RTC_INTMODE register.

Table 8.4.1 Interrupt Cycle Settings

RTCT[2:0]	Interrupt cycle
0x7	Reserved
0x6	1/128 second
0x5	1/256 second
0x4	1/512 second
0x3	1 hour
0x2	1 minute
0x1	1 second
0x0	1/64 second

RTCT[2:0] should be set while RTC interrupts are disabled. (See the procedure for enabling and disabling interrupts described below.)

Setting interrupt conditions

The RTC of the S1C33L26 supports level-sensed interrupt only.

Enabling and disabling interrupts

The RTC interrupt requests output to the ITC are enabled by setting RTCIEN/RTC_INTMODE register to 1 and disabled by setting it to 0.

RTC interrupts will be generated according to the divider and counter status and the time between writing 1 to RTCIEN and the first interrupt request is not fixed. Use the second and subsequent interrupts as valid.

Interrupt status

When the RTC is up and running, RTCIRQ/RTC_INTSTAT register is set at the cyclic interrupt intervals set up by RTCT[2:0]. When RTC interrupts are enabled by RTCIEN, interrupt requests are sent to the ITC.

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.

When RTCIEN is set to 0 (interrupt disabled), RTCIRQ is fixed at 0 (will not be set to 1).

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 to 0 (interrupt disabled) to prevent the occurrence of unwanted RTC interrupts. Also be sure to write 1 to RTCIRQ to reset it.

When a software reset is performed (RTCST → 1 → 0), RTCIRQ and RTCIEN are reset to 0 to disable the interrupt request output. Also RTCT[2:0] is reset to 0x1.

8.5 WAKEUP and #STBY Pins

The S1C33L26 has a battery backup function that allows the system to turn the system power (LV_{DD}, PLLV_{DD}, HV_{DD}, AV_{DD}) off with the RTC (including the OSC1 oscillator circuit) kept active and the BBRAM data maintained by supplying RTCV_{DD}. The RTC provides the WAKEUP and #STBY pins used for controlling this function.

The #STBY pin is used to disconnect the circuits driven with RTCV_{DD} (RTC, OSC1, and BBRAM) from the other circuits driven with LV_{DD}, PLLV_{DD}, HV_{DD}, and AV_{DD} (including the control registers for RTC and OSC1). The #STBY pin must be set to a high level during normal operation. Setting the #STBY pin to a low level from outside the IC disconnects the RTCV_{DD} circuits from the system allowing the system power (LV_{DD}, PLLV_{DD}, HV_{DD}, AV_{DD}) turned off.

The WAKEUP pin is an output pin of which the output can be controlled by the RTC interrupt or software. This output can control the external regulator to turn the system power (LV_{DD}, PLLV_{DD}, HV_{DD}, AV_{DD}) on and off. Note that leakage current flows to the RTCV_{DD} system if the system power is turned off when the #STBY pin is set to a high level. Therefore, the #STBY pin must be set to a low level before the system power is turned off.

Figure 8.5.1 shows an example of system standby/wakeup circuit using the WAKEUP and #STBY pins.

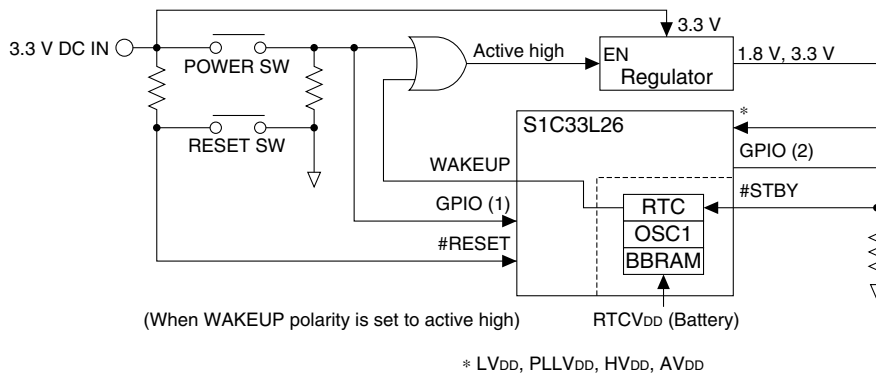


Figure 8.5.1 Example of System Standby/Wakeup Circuit

Note: The WAKEUP pin does not exist in the TQFP15-128pin package model.

Selecting the WAKEUP signal polarity

Use WUP_POL/RTC_WAKEUP register to select the WAKEUP output level when it is asserted by an RTC interrupt or software control.

The WAKEUP output is configured to active high signal when WUP_POL is set to 0 or active low signal when WUP_POL is set to 1. WUP_POL is not initialized at initial reset, therefore, it must be initialized with software when using the WAKEUP output.

Controlling the WAKEUP output

Controlling by an RTC interrupt

When the cause of RTC interrupt that has been selected with software (see Section 8.4) occurs, the WAKEUP signal is asserted similar to the interrupt request signal. The RTC maintains the WAKEUP signal at the active level until the system resumes operating and clears the RTC interrupt status bit RTCIRQ/RTC_INTSTAT register. The WAKEUP signal will be negated after RTCIRQ is cleared.

Software control

The WAKEUP output can also be controlled using WUP_CTL/RTC_WAKEUP register.

The WAKEUP signal is asserted by setting WUP_CTL to 1 and is negated by setting WUP_CTL to 0. WUP_CTL is not initialized at initial reset, therefore, it must be set to 1 (active) at the beginning with the initialize routine.

The table below shows the WAKEUP signal status according to the control bit.

Table 8.5.1 WAKEUP Signal Status

Control bit settings			WAKEUP pin status
WUP_POL	WUP_CTL	RTCIRQ	
1	1	1	0 (Low)
1	1	0	0 (Low)
1	0	1	0 (Low)
1	0	0	1 (High)
0	1	1	1 (High)
0	1	0	1 (High)
0	0	1	1 (High)
0	0	0	0 (Low)

When a software reset is performed (RTCRST → 1 → 0), WUP_CTL and WUP_POL are reset to 0 to set the WAKEUP signal to 0.

Control procedures

The following shows some power control procedures using the system standby/wakeup circuit shown in Figure 8.5.1. The description below assumes that the power (3.3 V) is supplied to the regulator and the WAKEUP signal polarity is set to active high.

Power On using the POWER SW

- (1) Press the POWER SW. The switch must be held down until Step (5) has completed.
- (2) The regulator is enabled to output voltage and the 1.8 V (and 3.3 V) voltage is supplied to the S1C33L26 LV_{DD}, PLLV_{DD}, HV_{DD}, and AV_{DD} pins.
- (3) The CPU starts operating and executes the initialize routine after power-on reset.
- (4) Configure GPIO (2) as an output port and set the port output level to 1 (high). This signal is fed to the #STBY pin resulting that the RTCV_{DD} system circuits will be connected to the system.
- (5) Write 0x2 to the RTC_WAKEUP register to set the WAKEUP polarity to active high and enable the WAKEUP pin to output 1 (high). This control fixes the regulator output to be enabled, thus the POWER SW can be released (turned off).
- (6) Read the key from the specific BBRAM location and check whether the backup data is valid or not (e.g. valid if 0xaa). Then if valid, read the backup data from the BBRAM.
- (7) Clear the key located in the BBRAM (e.g. write a value such as 0x00).
- (8) Execute other processing.

Keep the #STBY input = 1 and WAKEUP output = 1 conditions while the IC is operating.

Power Off using the POWER SW

The following procedure should be started under the above condition (#STBY input = 1 and WAKEUP output = 1).

- (1) Press the POWER SW.
- (2) The GPIO (1) port inputs 1 (high). Detect this status by reading the input data or using an interrupt from the port, and execute the sequence to place the S1C33L26 into battery backup mode.
- (3) Copy the data required to be saved into the BBRAM. In addition to this, write a key for indicating that the backup data is valid (e.g. 0xaa) to the specific location in the BBRAM.
- (4) Set the RTC interrupt conditions and enable the interrupt. (when restarting the system using an RTC interrupt)
- (5) Write 0x0 to the RTC_WAKEUP register to set the WAKEUP pin to output 0 (low).
- (6) Set the GPIO (2) port to output 0 (low). This signal is fed to the #STBY pin resulting that the RTCV_{DD} system circuits will be disconnected from the system.

- (7) (After the POWER SW is turned off if it is still on,) The regulator stops generating 1.8 V (and 3.3 V) and the power of the S1C33L26 except the RTCV_{DD} is turned off.

When automatically turning the system power off by software control, start the above procedure from Step (3).

Power On using an RTC interrupt

- (1) When an RTC interrupt occurs, the WAKEUP output level goes 1 (high).
- (2) The regulator is enabled to output voltage and the 1.8 V (and 3.3 V) voltage is supplied to the S1C33L26 LV_{DD}, PLLV_{DD}, HV_{DD}, and AV_{DD} pins.
- (3) The CPU starts operating and executes the initialize routine after power-on reset.
- (4) Configure GPIO (2) as an output port and set the port output level to 1 (high). This signal is fed to the #STBY pin resulting that the RTCV_{DD} system circuits will be connected to the system.
- (5) Write 0x2 to the RTC_WAKEUP register to set the WAKEUP polarity to active high and enable the WAKEUP pin to output 1 (high). This control fixes the regulator output to be enabled.
- (6) Reset RTCIRQ/RTC_INTSTAT register to 0.
- (7) Read the key from the specific BBRAM location and check whether the backup data is valid or not (e.g. valid if 0xaa). Then if valid, read the backup data from the BBRAM.
- (8) Clear the key located in the BBRAM (e.g. write a value such as 0x00).
- (9) Execute other processing.

8.6 Details of Control Registers

Table 8.6.1 RTC Register List

Address	Register name		Function
0x300a00	RTC_INTSTAT	RTC Interrupt Status Register	Indicates RTC interrupt status.
0x300a01	RTC_INTMODE	RTC Interrupt Mode Register	Sets up RTC interrupt modes.
0x300a02	RTC_CNTL0	RTC Control 0 Register	Controls the RTC.
0x300a03	RTC_CNTL1	RTC Control 1 Register	
0x300a04	RTC_SEC	RTC Second Register	Second counter data
0x300a05	RTC_MIN	RTC Minute Register	Minute counter data
0x300a06	RTC_HOUR	RTC Hour Register	Hour counter data
0x300a07	RTC_DAY	RTC Day Register	Day counter data
0x300a08	RTC_MONTH	RTC Month Register	Month counter data
0x300a09	RTC_YEAR	RTC Year Register	Year counter data
0x300a0a	RTC_WEEK	RTC Days of Week Register	Days of week counter data
0x300a0f	RTC_WAKEUP	RTC Wakeup Configuration Register	Sets up RTC wakeup conditions.

The following describes each RTC register. These are all 8-bit registers.

- Notes:**
- When data is written to the register, the “Reserved” bits must always be written as 0 and not 1.
 - 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 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”

RTC Interrupt Status Register (RTC_INTSTAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Interrupt Status Register (RTC_INTSTAT)	0x300a00 (8 bits)	D7-1	–	reserved	–	–	–	0 when being read.
		D0	RTCIRQ	Interrupt status	1 Occurred 0 Not occurred	X (0)	R/W	Reset by writing 1.

Init.: () indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

D[7:1] 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 (software reset value)
- 1 (W): Resets this bit to 0
- 0 (W): Has no effect

This bit is set at cyclic interrupt intervals set up by RTCT[2:0]/RTC_INTMODE register. When RTC interrupts have been enabled by RTCIEN/RTC_INTMODE register at this time, an interrupt request is sent to the ITC.

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.

RTC Interrupt Mode Register (RTC_INTMODE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Interrupt Mode Register (RTC_INTMODE)	0x300a01 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.
		D4-2	RTCT[2:0]	RTC interrupt cycle setup	RTCT[2:0] Cycle	X	R/W	
					0x7 reserved	(0x1)		
					0x6 1/128 second			
					0x5 1/256 second			
					0x4 1/512 second			
					0x3 1 hour			
					0x2 1 minute			
					0x1 1 second			
					0x0 1/64 second			
	D1	RTCIMD	reserved	1	X (1)	R/W	Always set to 1.	
	D0	RTCIEEN	RTC interrupt enable	1 Enable 0 Disable	X (0)	R/W		

Init.: () indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

D[7:5] Reserved**D[4:2] RTCT[2:0]: RTC Interrupt Cycle Setup Bits**

These bits select the RTC interrupt cycle.

Table 8.6.2 Interrupt Cycle Settings

RTCT[2:0]	Interrupt cycle
0x7	Reserved
0x6	1/128 second
0x5	1/256 second
0x4	1/512 second
0x3	1 hour
0x2	1 minute
0x1	1 second
0x0	1/64 second

(Default: indeterminate, software reset: 0x1)

RTCIRQ/RTC_INTSTAT register is set by a count-up pulse of the interrupt cycle counter selected. When RTC interrupts are enabled by RTCIEN, an interrupt request is sent to the ITC.

RTCT[2:0] should be set while RTC interrupts are disabled. (These bits may also be set simultaneously when RTC interrupts are enabled.)

D1 RTCIMD: Reserved (Always be sure to set to 1.)**D0 RTCIEN: 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 (software reset value)

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 and SLEEP mode cannot be turned off.

Note: The value of RTCIEN 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.

RTC Control 0 Register (RTC_CNTL0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC Control 0 Register (RTC_CNTL0)	0x300a02 (8 bits)	D7-5	–	reserved		–	–	0 when being read.	
		D4	RTC24H	24H/12H mode select	1 24H	0 12H	X (0)	R/W	
		D3	–	reserved		–	–	–	0 when being read.
		D2	RTCADJ	30-second adjustment	1 Adjust	0 –	X (0)	R/W	
		D1	RTCSTP	Divider run/stop control	1 Stop	0 Run	X (0)	R/W	
		D0	RTCSTP	Software reset	1 Reset	0 –	X (0)	R/W	

Init.: () indicates the value set after a software reset (RTCSTP → 1 → 0) is performed.

D[7:5] 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 (software reset value)

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) (software reset value)

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: Divider Run/Stop Control Bit

This bit starts or stops the divider. It also indicates divider operating status.

1 (W): Stops divider

0 (W): Starts divider

1 (R): Divider/counters are idle

0 (R): Divider/counters are operating (software reset value)

8 REAL-TIME CLOCK (RTC)

Setting this bit to 0 starts the divider; setting it to 1 stops the divider.

The value read from this bit is 0 when the divider/counters are operating, and 1 when the counters are idle.

This bit starts/stops the divider 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 divider stops while carry of a counter is taking place, the count value may be corrupted. Therefore, see Section 8.3.5 to ensure that carry is not taking place when the divider is stopped. This is not required when, for example, the contents of all counters are newly set again.

D0 **RTCRST: Software Reset Bit**

This bit resets the divider and output signals.

1 (R/W): Reset

0 (R/W): Negate reset (software reset value)

To perform software reset, write 1 to RTCRST and then write 0.

The software reset clears the 32 kHz to 2 Hz divider bits, negates the interrupt request and WAKEUP signals, and initializes some control bits.

When setting up the RTC, first perform software reset using RTCRST.

RTC Control 1 Register (RTC_CNTL1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Control 1 Register (RTC_CNTL1)	0x300a03 (8 bits)	D7-3	-	reserved	-	-	-	0 when being read.
		D2	RTCRDHL	Read buffer enable	1 Enable 0 Disable	X (0)	R/W	
		D1	RTCBSY	Counter busy flag	1 Busy 0 R/W possible	X (0)	R	
		D0	RTCHLD	Counter hold control	1 Hold 0 Running	X (0)	R/W	

Init.: () indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

D[7:3] **Reserved**

D2 **RTCRDHL: Read Buffer Enable Bit**

This bit enables or disables the read buffer.

1 (R/W): Enabled

0 (R/W): Disabled (software reset value)

In order to prevent carry over during reading counters, the RTC includes a read buffer to hold counter data. Before reading counter data, set RTCRDHL to 1 to load the current counter data to the read buffer. While RTCRDHL is set to 1, the buffered data is read out from the counter registers. Be sure to reset RTCRDHL to 0 after the buffered data is read out. This operation does not affect the counters. The counters keeps counting while RTCRDHL is set to 1.

D1 **RTCBSY: Counter Busy Flag Bit**

This flag indicates whether 1 is being carried over to the next-digit counter.

1 (R): Busy (while carry is taking place)

0 (R): Accessible for read/write (software reset value)

Attempting a write or stop operation may corrupt the counter values if 1 is being carried over. Therefore, this bit should be checked to confirm that the counters are not in a carry (busy) state before writing data to the counter registers.

When a value of 0 is read from RTCBSY after writing 1 to RTCHLD, it means that carry is not taking place. In this state, counter data can be written to.

After 1 is written to RTCHLD, the counters stop operating. So RTCBSY is fixed at 0, as carry will not take place. In this case, the counter hold function is also actuated, with a carry over of 1 to the 1-second counter disabled in hardware. The divider (counter for less than one second) continues operating.

Write data to the counter registers. After writing data, reset RTCHLD to 0.

If 1 is being carried over when data is being written to a counter in the hold state, 1 second is automatically added to correct the counter values when RTCHLD is reset to 0. This correction is only effective for 1 second and no correction is conducted on the carry encountered in the second time and on. In this case, the timekeeping data gets out of order. Therefore, be sure to reset RTCHLD to 0 as soon as possible after completing the required write operation.

When a value of 1 is read from RTCBSY after writing 1 to RTCHLD, it means that carry is taking place. In this case, writing 1 to RTCHLD is ignored and RTCHLD retains 0. A period of 4 ms per second is required for a carry over of 1 to the counters. In this case, repeat writing 1 to RTCHLD and checking RTCBSY, or write 1 to RTCHLD and check RTCBSY after waiting for 4 ms.

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 (software reset value)

For the operation of this bit, see the description of RTCBSY above.

RTC Second Register (RTC_SEC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Second Register (RTC_SEC)	0x300a04 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	RTCSH[2:0]	RTC 10-second counter	0 to 5	X (*)	R/W	
		D3–0	RTCSL[3:0]	RTC 1-second counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Note: Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)

D7 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.

RTC Minute Register (RTC_MIN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Minute Register (RTC_MIN)	0x300a05 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	RTCMIH[2:0]	RTC 10-minute counter	0 to 5	X (*)	R/W	
		D3–0	RTCMIL[3:0]	RTC 1-minute counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Note: Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)

D7 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.

RTC Hour Register (RTC_HOUR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Hour Register (RTC_HOUR)	0x300a06 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	RTCAP	AM/PM indicator	1 PM 0 AM	X (*)	R/W	
		D5–4	RTCHH[1:0]	RTC 10-hour counter	0 to 2 or 0 to 1	X (*)	R/W	
		D3–0	RTCHL[3:0]	RTC 1-hour counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

- Notes:**
- Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)
 - Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D7 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/RTC_CNTL0 register 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/RTC_CNTL0 register 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 o'clock or 24 o'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 o'clock or 24 o'clock.

RTC Day Register (RTC_DAY)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Day Register (RTC_DAY)	0x300a07 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	RTCDH[1:0]	RTC 10-day counter	0 to 3	X (*)	R/W	
		D3–0	RTCDL[3:0]	RTC 1-day counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

- Notes:**
- Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)
 - Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:6] 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.

RTC Month Register (RTC_MONTH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Month Register (RTC_MONTH)	0x300a08 (8 bits)	D7-5	—	reserved	—	—	—	0 when being read.
		D4	RTCMOH	RTC 10-month counter	0 to 1	X (*)	R/W	
		D3-0	RTCMOL[3:0]	RTC 1-month counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

- Notes:**
- Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)
 - Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:5] 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.

RTC Year Register (RTC_YEAR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Year Register (RTC_YEAR)	0x300a09 (8 bits)	D7-4	RTCYH[3:0]	RTC 10-year counter	0 to 9	X (*)	R/W	
		D3-0	RTCYL[3:0]	RTC 1-year counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

- Notes:**
- Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)
 - Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

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.

RTC Days of Week Register (RTC_WEEK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC Days of Week Register (RTC_WEEK)	0x300a0a (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2-0	RTCWK[2:0]	RTC days of week counter	RTCWK[2:0] Days of week	X (*)	R/W		
					0x7	–			
					0x6	Saturday			
					0x5	Friday			
					0x4	Thursday			
					0x3	Wednesday			
					0x2	Tuesday			
					0x1	Monday			
					0x0	Sunday			

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

- Notes:**
- Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)
 - Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:3] Reserved

D[2:0] RTCWK[2:0]: RTC Days of Week Counter Bits

These bits are the septenary counter (that counts from 0 to 6) bits 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 8.6.3 lists the basic correspondence.

Table 8.6.3 Correspondence between Counter Values and Days of the Week

RTCWK[2:0]	Days of the week
0x6	Saturday
0x5	Friday
0x4	Thursday
0x3	Wednesday
0x2	Tuesday
0x1	Monday
0x0	Sunday

(Default: indeterminate, software reset: previous value retained)

RTC Wakeup Configuration Register (RTC_WAKEUP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC Wakeup Configuration Register (RTC_WAKEUP)	0x300a0f (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.	
		D1	WUP_CTL	WAKEUP control	1 Active	0 Inactive	X (0)	R/W	
		D0	WUP_POL	WAKEUP polarity select	1 Active low	0 Active high	X (0)	R/W	

Init.: () indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

D[7:2] Reserved

D1 WUP_CTL: WAKEUP Control Bit

This bit controls the WAKEUP output.

1 (R/W): Active

0 (R/W): Inactive (software reset value)

This bit is used to control the WAKEUP output with software. The WAKEUP signal will also be asserted when a cause of RTC interrupt occurs.

D0 WUP_POL: WAKEUP Polarity Select Bit

This bit selects the active level of the WAKEUP output signal.

1 (R/W): Active low

0 (R/W): Active high (software reset value)

9 SRAM Controller (SRAMC)

The S1C33L26 includes a bus controller that controls access to external memories. The bus controller consists of an SRAM controller (SRAMC) for controlling the SRAM, an SDRAM controller (SDRAMC) for controlling the SDRAM, and a data queue buffer (DQB) for efficiently reading from external memories.

The following describes the SRAMC. For information on the SDRAMC and DQB, see the “SDRAM Controller (SDRAMC)” chapter.

9.1 SRAMC Module Overview

The SRAM controller (SRAMC) is a module for controlling the external bus. It can output up to six chip enable signals and configure the access cycle and the connected device type for respective areas assigned to each chip enable signal. When the CPU or DMAC accesses an external memory space, the SRAMC reads/writes from/to the connected memory or I/O device according to the defined access conditions.

The features of the SRAMC are described below.

- 26-bit address bus (maximum 64M-byte address space)
- 8-bit or 16-bit selectable data bus
- Up to six chip enable signals are available for connecting external devices.
- Supports connections to a Flash ROM, SRAM, and other devices such as an LCD driver.
- Programmable bus access wait cycle (0 to 15 cycles)
- Supports little endian access.
- Supports memory mapped I/O devices.
- Supports either A0 or BS (bus strobe) access type.
- Supports external wait requests via the #WAIT pin.

9.2 SRAMC Pins

Table 9.2.1 lists the pins used by the SRAMC.

Table 9.2.1 SRAMC Pin List

Pin name	I/O	Qty	Function
D[15:0]	I/O	16	External data bus D[15:0]
A0/#BSL	O	1	External address bus A0 / Bus strobe (low byte) signal output
A[25:1]	O	25	External address bus A[25:1] Note that the A[25:24] pins do not exist in the TQFP15-128pin model.
#CE10, #CE9, #CE8, #CE7, #CE5, #CE4	O	6	Chip enable signal outputs
#RD	O	1	Read signal output
#WRL	O	1	Write (low byte) signal output
#WRH/#BSH	O	1	Write (high byte) signal / Bus strobe (high byte) signal output
#WAIT	I	1	External wait request signal input

Notes:

- Some of the bus control pins listed above are shared with general-purpose I/O ports and they may be configured for I/O ports at initial reset. 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 bit.

For details on how to switch over the pin functions, see the “I/O Ports (GPIO)” chapter.

- The bus control signals can be pulled up or forcibly driven low via software. For more information, see the “I/O Ports (GPIO)” chapter.

9.3 SRAMC Operating Clock

The SRAMC operates with BCLK supplied from the CMU. BCLK does not stop in normal mode and in HALT mode by default. It can be stopped in HALT mode using a CMU control register. BCLK can also be output to external devices from the CMU_CLK pin. For more information on BCLK, see the “Clock Management Unit (CMU)” chapter. In SLEEP mode, the CMU stops supplying BCLK to the SRAMC.

PCLK2 and SDCLK are also used for accessing the SRAMC control registers and the area for SDRAM, respectively.

9.4 External Memory Areas

The SRAMC supports an external memory space, which is divided into 14 areas as shown in Figure 9.4.1.

			#CE4	#CE5	#CE7	#CE8	#CE9	#CE10
Area 22	0xffff ffff 0x8000 0000	External memory 2G (64M) bytes (*2)					●	
Area 21	0x7fff ffff 0x4000 0000	External memory 1G (64M) bytes (*2)				●		
Area 20	0x3fff ffff 0x2000 0000	External memory 512M (64M) bytes (*2)						●
Area 19	0x1fff ffff	External memory						
(*1)	0x1000 0000	256M (64M) bytes (*2)			●			
Area 18	0x0fff ffff 0x0c00 0000	Reserved						
Area 17	0x0bfff ffff 0x0800 0000	Reserved						
Area 16	0x07ff ffff 0x0600 0000	External memory 32M bytes		●				
Area 15	0x05ff ffff 0x0400 0000	External memory 32M bytes		●				
Area 14	0x03ff ffff 0x0300 0000	External memory 16M bytes	●					
Area 13	0x02ff ffff 0x0200 0000	External memory 16M bytes						●
Area 12	0x01ff ffff 0x0180 0000	Reserved						
Area 11	0x017f ffff 0x0100 0000	Reserved						
Area 10	0x00ff ffff 0x00c0 0000	External memory 4M bytes						●
Area 9	0x00bf ffff 0x0080 0000	External memory 4M bytes					●	
Area 8	0x007f ffff 0x0060 0000	External memory 2M bytes				●		
Area 7	0x005f ffff (*1) 0x0040 0000	External memory 2M bytes			●			
Area 6	0x003f ffff 0x0030 0000	(Reserved area for internal peripheral modules)						
Area 5	0x002f ffff 0x0020 0000	External memory 1M bytes		●				
Area 4	0x001f ffff 0x0010 0000	External memory 1M bytes	●					

*1 Usable as an SDRAM area.

*2 Since the address bus is 26-bit wide, valid area for each area is 64M bytes from the top.

Figure 9.4.1 S1C33L26 External Memory Space

Areas 4, 5, 7 to 10, 13 to 16, and 19 to 22 comprise an external memory area accessible from the SRAMC, to which external memory devices may be connected. The access conditions can be set by area, including the device type and size as well as the number of wait cycles.

9.4.1 Chip Enable Signals

The SRAMC provides maximum 26 bits of an external address bus, 16 bits of an external data bus, and 6 chip-enable pins (#CE4, #CE5, #CE7 to #CE10), allowing access to a maximum 336M-byte address space.

Two or more areas are assigned to each chip-enable signal. Table 9.4.1.1 shows the relationship between the chip-enable pins and corresponding areas.

Table 9.4.1.1 Chip Enable Pins and Corresponding Areas

#CE pin	Corresponding area	Available area capacity for a series of adjacent addresses					
		Area	Capacity	Area	Capacity	Area	Capacity
#CE4	Areas 4 and 14	Area 4	1MB	Area 14	16MB	–	–
#CE5	Areas 5, 15, and 16	Area 5	1MB	Areas 15 + 16	64MB	–	–
#CE7	Areas 7 and 19	Area 7	2MB	Area 19	64MB	–	–
#CE8	Areas 8 and 21	Area 8	2MB	Area 21	64MB	–	–
#CE9	Areas 9 and 22	Area 9	4MB	Area 22	64MB	–	–
#CE10	Areas 10, 13, and 20	Area 10	4MB	Area 13	16MB	Area 20	64MB

The #CE_x signal becomes active when an address in any corresponding area is accessed.

9.4.2 Area Condition Settings

Bus access conditions can be set by area for each #CE_x signal. Therefore, the same conditions are set for two or more areas accommodated by the respective #CE_x signals.

This section describes the parameters to be set individually for each #CE_x area and the relevant control bits.

The SRAMC control registers are initialized by an initial reset. These registers should be set up in software to suit the external device configuration or specification as required.

Note: Letter 'X' in the control bit and #CE names denotes a #CE area number (4, 5, or 7 to 10).

Endian mode

The S1C33L26 supports little endian mode only.

Device type

The device size can be selected from 8 bits and 16 bits. Additionally, for a 16-bit device, the device type can also be selected from the A0 (default) or BSL modes.

For selection, use CE_xTYPE[1:0]/SRAMC_TYPE register.

Table 9.4.2.1 Device Type Selections

CE _x TYPE[1:0]	Device type	Pins to be used
0x3–0x2	8-bit device	A[25:0], D[7:0], #CE _x , #RD, #WRL
0x1	16-bit BSL device	A[25:1], D[15:0], #CE _x , #RD, #WRL, #BSL(A0), #BSH
0x0	16-bit A0 device	A[25:1], D[15:0], #CE _x , #RD, #WRL, #WRH

(Default: 0x0)

#CE setup time

The setup time for #CE_x signals (from a #CE_x falling edge to the read/write signal falling edge) can be set to within the range from 1 to 4 cycles. Use CE_xSETUP[1:0] in the SRAMC_TMG47 and SRAMC_TMG810 registers for settings.

Table 9.4.2.2 #CE Setup Time Settings

CE _x SETUP[1:0]	Setup time
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x3)

#CE hold time

The hold time for #CE_x signals (from a read/write signal rising edge to the #CE_x rising edge) can be set to within the range from 1 to 4 cycles. Use CExHOLD[1:0] in the SRAMC_TMG47 and SRAMC_TMG810 registers for settings.

Table 9.4.2.3 #CE Hold Time Settings

CExHOLD[1:0]	Hold time
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x3)

Static wait cycles

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. According to the specifications of the connected device, set an appropriate wait cycle using CExWAIT[3:0] in the SRAMC_TMG47 and SRAMC_TMG810 registers. If CExWAIT[3:0] is set to 0, no static wait cycle is inserted. In this case, the minimum read/write pulse width will be one cycle.

Table 9.4.2.4 Static Wait Cycle Settings

CExWAIT[3:0]	Static wait cycle	Read/write cycle
0xf	15 cycles	16 cycles (+ #WAIT)
0xe	14 cycles	15 cycles (+ #WAIT)
:	:	:
0x1	1 cycle	2 cycles (+ #WAIT)
0x0	0 cycles	1 cycle (+ #WAIT)

(Default: 0xf)

The area to which an SRAM device is connected allows dynamic wait control using the #WAIT pin in addition to the static wait control.

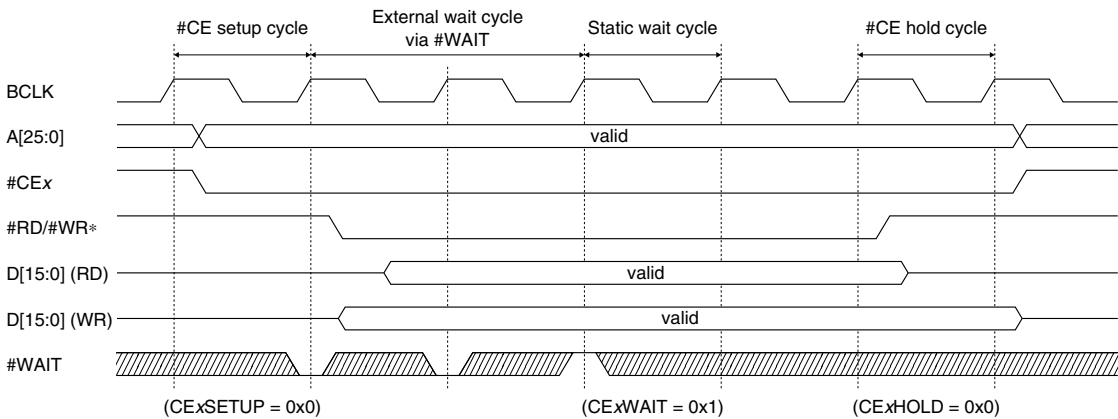


Figure 9.4.2.1 Example of Timing Parameter Settings

9.5 Connection of External Devices and Bus Operation

9.5.1 Connecting External Devices

The following shows examples of connecting the S1C33L26 and SRAM.

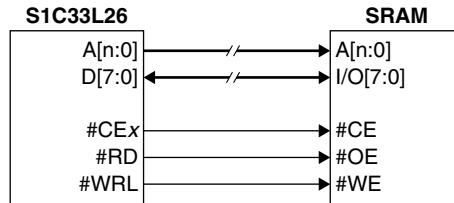


Figure 9.5.1.1 Example of 8-bit SRAM Connection

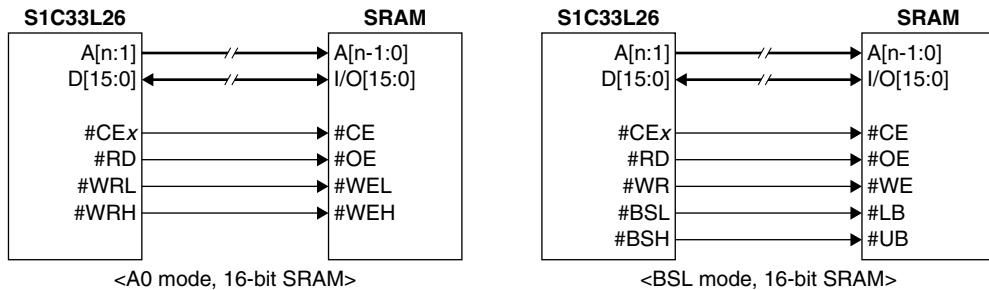


Figure 9.5.1.2 Example of 16-bit SRAM Connection

9.5.2 Data Configuration in Memory

The S1C33L26 SRAMC handles byte (8-bit), halfword (16-bit), and word (internal 32-bit) data. To access data in a 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 contents 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 9.5.2.1 shows where each type of data is located in a memory.

Table 9.5.2.1 Data Locations in Memory

Data type	Location
Byte	Byte boundary (all addresses)
Halfword	Halfword boundary (A0 = 0)
Word	Word boundary (A[1:0] = 0b00)

All halfword and word data in a memory are accessed in little endian mode. To increase memory efficiency, try locating the same type of data at continuous addresses to reduce blank areas created by positioning at boundary addresses as much as possible.

9.5.3 External Bus Operation

The internal data bus size in the S1C33L26 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 9.5.3.1 shows bus operations in A0 and BSL modes.

For details on how to connect memory, see Section 9.5.1, “Connecting External Devices.”

Table 9.5.3.1 Bus Operations

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
	Halfword	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
	Halfword	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

9.6 Bus Access Timing Charts

9.6.1 SRAM Read/Write Timing 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

#CE setup/hold time: 1 cycle

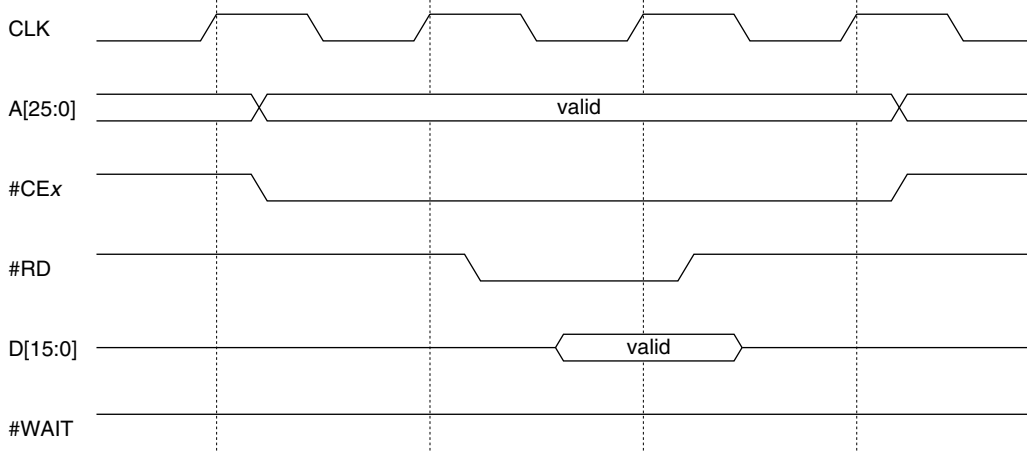


Figure 9.6.1.1 SRAM Read Timing with No Static Wait Cycle

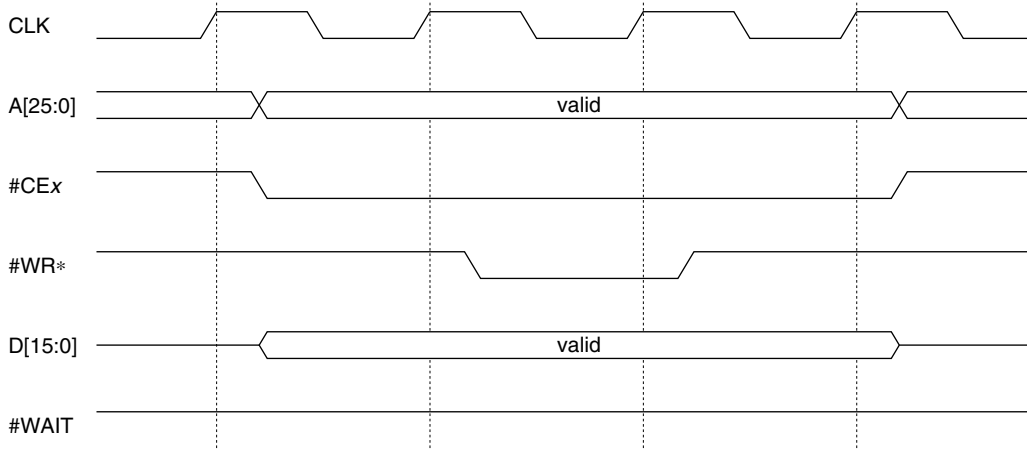


Figure 9.6.1.2 SRAM Write Timing with No Static Wait Cycle

2. SRAM read/write timings with static wait cycles

[Example settings]

- Device size: 16 bits
- Number of static wait cycles: 2 cycles
- #CE setup/hold time: 1 cycle

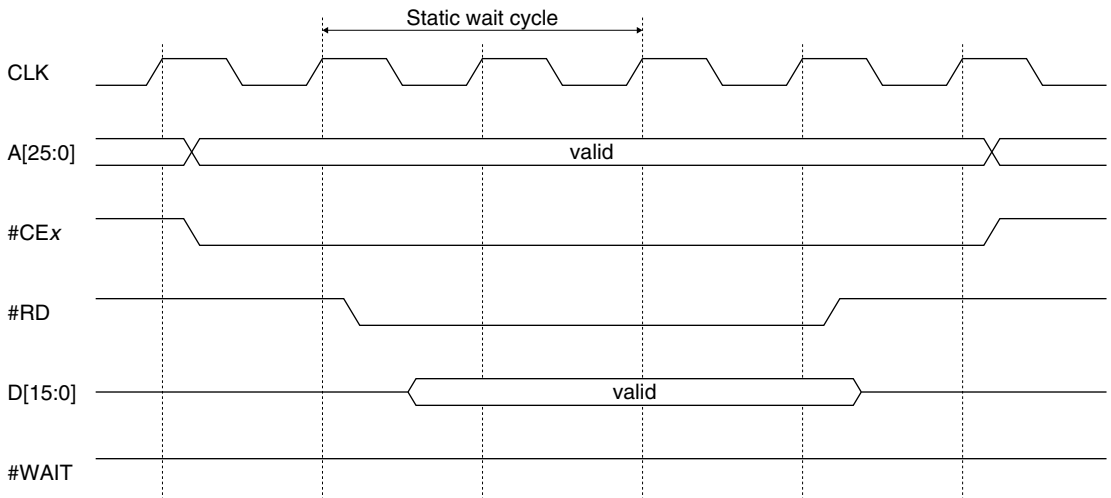


Figure 9.6.1.3 SRAM Read Timing with Static Wait Cycle

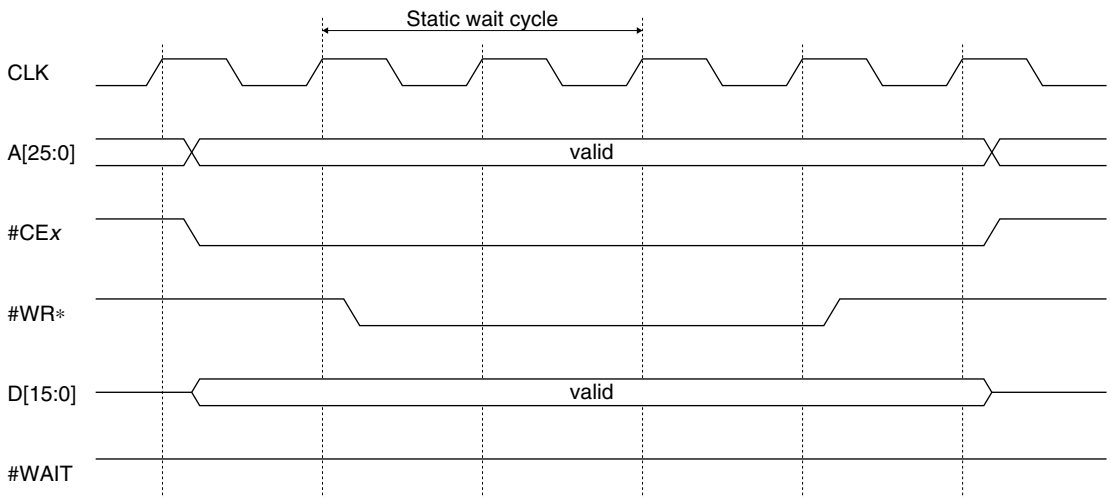


Figure 9.6.1.4 SRAM Write Timing with Static Wait Cycle

9.6.2 SRAM Read/Write Timings with External #WAIT

Wait cycles can be inserted from the #WAIT pin only for SRAM-type devices.

The external #WAIT signal is sampled on the rising edges of BCLK after the #CE setup cycles end and no later than 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 (see Note below)

#CE setup/hold time: 1 cycle (see Note below)

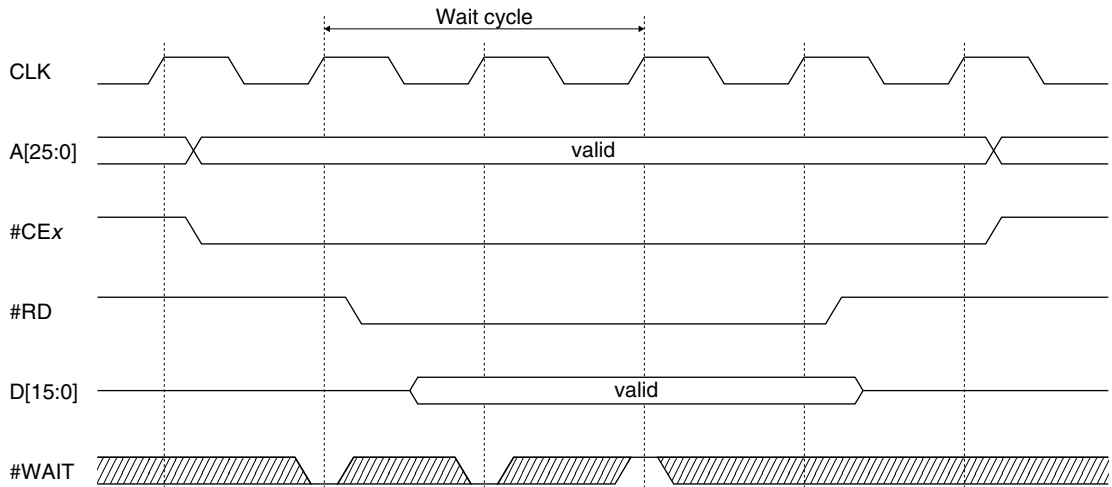


Figure 9.6.2.1 SRAM Read Timing with External #WAIT

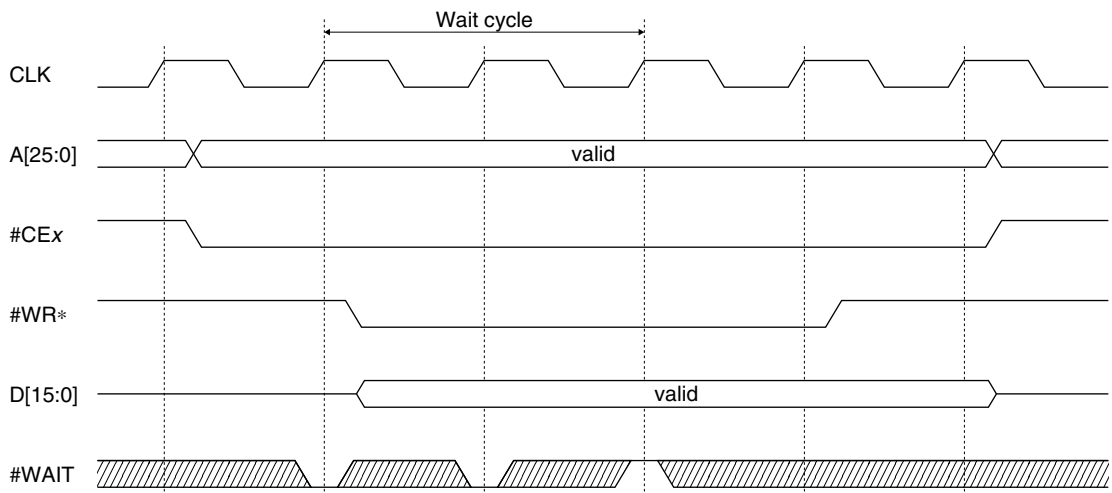


Figure 9.6.2.2 SRAM Write Timing with External #WAIT

9 SRAM CONTROLLER (SRAMC)

Note: Figures 9.6.2.1 and 9.6.2.2 assume a very low operating speed for convenience of explanation. When actually using an external wait request, to allow for a delay due to noise filter of the #WAIT pin, be sure to set the #CE setup cycles (CE_{SETUP}[1:0]) or static wait cycles (CE_{WAIT}[3:0]) as follows:

- When using #WAIT, set CE_{SETUP}[1:0] to 0x1 or higher, or set CE_{WAIT}[3:0] to 0x1 or higher.
- When a #CE_{EX} signal is used to generate a #WAIT signal, set the conditions no lower than the following.

CE_{SETUP}[1:0] = 0x1 and CE_{WAIT}[3:0] = 0x2, or
 CE_{SETUP}[1:0] = 0x2 and CE_{WAIT}[3:0] = 0x1, or
 CE_{SETUP}[1:0] = 0x3 and CE_{WAIT}[3:0] = 0x0

- When a #RD/#WRH/#WRL signal is used to generate a #WAIT signal, set the condition no lower than the following.

CE_{WAIT}[3:0] = 0x3

When settings are other than the listed above, external wait is ineffective.

9.7 Control Register Details

Table 9.7.1 List of SRAMC Registers

Address	Register name	Function
0x302220	SRAMC_TMG47	#CE[7:4] Access Timing Configuration Register
0x302224	SRAMC_TMG810	#CE[10:8] Access Timing Configuration Register
0x302228	SRAMC_TYPE	#CE[10:4] Device Configuration Register

The following describes each SRAMC register. These are all 32-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

#CE[7:4] Access Timing Configuration Register (SRAMC_TMG47)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
#CE[7:4] Access Timing Configuration Register (SRAMC_TMG47)	0x302220 (32 bits)	D31–30	CE7SETUP [1:0]	#CE7 setup cycle	CE7SETUP[1:0]	Setup cycle	0x3	R/W		
					0x3	4 cycles				
					0x2	3 cycles				
					0x1	2 cycles				
							0x0	1 cycle		
		D29–28	CE7HOLD [1:0]	#CE7 hold cycle	CE7HOLD[1:0]	Hold cycle	0x3	R/W		
					0x3	4 cycles				
					0x2	3 cycles				
					0x1	2 cycles				
							0x0	1 cycle		
		D27–24	CE7WAIT [3:0]	#CE7 static wait cycle	CE7WAIT[3:0]	Wait cycle	0xf	R/W		
					0xf	15 cycles				
0xe	14 cycles									
:	:									
					0x1	1 cycle				
					0x0	0 cycles				
D23–16	–	–	reserved	–	–	–	–	1 when being read.		
D15–14	CE5SETUP [1:0]	#CE5 setup cycle	CE5SETUP[1:0]	Setup cycle	0x3	R/W				
			0x3	4 cycles						
			:	:						
			0x0	1 cycle						
D13–12	CE5HOLD [1:0]	#CE5 hold cycle	CE5HOLD[1:0]	Hold cycle	0x3	R/W				
			0x3	4 cycles						
			:	:						
			0x0	1 cycle						
D11–8	CE5WAIT [3:0]	#CE5 static wait cycle	CE5WAIT[3:0]	Wait cycle	0xf	R/W				
			0xf	15 cycles						
			:	:						
			0x0	0 cycles						
D7–6	CE4SETUP [1:0]	#CE4 setup cycle	CE4SETUP[1:0]	Setup cycle	0x3	R/W				
			0x3	4 cycles						
			:	:						
			0x0	1 cycle						

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
#CE[7:4] Access Timing Configuration Register (SRAMC_ TMG47)	0x302220 (32 bits)	D5–4	CE4HOLD [1:0]	#CE4 hold cycle	CE4HOLD[1:0]	Hold cycle	0x3	R/W
					0x3	4 cycles		
					:	:		
					0x0	1 cycle		
		D3–0	CE4WAIT [3:0]	#CE4 static wait cycle	CE4WAIT[3:0]	Wait cycle	0xf	R/W
					0xf	15 cycles		
					:	:		
					0x0	0 cycles		

See the descriptions of the SRAMC_TMG810 register.

#CE[10:8]Access Timing Configuration Register (SRAMC_TMG810)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
#CE[10:8] Access Timing Configuration Register (SRAMC_ TMG810)	0x302224 (32 bits)	D31–24	–	reserved	–	–	–	–	1 when being read.
		D23–22	CE10SETUP [1:0]	#CE10 setup cycle	CE10SETUP[1:0]	Setup cycle	0x3	R/W	
					0x3	4 cycles			
					0x2	3 cycles			
					0x1	2 cycles			
					0x0	1 cycle			
		D21–20	CE10HOLD [1:0]	#CE10 hold cycle	CE10HOLD[1:0]	Hold cycle	0x3	R/W	
					0x3	4 cycles			
					0x2	3 cycles			
					0x1	2 cycles			
					0x0	1 cycle			
		D19–16	CE10WAIT [3:0]	#CE10 static wait cycle	CE10WAIT[3:0]	Wait cycle	0xf	R/W	
					0xf	15 cycles			
					0xe	14 cycles			
			:	:					
			0x1	1 cycle					
			0x0	0 cycles					
D15–14	CE9SETUP [1:0]	#CE9 setup cycle	CE9SETUP[1:0]	Setup cycle	0x3	R/W			
			0x3	4 cycles					
			:	:					
			0x0	1 cycle					
D13–12	CE9HOLD [1:0]	#CE9 hold cycle	CE9HOLD[1:0]	Hold cycle	0x3	R/W			
			0x3	4 cycles					
			:	:					
			0x0	1 cycle					
D11–8	CE9WAIT [3:0]	#CE9 static wait cycle	CE9WAIT[3:0]	Wait cycle	0xf	R/W			
			0xf	15 cycles					
			:	:					
			0x0	0 cycles					
D7–6	CE8SETUP [1:0]	#CE8 setup cycle	CE8SETUP[1:0]	Setup cycle	0x3	R/W			
			0x3	4 cycles					
			:	:					
			0x0	1 cycle					
D5–4	CE8HOLD [1:0]	#CE8 hold cycle	CE8HOLD[1:0]	Hold cycle	0x3	R/W			
			0x3	4 cycles					
			:	:					
			0x0	1 cycle					
D3–0	CE8WAIT [3:0]	#CE8 static wait cycle	CE8WAIT[3:0]	Wait cycle	0xf	R/W			
			0xf	15 cycles					
			:	:					
			0x0	0 cycles					

The SRAMC_TMG47 and the SRAMC_TMG810 registers are used to set the SRAM access timing for each #CE area.

Letter ‘x’ in the control bit and #CE names denotes a #CE area number (4, 5, or 7 to 10).

D[31:30], D[23:22], D[15:14], D[7:6]

CE_xSETUP[1:0]: #CE_x Setup Cycle Bits

Configures the #CE_x signal setup time (#CE_x falling edge to read/write signal falling edge).

Table 9.7.2 #CE Setup Time Settings

CE _x SETUP[1:0]	Setup time
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x3)

D[29:28], D[21:20], D[13:12], D[5:4]

CExHOLD[1:0]: #CEx Hold Cycle Bits

Configures the #CEx signal hold time (read/write signal rising edge to #CEx rising edge).

Table 9.7.3 #CE Hold Time Settings

CExHOLD[1:0]	Hold time
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x3)

D[27:24], D[19:16], D[11:8], D[3:0]

CExWAIT[3:0]: #CEx Static Wait Cycle Bits

Configures the #CEx signal 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. If CExWAIT[3:0] is set to 0, no static wait cycle is inserted. In this case, the minimum read/write pulse width will be one cycle.

Table 9.7.4 Static Wait Cycle Settings

CExWAIT[3:0]	Static wait cycle	Read/write cycle
0xf	15 cycles	16 cycles (+ #WAIT)
0xe	14 cycles	15 cycles (+ #WAIT)
:	:	:
0x1	1 cycle	2 cycles (+ #WAIT)
0x0	0 cycles	1 cycle (+ #WAIT)

(Default: 0xf)

The area to which an SRAM device is connected allows dynamic wait control using the #WAIT pin in addition to the static wait control.

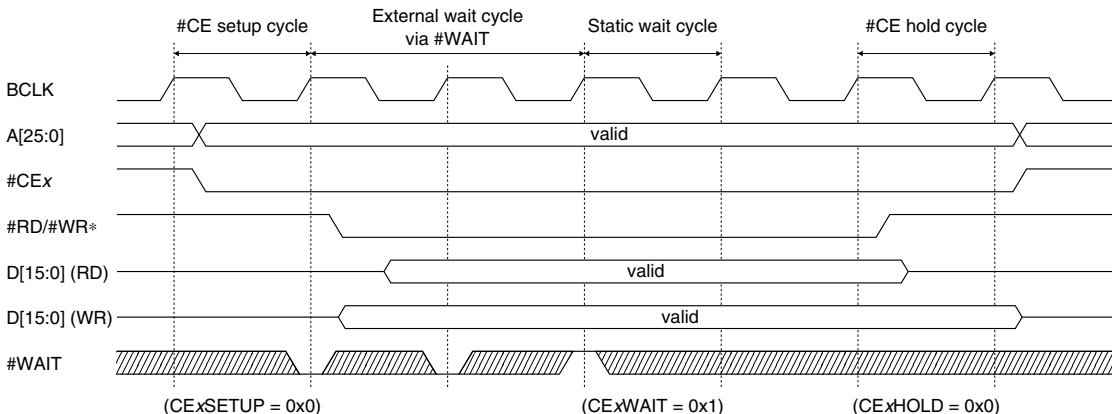


Figure 9.7.1 Example of Timing Parameter Settings

#CE[10:4] Device Configuration Register (SRAMC_TYPE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
#CE[10:4] Device Configuration Register (SRAMC_TYPE)	0x302228 (32 bits)	D31–14	–	reserved	–	–	–	0 when being read.	
		D13–12	CE10TYPE [1:0]	#CE10 device type	CExTYPE[1:0] 0x3–0x2 0x1 0x0	Device type 8-bit device 16-bit BSL type 16-bit A0 type	0x0	R/W	
		D11–10	CE9TYPE [1:0]	#CE9 device type			0x0	R/W	
		D9–8	CE8TYPE [1:0]	#CE8 device type			0x0	R/W	
		D7–6	CE7TYPE [1:0]	#CE7 device type			0x0	R/W	
		D5–4	–	reserved	–	–	–	0 when being read.	
		D3–2	CE5TYPE [1:0]	#CE5 device type	CExTYPE[1:0] 0x3–0x2 0x1 0x0	Device type 8-bit device 16-bit BSL type 16-bit A0 type	0x0	R/W	
		D1–0	CE4TYPE [1:0]	#CE4 device type			0x0	R/W	

D[31:14], D[5:4]**Reserved****D[13:12], D[11:10], D[9:8], D[7:6], D[3:2], D[1:0]****CExTYPE[1:0]: #CE_x Device Type Bits**

Selects the device type for each #CE area.

Table 9.7.5 Device Type Selections

CExTYPE[1:0]	Device type	Pins to be used
0x3–0x2	8-bit device	A[25:0], D[7:0], #CE _x , #RD, #WRL
0x1	16-bit BSL device	A[25:1], D[15:0], #CE _x , #RD, #WRL, #BSL(A0), #BSH
0x0	16-bit A0 device	A[25:1], D[15:0], #CE _x , #RD, #WRL, #WRH

(Default: 0x0)

10 SDRAM Controller (SDRAMC)

The S1C33L26 includes a bus controller that controls access to external memories. The bus controller consists of an SRAM controller (SRAMC) for controlling the SRAM, an SDRAM controller (SDRAMC) for controlling the SDRAM, and a data queue buffer (DQB) for efficiently reading from external memories.

The following describes the SDRAMC and DQB. For information on the SRAMC, see the “SRAM Controller (SRAMC)” chapter.

10.1 SDRAMC Module Overview

The SDRAM controller (SDRAMC) allows up to 64MB of SDRAM to be connected directly to Areas 7 and 19. The main features of the SDRAMC are outlined below.

- Supports direct connection of an SDRAM from minimum 16M bits (2MB) to maximum 512M bits (64MB).
- The operating clock frequency: Maximum 72 MHz
(can be set at the same or double of the CPU clock frequency)
- Data bus width: 16 bits (16-bit SDRAM × 1 or 8-bit SDRAM × 2)
- CAS latency: Can be set to 1, 2 or 3.
- Supports burst transfers (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.
- Includes a programmable 12-bit auto refresh counter.
Necessary refreshing enabled irrespective of the clock frequency used.
- Provided with intelligent self-refresh mode for low-power operation.
- Supports the EMRS (Extended Mode Register Set) command to program drive strength, temperature compensated self refresh, and partial array self refresh, in addition to MRS (Mode Register Set).

10.2 SDRAMC Pins

Table 10.2.1 lists the pins used by the SDRAMC.

Table 10.2.1 SDRAMC Pin List

Pin name	I/O	Qty	Function
A[15:14]/SDBA[1:0]	O	2	Bank select signal output
A[13:1]/SDA[12:0]	O	13	Address signal output
D[15:0]	I/O	16	Data signal I/O
SDCKE	O	1	SDRAM clock enable signal output
SDCLK	O	1	SDRAM clock output
#SDCS	O	1	SDRAM chip enable signal output
A20/#SDRAS	O	1	SDRAM row address strobe signal output
A19/#SDCAS	O	1	SDRAM column address strobe signal output
A18/#SDWE	O	1	SDRAM write signal output
A16/DQML	O	1	SDRAM data (to select low-order byte) input/output mask signal output
A17/DQMH	O	1	SDRAM data (to select high-order byte) input/output mask signal output

- Notes:**
- Some of the bus control pins listed above are shared with general-purpose I/O ports and they may be configured for I/O ports at initial reset. 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 bit.
For details on how to switch over the pin functions, see the “I/O Ports (GPIO)” chapter.
 - The bus control signals can be pulled up or forcibly driven low via software. For more information, see the “I/O Ports (GPIO)” chapter.

10.3 SDRAM Clock and Double Frequency Mode

The SDCLK clock is supplied to the SDRAMC from the CMU for use as the operating clock of the SDRAMC module as well as the SDRAM clock.

The CMU register can be used to control the clock supply (on/off). When no SDRAM is used, stop the clock.

SDCLK does not stop even in HALT mode. To stop the clock in HALT mode, turn the clock supply off in the CMU before executing the halt instruction.

SDCLK stops in SLEEP mode.

PCLK2 is also used for accessing the SDRAMC control registers.

For details on how to set and control the clock, refer to the “Clock Management Unit (CMU)” chapter.

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 30 MHz clock, the SDRAM can be operated with a 60 MHz clock.

To set the SDRAMC in double frequency mode:

- (1) Set MCLK (main system clock) to the SYSCLK (= SDCLK) frequency $\times 1/2$.
- (2) Set DBF/SDRAMC_APP register to 1.

SDCLK frequency limitations

The SDCLK clock frequency is limited to maximum 72 MHz.

Normal mode: $SDCLK = MCLK \leq 60 \text{ MHz}$

Double frequency mode: $SDCLK = 2MCLK \leq 72 \text{ MHz}$, $MCLK \leq 36 \text{ MHz}$

10.4 Configuration of SDRAM

10.4.1 SDRAM Area

A #CE7 area (Area 7 or Area 19) is reserved for the SDRAMC. Note that the #CE7 area is configured for an SRAM area controlled with the SRAMC as the SDRAMC is disabled at initial reset. Therefore, to use the SDRAM, the #CE7 area must be configured for the SDRAM area by setting SDON/SDRAMC_INIT register to 1.

Note: Setting SDON to 1 overrides the external SRAM access conditions for the #CE7 area set in the SRAMC.

10.4.2 SDRAM Size and Access Condition Settings

Table 10.4.2.1 lists the conditions related to SDRAM size and timing parameters that the SDRAMC can accommodate.

Table 10.4.2.1 SDRAM Setup Items

Item	Setting contents	Control bit settings
SDRAM address configuration	32M × 16 bits × 1	ADDRC[2:0]/SDRAMC_CFG register = 0x7
	16M × 16 bits × 1	ADDRC[2:0]/SDRAMC_CFG register = 0x3
	8M × 16 bits × 1	ADDRC[2:0]/SDRAMC_CFG register = 0x2
	4M × 16 bits × 1	ADDRC[2:0]/SDRAMC_CFG register = 0x1
	1M × 16 bits × 1 (default)	ADDRC[2:0]/SDRAMC_CFG register = 0x0 (default)
	16M × 8 bits × 2	ADDRC[2:0]/SDRAMC_CFG register = 0x6
	8M × 8 bits × 2	ADDRC[2:0]/SDRAMC_CFG register = 0x5
CAS latency	3, 2, or 1	CAS[1:0]/SDRAMC_APP register = 0x3, 0x2 or 0x1
Burst length	2 (fixed)	–
tRP, tRCD	1 (default) to 4 cycles	T24NS[1:0]/SDRAMC_CFG register = 0x0 (default) to 0x3
tRAS	1 (default) to 8 cycles	T60NS[2:0]/SDRAMC_CFG register = 0x0 (default) to 0x7
tRC, tRFC, tXSR	1 to 16 cycles (default: 15 cycles)	T80NS[3:0]/SDRAMC_CFG register = 0x0 to 0xf (default: 0xe)

SDRAM address configuration

Use ADDR[2:0]/SDRAMC_CFG register to select the SDRAM size and chip configuration. This selection also sets up the bank size, column address size (page size), and row address size.

Table 10.4.2.2 SDRAM Size Selections and SDRAM Address

ADDR[2:0]	0x0 (default)	0x1	0x2	0x3	0x7	0x4	0x5	0x6
SDRAM device	16-bit device					–	Two 8-bit devices	
Capacity (M bit)	16	64	128	256	512	–	64 × 2	128 × 2
Data width	16 bits							
Row size	2048	4096	4096	8192	8192	–	4096	4096
Column size	256	256	512	512	1024	–	512	1024
Number of banks	2	4	4	4	4	–	4	4
A25	–	–	–	–	R10	–	–	–
A24	–	–	–	R12	R12	–	–	R11
A23	–	–	R11	R11	R11	–	R11	R10
A22	–	R11	R9	R9	R9	–	R9	R9
A21	–	R8	R8	R8	R8	–	R8	R8
A20	R7	R7	R7	R7	R7	–	R7	R7
A19	R6	R6	R6	R6	R6	–	R6	R6
A18	R5	R5	R5	R5	R5	–	R5	R5
A17	R4	R4	R4	R4	R4	–	R4	R4
A16	R3	R3	R3	R3	R3	–	R3	R3
A15	R2	R2	R2	R2	R2	–	R2	R2
A14	R1	R1	R1	R1	R1	–	R1	R1
A13	R0	R0	R0	R0	R0	–	R0	R0
A12	R10	R10	R10	R10	B1	–	R10	B1
A11	R9	R9	B0	B0	B0	–	B0	B0
A10	R8	B1	B1	B1	C9	–	B1	C9
A9	B0	B0	C8	C8	C8	–	C8	C8
A8	C7	C7	C7	C7	C7	–	C7	C7
A7	C6	C6	C6	C6	C6	–	C6	C6
A6	C5	C5	C5	C5	C5	–	C5	C5
A5	C4	C4	C4	C4	C4	–	C4	C4
A4	C3	C3	C3	C3	C3	–	C3	C3
A3	C2	C2	C2	C2	C2	–	C2	C2
A2	C1	C1	C1	C1	C1	–	C1	C1
A1	C0	C0	C0	C0	C0	–	C0	C0
A0	DQM	DQM	DQM	DQM	DQM	–	DQM	DQM

When reading/writing byte data, the SDRAMC decodes A0/#BSL and #WRH/#BSH into DQML and DQMH. Upper address bits that are not used (depending on memory size) are all fixed to 0.

10 SDRAM CONTROLLER (SDRAMC)

Figures 10.4.2.1 and 10.4.2.2 show examples of how to connect SDRAMs.

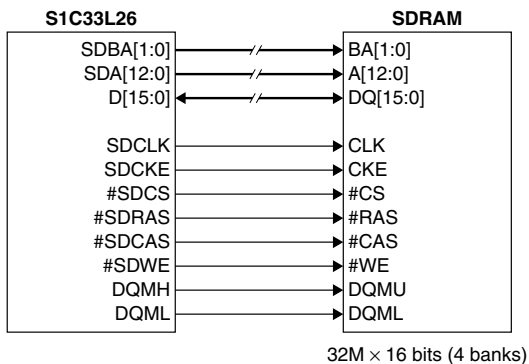


Figure 10.4.2.1 64MB SDRAM Connection Example

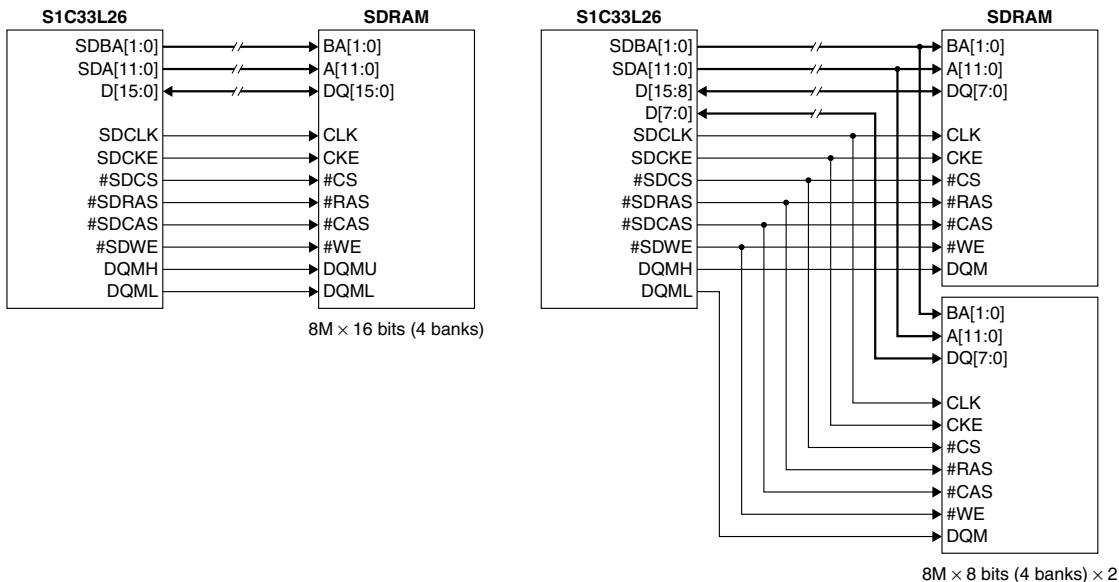


Figure 10.4.2.2 16MB SDRAM Connection Example

Timing setup

The following parameters can be configured according to the SDRAM specifications.

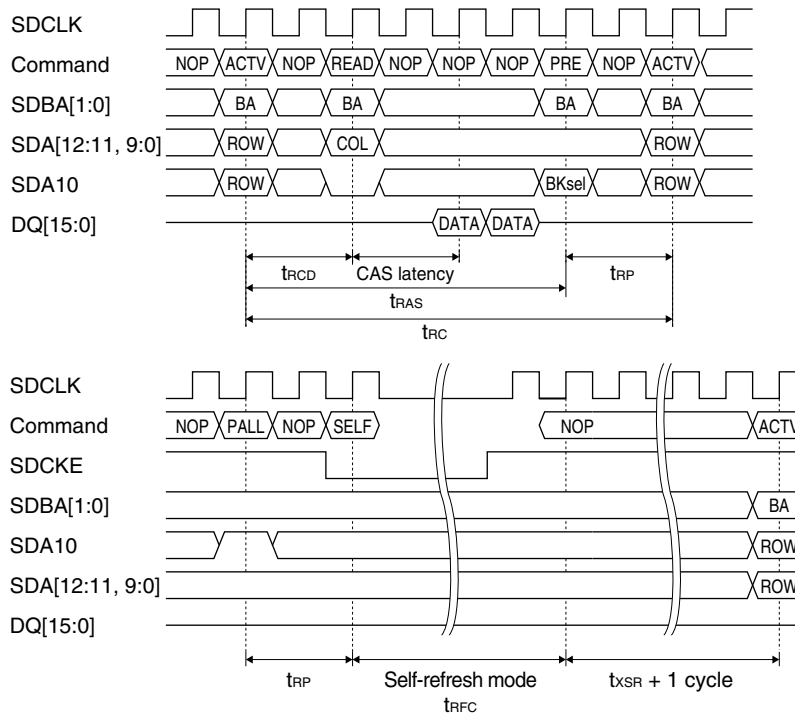


Figure 10.4.2.3 SDRAM Timing Parameters

(1) CAS Latency

CAS latency refers to the number of SDCLK clock cycles that run until data is output from the SDRAM after the READ command is issued. In this SDRAM interface, CAS latency can be set from 1 to 3 using CAS[1:0]/SDRAMC_APP register.

Table 10.4.2.3 CAS Latency Settings

CAS[1:0]	CAS latency
0x3	3
0x2	2
0x1	1
0x0	Reserved

(Default: 0x2)

(2) trC, trFC, txSR

trC: ACTIVE to ACTIVE command cycle time

trFC: Auto-refresh cycle time

txSR: Self-refresh end to ACTIVE command period

These timing parameters can be set from 1 to 16 cycles in SDCLK using T80NS[3:0]/SDRAMC_CFG register.

Table 10.4.2.4 t_{RC} , t_{RFC} , and t_{XSR} Settings

T80NS[3:0]	t_{RC} , t_{RFC} , t_{XSR}
0xf	16 cycles
0xe	15 cycles
0xd	14 cycles
0xc *	13 cycles
0xb	12 cycles
0xa	11 cycles
0x9	10 cycles
0x8 *	9 cycles
0x7	8 cycles
0x6	7 cycles
0x5	6 cycles
0x4 *	5 cycles
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0 *	1 cycle

(Default: 0xe)

* Recommended settings (For more information, see “(2) External SRAM access rate while the SDRAM is in self-refresh status” in Section 3.8.)

(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]/SDRAMC_CFG register.

Table 10.4.2.5 t_{RAS} Settings

T60NS[2:0]	t_{RAS}
0x7	8 cycles
0x6	7 cycles
:	:
0x1	2 cycles
0x0	1 cycle

(Default: 0x0)

(4) t_{RP} , t_{RCD}

t_{RP} : PRECHARGE to ACTIVE command period

t_{RCD} : ACTIVE to READ/WRITE delay time

These timing parameters can be set from 1 to 4 cycles in SDCLK using T24NS[1:0]/SDRAMC_CFG register.

Table 10.4.2.6 t_{RP} and t_{RCD} Settings

T24NS[1:0]	t_{RP} , t_{RCD}
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x0)

10.5 Control and Operation of SDRAM Interface

10.5.1 Initializing SDRAM

To use the SDRAM, it must be initialized by following the procedure below.

1. Initializing the SDRAMC registers

Set up the SDRAMC registers in the following order:

- (1) SDRAMC_CFG register
Set the SDRAM size, address configuration and access timing parameters.
- (2) SDRAMC_REF register
Set the auto-refresh and self-refresh counters.
- (3) SDRAMC_INIT register
Set SDON to 1 to enable SDRAMC.
- (4) SDRAMC_APP register
Set the CAS latency. Also enable double frequency mode if necessary.

2. Waiting after SDRAM power-on

After the power to the SDRAM is turned on, a NOP state (#SDCS = 1) must be maintained at least for a certain length of time (e.g., 100 μ s or 200 μ s). Because this duration varies with each SDRAM, refer to the specifications of the SDRAM being used.

3. Executing an SDRAM initial sequence

In order to initialize the SDRAM, the PALL (Precharge All banks), REF (Auto Refresh), and MRS (Mode Register Set) commands must be executed sequentially. Note that the initialization sequence depends on the SDRAM used.

Example 1: PALL → REF → REF → MRS (→ EMRS)

Example 2: PALL → MRS → 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 separately using the control bit shown below.

To execute the PALL (Precharge All) command:

Write 0x12 to the SDRAMC_INIT register to set INIPRE to 1.

Then write any data to any address in the SDRAM. This dummy write causes the PALL command to be sent to the SDRAM.

To execute the REF (Auto Refresh) command:

Write 0x11 to the SDRAMC_INIT register to set INIREF to 1.

Then write any data to any address in the SDRAM. This dummy write causes the REF command to be sent to the SDRAM.

When executing the REF command twice or more, insert the nop instruction between the executions.

REF command execution → nop instruction execution → REF command execution (→ REF → nop → REF...)

The SDRAM timing parameters set in the SDRAMC_CFG register is disabled when this initialization sequence is executed. 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 SDRAMC_INIT register to set INIMRS to 1.

Then write any data to a specific address of SDRAM shown below according to the CAS latency (MRS) or extended mode parameters (EMRS).

10 SDRAM CONTROLLER (SDRAMC)

Table 10.5.1.1 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 the SDRAM specification.						

For example, to execute an MRS command with CAS latency = 2, write any value to address 0x10000442 (when the SDRAM is mapped to Area 19) after writing 0x14 to the SDRAMC_INIT register.

- Notes:**
- The CAS latency specified in the MRS command must be the same as the setting for CAS[1:0]/SDRAMC_APP register.
 - After the initial sequence commands are executed, the command enable bit must be set to 0. Write 0x10 to the SDRAMC_INIT register after the last initialization command has been executed.
 - The self-refresh function must be disabled until the SDRAM has finished initialization.

4. Checking if the SDRAM has been initialized

INIDO/SDRAMC_INIT register is reset to 0 after power-on, and is set to 1 upon completion of the SDRAM initialization sequence shown above. Make sure that INIDO is set to 1 before the SDRAM is accessed.

In addition to being reset at power-on, INIDO is reset to 0 by writing 0 to SDON/SDRAMC_INIT register.

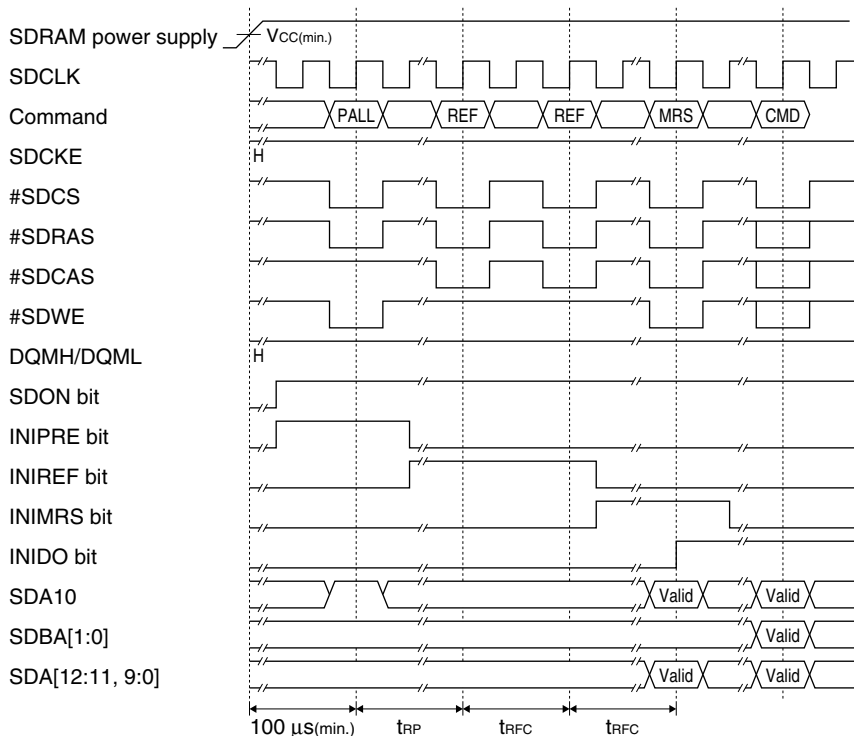


Figure 10.5.1.1 SDRAM Power-on and Initialization

10.5.2 SDRAM Commands

The SDRAM is controlled by commands that are comprised of a combination of high or low logic level signals. The table below lists the commands output by the SDRAMC.

Table 10.5.2.1 List of the Supported SDRAM Commands

Command	SDCKE	DQM (DQMH/L)	SDBA[1:0]	SDA10	SDA[12:11] SDA[9:0]	#SDCS	#SDRAS	#SDCAS	#SDWE
Deselect	H	–	–	–	–	H	–	–	–
Bank Active (ACTV)	H	–	V	V	V	L	L	H	H
Specified Bank Precharge (PRE)	H	–	V	L	–	L	L	H	L
Precharge All banks (PALL)	H	–	–	H	–	L	L	H	L
Write (WRIT)	H	–	V	L	V	L	H	L	L
Read (READ)	H	–	V	L	V	L	H	L	H
Mode Register Set (MRS)	H	–	–	V	V	L	L	L	L
No Operation (NOP)	H	–	–	–	–	L	H	H	H
Auto-Refresh (REF)	H	–	–	–	–	L	L	L	H
Self-Refresh (SELF)	H → L	–	–	–	–	L	L	L	H
End Self-Refresh	L → H	–	–	–	–	H	–	–	–
Data Write/Output Enable	H	L	–	–	–	–	–	–	–
Data Write/Output Disable	H	H	–	–	–	–	–	–	–

V = Valid, – = Optional/Unknown, 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.

10.5.3 SDRAM Bus Operations

The external data bus of the S1C33L26 is sized to 16 bits. Depending on the device size and data size of the instruction executed, two or more bus operations may occur. The table below shows bus operations in the SDRAM area.

Table 10.5.3.1 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	DQML	–	D[7:0]	1
			*	1	DQMH	D[7:0]	–	1
		R	*	0	DQMH/L	–	D[7:0]	1
			*	1	DQMH/L	D[7:0]	–	1
	Halfword	W	*	*	DQMH/L	D[15:0]		1
		R	*	*	DQMH/L	D[15:0]		1
	Word	W	0	*	DQMH/L	D[15:0]		1st
			1	*	DQMH/L	D[31:16]		2nd
		R	0	*	DQMH/L	D[15:0]		1st
			1	*	DQMH/L	D[31:16]		2nd

10.5.4 Read/Write Cycles

Read cycle

The SDRAMC always reads data from the SDRAM in bursts. The burst length is fixed to 2. Figure 10.5.4.1 shows an example of timing chart when reading out 2-word data from the same row address.

Parameter setting example: CAS latency = 2, t_{RCD} = 2 cycles, t_{RAS} = 4 cycles, t_{RP} = 2 cycles

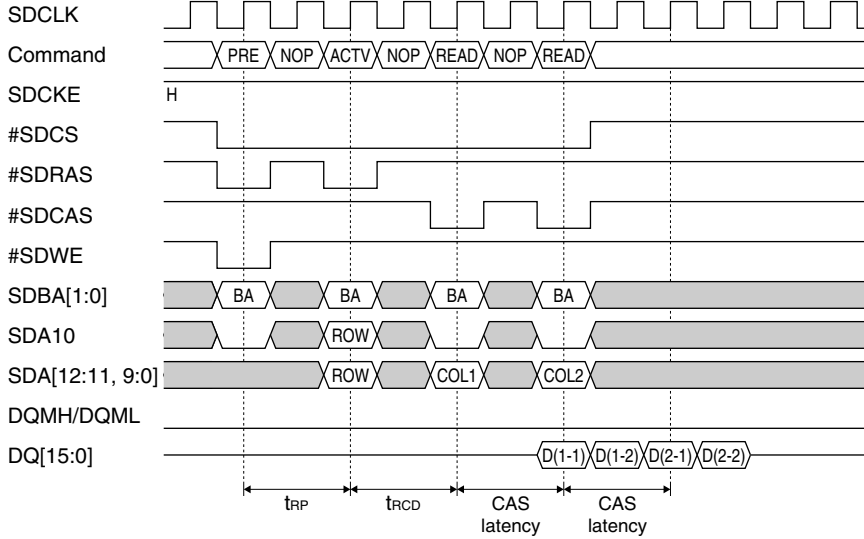


Figure 10.5.4.1 Burst Read in the Same Page

Figure 10.5.4.2 shows an example of a timing chart in cases where the row address is changed during burst read.

Parameter setting example: CAS latency = 2, t_{RCD} = 2 cycles, t_{RAS} = 4 cycles, t_{RP} = 2 cycles

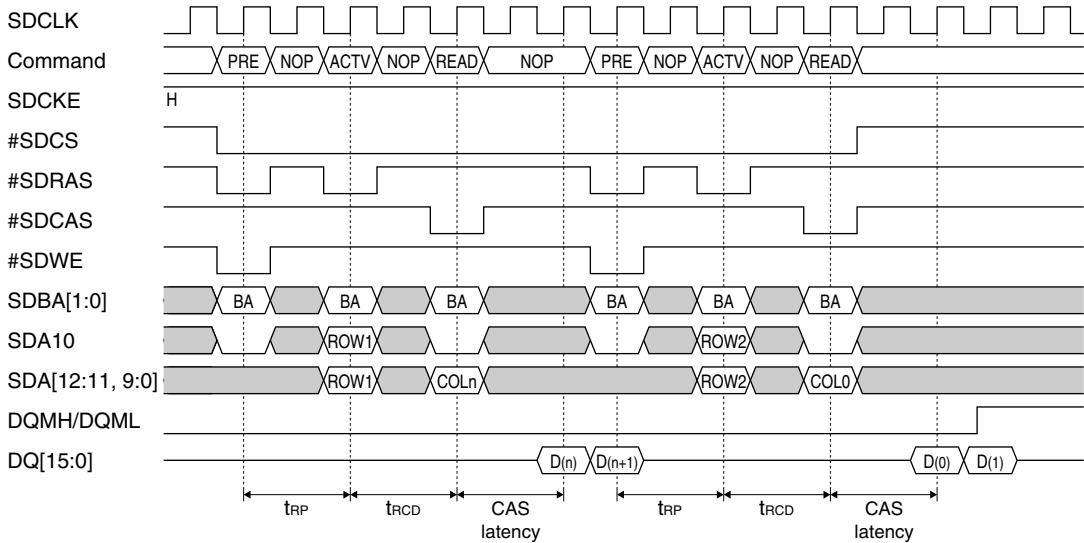


Figure 10.5.4.2 Changing Row Address During Burst Read

Write cycle

When writing to the SDRAM, data are always written in a single operation.

Parameter setting example: CAS latency = 2, trCD = 2 cycles, tRAS = 4 cycles, tRP = 2 cycles

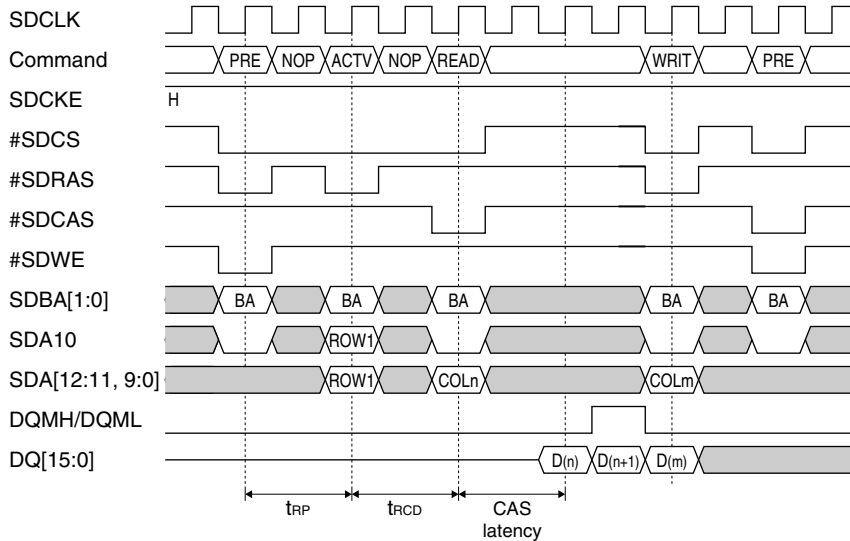


Figure 10.5.4.3 Burst Read to Single Write (same page)

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.

Parameter setting example: CAS latency = 2, trCD = 2 cycles, tRAS = 4 cycles, tRP = 2 cycles

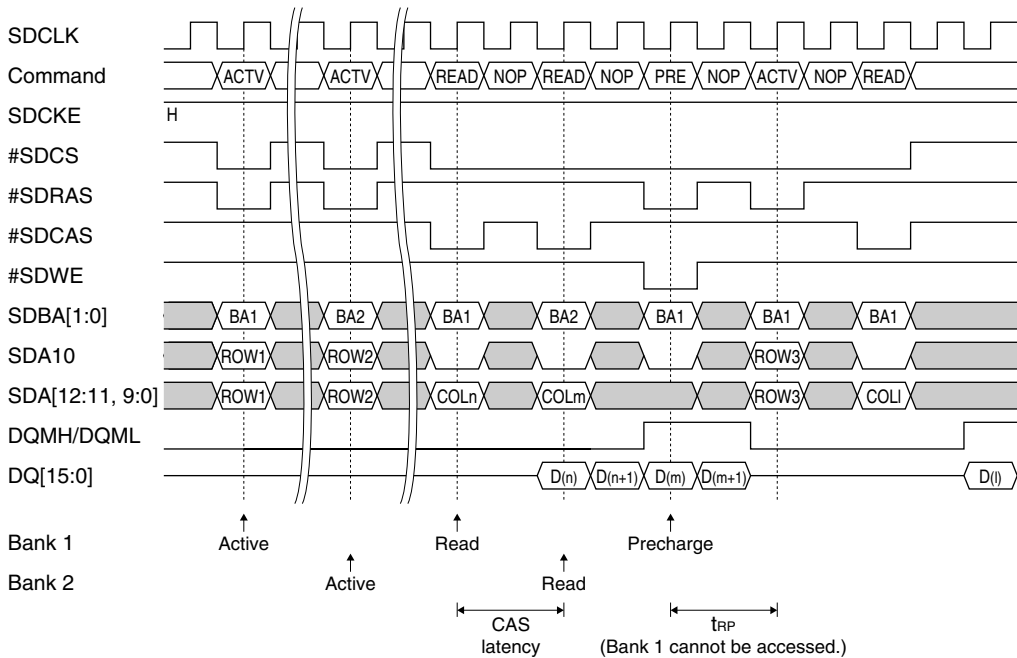


Figure 10.5.4.4 Bank Interleaved Access

10.5.5 SDRAM Refresh

The SDRAMC supports two SDRAM refresh modes: auto-refresh and self-refresh.

Auto-refresh

The SDRAMC includes 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]/SDRAMC_REF register.

AURCO[11:0] should be set to the appropriate value meeting the specifications of the SDRAM. The count value is obtained by the equation below.

$$AURCO \leq \frac{RFP}{ROWS} \times f_{CLK} - BL - CL - 2 \times t_{RP} - t_{RCD} - 3$$

- RFP: Maximum refresh period [s]
- ROWS: Row address size
- f_{CLK}: SDCLK clock frequency [Hz]
- BL: Burst length (= 2)
- CL: CAS latency
- t_{RP}: PRECHARGE command period [Number of cycles]
- t_{RCD}: ACTIVE to READ/WRITE delay time [Number of cycles]

If RFP = 64 ms, ROWS = 4,096, f_{CLK} = 20 MHz, CL = 2, t_{RP} = 2, and t_{RCD} = 2, for example, the value to set is calculated as follows:

$$AURCO \leq \frac{0.064}{4,096} \times 20,000,000 - 2 - 2 - 2 \times 2 - 2 - 3 = 299$$

Therefore, set a value equal to or less than 299 (0x12b) to AURCO[11:0].

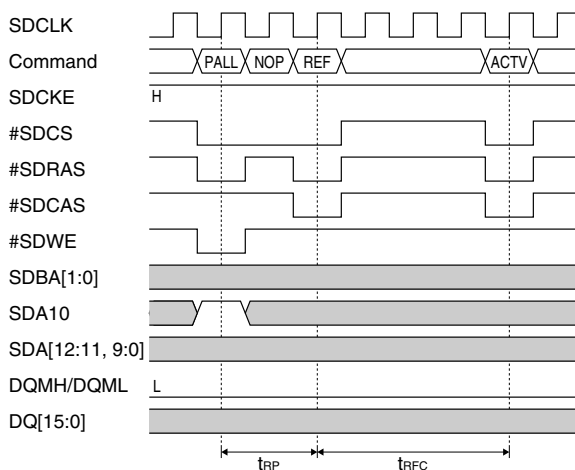


Figure 10.5.5.1 Auto-Refresh

Self-refresh

Self-refresh uses the self-refresh function of the SDRAM and does not require any clocks during the refresh period, thus helping to reduce power consumption. This self-refresh function is also used for data retention in the power-down mode. To cause the SDRAM to be self-refreshed, set SELEN/SDRAMC_REF register to 1. This enables the SDRAMC to send the self-refresh command (which sets the SDCKE output to low) to the SDRAM. The command is actually sent a certain time after accessing or auto-refreshing the SDRAM, so the SDRAMC contains a self-refresh counter to count this time. The counter counts on SDCLK clock edges, and when the designated count is reached, the SDRAMC 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 counter value can be specified in a range of 1 to 127 using the SELCO[6:0]/SDRAMC_REF register. Do not set the counter to 0 when the self-refresh function is enabled.

If an access to the SDRAM occurs while the SDRAM is in self-refresh mode, SDCKE is reset to high, thereby self-refresh mode is canceled. After the SDRAM access has finished, the SDRAMC 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.

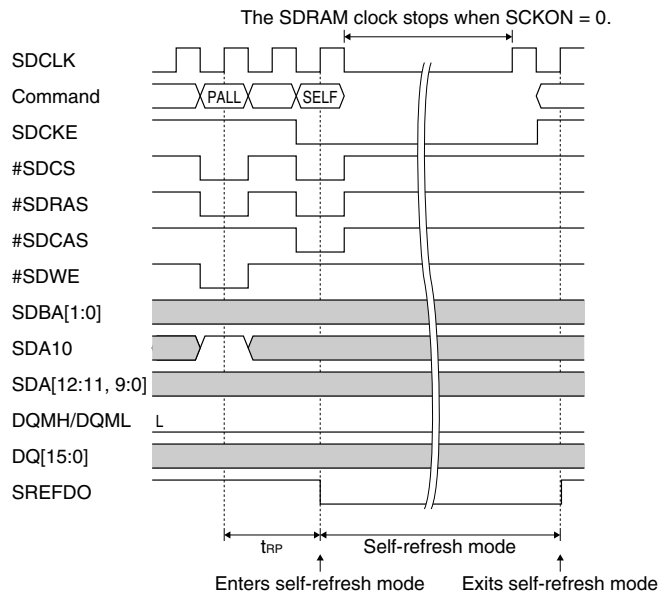


Figure 10.5.5.2 Self-Refresh

During self-refresh (while SDCKE = low), the SREFDO/SDRAMC_REF register remains 0. Therefore, it is possible to determine whether or not self-refresh is in operation by reading this status bit.

Furthermore, SDRAM clock output during self-refresh can be turned off in order to reduce power consumption by setting the SCKON/SDRAMC_REF register to 0.

To cancel self-refresh mode

Perform the following procedure to cancel self-refresh mode:

1. Disable self-refresh mode by clearing SELEN to 0.
2. Read data from an SDRAM address (any address can be specified).
3. Read SREFDO to check if self-refresh mode is canceled (SREFDO = 1).

Note that auto-refresh mode cannot be canceled.

Note: Be sure to avoid setting SDON/SDRAMC_INIT register to 0 (SDRAMC disabled) during self-refreshing. Before disabling the SDRAMC, always make sure the SDRAMC is not in self-refresh mode.

10.5.6 Power-Down Mode

The S1C33L26 supports two power-down modes for the C33 PE Core (HALT and SLEEP).

HALT mode

The SDRAM clock will be supplied in HALT mode, if it is not disabled in normal mode.

The SDRAM clock can be stopped before executing the halt instruction by controlling the CMU. To maintain data in the SDRAM during HALT status with no SDRAM clock supplied, place the SDRAM in self-refresh mode by setting SELEN/SDRAMC_REF register to 1 before disabling the SDRAM clock supply.

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 I/O ports.
4. Drive the data and address buses to low.
5. Set SDON/SDRAMC_INIT register to 0 to disable the SDRAMC.
6. Execute the slp instruction.

Perform the following procedure when the CPU wakes up from SLEEP status:

1. The CPU wakes up from SLEEP status.
2. Configure the port functions for the SDRAM.
3. Release the data and address buses from forced low driving.
4. Turn the SDRAM power on.
5. Wait at least 100 or 200 μ s for the SDRAM to be stabilized, according to the SDRAM specifications.
6. Set SDON/SDRAMC_INIT register to 1 to enable the SDRAMC.
7. Initialize the SDRAMC.

10.6 Data Queue Buffer (DQB)

The bus controller of the S1C33L26 also includes a data queue buffer (DQB) to increase the C33 PE Core memory access performance.

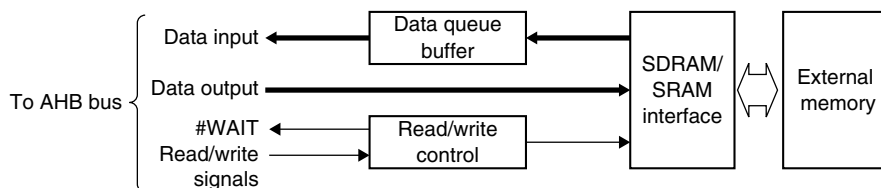


Figure 10.6.1 Data Queue Buffer

The DQB is a 4×16 -bit buffer used as an instruction/data buffer for reading from the SDRAM or the external memories of the SRAMC.

The DQB acts as a pure read buffer for storing all data read from the SDRAM or external memories of the SRAMC, regardless of whether the target is an instruction or data. Note that the DQB cannot be disabled.

Table 10.6.1 lists the DQB status corresponding to the bus operation for the SDRAM or the external memories of SRAMC.

Table 10.6.1 Correspondence between DQB Status and Bus Operation

Bus operation	DQB status
CPU instruction fetch	Enabled
CPU vector fetch	Enabled
CPU data read	Enabled
CPU data write	Disabled
CPU stack read	Enabled
CPU stack write	Disabled
DMAC data read	Enabled
DMAC data write	Disabled

If the CPU executes writing to the address of data buffered in the DQB, the buffered data concerned are flushed. Initial resetting also resets and empty the DQB.

10.7 Control Register Details

Table 10.7.1 List of SDRAMC Registers

Address	Register name		Function
0x302200	SDRAMC_INIT	SDRAM Initialization Register	Enable SDRAMC and control SDRAM initialization
0x302204	SDRAMC_CFG	SDRAM Configuration Register	Set SDRAM size and timing parameters
0x302208	SDRAMC_REF	SDRAM Refresh Control Register	Control SDRAM refresh
0x302210	SDRAMC_APP	SDRAM Application Configuration Register	Set CAS latency and double frequency mode

The following describes each SDRAMC register. These are all 32-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

SDRAM Initialization Register (SDRAMC_INIT)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
SDRAM Initialization Register (SDRAMC_INIT)	0x302200 (32 bits)	D31-5	–	reserved	–			–	–	0 when being read.
		D4	SDON	SDRAM controller enable	1	Enable	0 Disable	0	R/W	
		D3	INIDO	SDRAM initialization status	1	Finished	0 Busy	0	R	
		D2	INIMRS	MRS command enable for init.	1	Enable	0 Disable	0	R/W	
		D1	INIPRE	PALL command enable for init.	1	Enable	0 Disable	0	R/W	
		D0	INIREF	REF command enable for init.	1	Enable	0 Disable	0	R/W	

D[31:5] Reserved

D4 **SDON: SDRAM Controller Enable Bit**

Enables the SDRAM controller.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When SDON is set to 1, the SDRAM controller activates and outputs the SDRAM clock from the SD-CLK pin. Before setting SDON to 1, be sure to start SDRAMC clock supply to the SDRAM controller.

Note: Be sure to avoid setting SDON to 0 (SDRAMC disabled) during self-refreshing. Before disabling the SDRAMC, always make sure the SDRAMC is not in self-refresh mode.

D3 **INIDO: SDRAM Initialization Status Bit**

Indicates that the SDRAM has finished initialization (Mode Register Set).

1 (R): Initialization finished

0 (R): Before initialization (default)

INIDO is set to 1 when the initialization sequence is completed. Make sure that INIDO is set to 1 before the SDRAM is accessed.

D2 **INIMRS: MRS Command Enable for Initialization Bit**

Enables to output the MRS (Mode Register Set) command for initializing the SDRAM.

1 (R/W): Enabled

0 (R/W): Disabled (default)

10 SDRAM CONTROLLER (SDRAMC)

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 used. Refer to the specification 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 to set INIMRS to 1. Then write any data to a specific address of SDRAM shown below according to the CAS latency (MRS) or extended mode parameters (EMRS).

Table 10.7.2 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 the SDRAM specification.					

For example, to execute an MRS command with CAS latency = 2, write any value to address 0x10000442 (when the SDRAM is mapped to Area 19) after writing 0x14 to the SDRAMC_INIT register.

Note: The CAS latency specified in the MRS command must be the same as the setting for CAS[1:0]/SDRAMC_APP register.

D1 INIPRE: PALL Command Enable for Initialization Bit

Enables to output the PALL (Precharge All) command for initializing the SDRAM.

1 (R/W): Enabled

0 (R/W): Disabled (default)

To execute the PALL (Precharge All) command, write 0x12 to this register to set INIPRE 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 to the SDRAM. See INIMRS for the initialization sequence.

D0 INIREF: REF Command Enable for Initialization Bit

Enables to output the REF (Auto-Refresh) command for initializing the SDRAM.

1 (R/W): Enabled

0 (R/W): Disabled (default)

To execute the REF (Auto-Refresh) command, write 0x11 to this register to set INIREF 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 to the SDRAM. See INIMRS for the initialization sequence.

When executing the REF command twice or more, insert the nop instruction between the executions.

REF command execution → nop instruction execution → REF command execution (→ REF → nop → REF...)

Notes:

- The SDRAM timing parameters set in the SDRAMC_CFG register is disabled when the initialization sequence is executed. Therefore, enough number of nop instructions must be executed to satisfy the SDRAM timings.

- After the initial sequence commands are executed, the command enable bit must be set to 0. Write 0x10 to the SDRAMC_INIT register after the last initialization command has been executed.

- The self-refresh function must be disabled until the SDRAM has finished initialization.

SDRAM Configuration Register (SDRAMC_CFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SDRAM Configuration Register (SDRAMC_CFG)	0x302204 (32 bits)	D31–14	–	reserved	–	–	–	0 when being read.
		D13–12	T24NS[1:0]	Number of SDRAM trp and trcd cycles	T24NS[1:0] # of cycles 0x3 4 cycles 0x2 3 cycles 0x1 2 cycles 0x0 1 cycle	0x0	R/W	
		D11	–	reserved	–	–	–	0 when being read.
		D10–8	T60NS[2:0]	Number of SDRAM tras cycles	T60NS[2:0] # of cycles 0x7 8 cycles 0x6 7 cycles : : 0x1 2 cycles 0x0 1 cycle	0x0	R/W	
		D7–4	T80NS[3:0]	Number of SDRAM trc, trfc and txsr cycles	T80NS[3:0] # of cycles 0xf 16 cycles 0xe 15 cycles : : 0x1 2 cycles 0x0 1 cycle	0xe	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	ADDRC[2:0]	SDRAM address configuration	ADDRC[2:0] Configuration 0x7 512M bits 0x6 128M bits x 2 0x5 64M bits x 2 0x4 reserved 0x3 256M bits 0x2 128M bits 0x1 64M bits 0x0 16M bits	0x0	R/W	Do not set to 0x4.

D[31:14] Reserved

D[13:12] T24NS[1:0]: Number of SDRAM trp and trcd Cycles Bits

Sets the trp and trcd SDRAM timing parameters.

- trp PRECHARGE to ACTIVE command period
- trcd ACTIVE to READ/WRITE delay time

Table 10.7.3 trp and trcd Settings

T24NS[1:0]	trp, trcd
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x0)

D11 Reserved

D[10:8] T60NS[2:0]: Number of SDRAM tras Cycles Bits

Sets the tras SDRAM timing parameter.

- tras ACTIVE to PRECHARGE command period

Table 10.7.4 tras Settings

T60NS[2:0]	tras
0x7	8 cycles
0x6	7 cycles
:	:
0x1	2 cycles
0x0	1 cycle

(Default: 0x0)

D[7:4] T80NS[3:0]: Number of SDRAM trc, trfc, and txsr Cycles Bits

Sets the trc, trfc and txsr SDRAM timing parameters.

- trc ACTIVE to ACTIVE command cycle time
- trfc Auto-refresh cycle time
- txsr Self-refresh end to ACTIVE command period

10 SDRAM CONTROLLER (SDRAMC)

Table 10.7.5 trc, trfc, and txsr Settings

T80NS[3:0]	trc, trfc, txsr
0xf	16 cycles
0xe	15 cycles
0xd	14 cycles
0xc *	13 cycles
0xb	12 cycles
0xa	11 cycles
0x9	10 cycles
0x8 *	9 cycles
0x7	8 cycles
0x6	7 cycles
0x5	6 cycles
0x4 *	5 cycles
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0 *	1 cycle

(Default: 0xe)

* Recommended settings (For more information, see “(2) External SRAM access rate while the SDRAM is in self-refresh status” in Section 3.8.)

D3 Reserved

D[2:0] ADDR[2:0]: SDRAM Address Configuration Bits

Selects SDRAM size and chip configuration. This selection also sets up the bank size, column address size (page size), and row address size.

Table 10.7.6 SDRAM Size Selections

ADDR[2:0]	Number of banks	Row size	Column size	SDRAM chip configuration	Memory size
0x7	4	8K	1K	32M × 16 bits × 1	64M bytes
0x6	4	4K	1K	16M × 8 bits × 2	32M bytes
0x5	4	4K	512	8M × 8 bits × 2	16M bytes
0x4	Reserved (Do not set ADDR[2:0] to 0x4.)				
0x3	4	8K	512	16M × 16 bits × 1	32M bytes
0x2	4	4K	512	8M × 16 bits × 1	16M bytes
0x1	4	4K	256	4M × 16 bits × 1	8M bytes
0x0	2	2K	256	1M × 16 bits × 1	2M bytes

(Default: 0x0)

SDRAM Refresh Register (SDRAMC_REF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
SDRAM Refresh Control Register (SDRAMC_REF)	0x302208 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.	
		D25	SREFDO	SDRAM self-refresh status	1 Finished 0 Busy	0	R		
		D24	SCKON	SDRAM clock during self-refresh	1 Enable 0 Disable	0	R/W		
		D23	SELEN	SDRAM self-refresh enable	1 Enable 0 Disable	0	R/W		
		D22–16	SELCO[6:0]	SDRAM self-refresh counter	0x0 to 0x7f	0x7f	R/W		
		D15–12	–	reserved	–	–	–	–	0 when being read.
		D11–0	AURCO[11:0]	SDRAM auto-refresh counter	0x0 to 0xfff	0x8c	R/W		

D[31:26] Reserved

D25 SREFDO: SDRAM Self-Refresh Status Bit

Indicates the SDRAM self-refresh status.

1 (R): Self-refresh has finished

0 (R): Self-refresh mode (default)

SREFDO is set to 0 while the SDRAM is placed into self-refresh mode. Otherwise, SREFDO is set to 1. Before entering the 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 During Self-Refresh Bit

Specifies whether to stop the SDRAM clock during self-refresh or not.

1 (R/W): Enabled (output continued)

0 (R/W): Disabled (output disabled) (default)

Writing 0 to SCKON disables the SDRAM clock output from the SDCLK pin while the SDRAM is in self-refresh mode. This helps to reduce the current consumption.

If SCKON is 1, the SDRAM clock is always output from the SDCLK pin even if the SDRAM is in self-refresh mode.

D23 SELEN: SDRAM Self-Refresh Enable Bit

Enables the self-refresh control function of the SDRAM.

1 (R/W): Enabled

0 (R/W): Disabled (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].

To cancel self-refresh mode, perform the following procedure:

1. Disable self-refresh mode by clearing SELEN to 0.
2. Read data from an SDRAM address (any address can be specified).
3. Read SREFDO to check if self-refresh mode is canceled (SREFDO = 1).

D[22:16] SELCO[6:0]: SDRAM Self-Refresh Counter Bits

Sets the value for the self-refresh counter. (Default: 0x7f)

If SELEN 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, SDCKE returns high, thereby self-refresh mode is canceled.

D[15:12] Reserved**D[11:0] AURCO[11:0]: SDRAM Auto-Refresh Counter Bits**

Sets 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.

$$\text{AURCO} \leq \frac{\text{RFP}}{\text{ROWS}} \times f_{\text{CLK}} - \text{BL} - \text{CL} - 2 \times \text{TRP} - \text{TRCD} - 3$$

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/WRITE delay time [Number of cycles]

SDRAM Application Configuration Register (SDRAMC_APP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
SDRAM Application Configuration Register (SDRAMC_APP)	0x302210 (32 bits)	D31-6	–	reserved	–	–	–	0 when being read.	
		D5	DBF	Double frequency mode enable	1 Enable 0 Disable	0	W		
		D4	–	reserved	–	–	–	–	
		D3-2	CAS[1:0]	CAS latency setup	CAS[1:0]	CAS latency	0x2	R/W	
					0x3	3			
					0x2	2			
0x1	1								
0x0	reserved								
D1	–	reserved	–	–	–	0 when being read.			
D0	–	reserved	–	0	R/W	Do not set to 1.			

D[31:6] Reserved

D5 **DBF: Double Frequency Mode Enable Bit**

Enables double frequency mode.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Setting DBF to 1 sets the SDRAMC to operate in the double frequency mode where it can use the SDRAM clock (72 MHz max.) that is two times faster than the CPU clock (36 MHz max.). Set DBF to 0 to use the SDRAM clock at the same frequency as the CPU clock (60 MHz max.).

D4 **Reserved**

D[3:2] **CAS[1:0]: CAS Latency Setup Bits**

Sets the CAS latency.

CAS latency refers to the number of SDCLK clock cycles counted until data is output from the SDRAM after issuing the READ command.

Table 10.7.7 CAS Latency Settings

CAS[1:0]	CAS latency
0x3	3
0x2	2
0x1	1
0x0	Reserved

(Default: 0x2)

D1 **Reserved**

D0 **Reserved (Always set this bit to 0.)**

10.8 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. Do not perform these operations when the SDRAM may be accessed.

- Placing the S1C33L26 into SLEEP status
- Disabling the clock supply to the SDRAMC module

Besides from the CPU, the SDRAM can be accessed from the DMAC (if DMA transfers are enabled toward the SDRAM). In this case, before performing the above operations, stop the DMAC to disable its access to the SDRAM.

11 Cache Controller (CCU)

11.1 CCU Module Overview

In order to enable fast access to instructions and data, the S1C33L26 incorporates a cache controller (CCU) that runs by the 4-Way set associative method. Addresses 0x1f800 to 0x1fbff (1K bytes) and 0x1fc00 to 0x1ffff (1K bytes) in Area 0 are used as cache memories for instructions and data, respectively, enabling fast access to external ROM/SRAM/SDRAM in the specified area (excluding access to the embedded memory and internal peripheral modules from data to be cached).

The cache can be locked before executing an interrupt handler routine of the specified interrupt level, this makes it possible to avoid refilling the cache by a lower priority interrupt when a routine that requires high-speed processing has been cached. Also, the cache is automatically locked in debugging mode, enabling debugging in a hardware break in the same timing as the normal operation.

The main features of the CCU are outlined below.

- Cache adopting the 4-Way set associative method with separate memories for instructions (1K bytes) and data (1K bytes).
- One area can be selected separately for each of the categories, instructions and data, (from Areas 14 to 22) as the area for caching.
- One-word write buffer is built in to support write through mode.
- Refill is performed using the LRU algorithm.
- A four-word burst reading function is provided to reduce waiting time for refill.
- A locking function works in debugging and interrupts (with specification of the interrupt level).
- An automatic flush function is provided for the instruction cache to work in response to software PC break in debugging.
- The instruction cache RAM and data cache RAM can be used as a general-purpose RAM when the cache function is disabled.

Table 11.1.1 Cache Speed

Status	Number of cycles
Reading from the instruction or data cache (upon hitting)	2 cycles *1
Concurrent reading from the instruction and data caches (upon hitting)	
Reading from IRAM	
Writing to the data cache (upon hitting and emptying the write buffer)	2 cycles *2
Writing to the data cache (upon mishitting and emptying the write buffer)	
Writing to IRAM	1 cycle
Reading from the instruction or data cache (upon mishitting)	Depends on the access setting for the external memory
Writing to the data cache (when the write buffer is full)	

*1: As the instruction and data caches can be accessed at the same time, performance is maintained even when data are accessed while instructions are being fetched.

*2: Data are written to the data cache and the write buffer at the same time in the first cycle and written to an external memory in the next cycle.

Notes:

- The CCU does not have a snooping function (for maintaining the data in the cache memory to match those in the external memory). The cache and the external memory are maintained in synch if reading/writing is only executed in the C33 PE Core. Use software to secure data integrity in cases where data are shared with the DMAC.

- When the CPU executes the halt or slp instruction, the clocks supplied to the C33 PE Core and cache both stop. To avoid unexpected bus operations, lock or disable the cache and make sure that the cache has been locked or disabled using the CCU_STAT register.

11.2 Cache Configuration

The CCU uses addresses 0x1f800 to 0x1fbff (1K bytes) and 0x1fc00 to 0x1fff (1K bytes) in Area 0 for the instruction cache and the data cache, respectively. The instruction cache and the data cache can be separately enabled in software. When using the CCU, take care not to allow data to be accidentally written to the cache from an application. The cache memory spaces only store data sections; the TAG sections are stored in the memory within the CCU module.

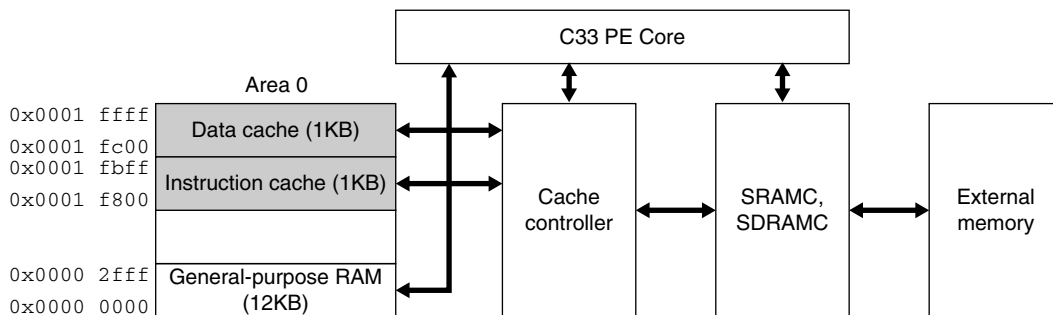


Figure 11.2.1 Cache Memory

The CCU adopts the 4-Way set associative method.

One frame is composed of cache data containing four lines (4 × 4 words), and one Way consists of four frames.

Four frames located at the corresponding area in each Way are managed under one LRU entry.

Data to a cache from the external memory are loaded in units of a line (four words).

Figure 11.2.2 shows the cache configuration.

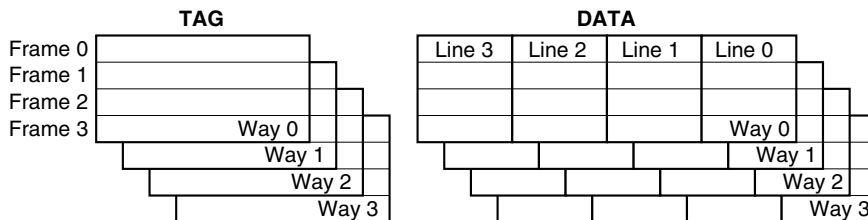


Figure 11.2.2 Cache Configuration

TAG and Data Sections

Each frame is divided into the TAG and Data sections as shown in Figure 11.2.3. The TAG section stores 18-bit addresses for comparison. The Data section consists of four words (16 bytes each) by four lines.

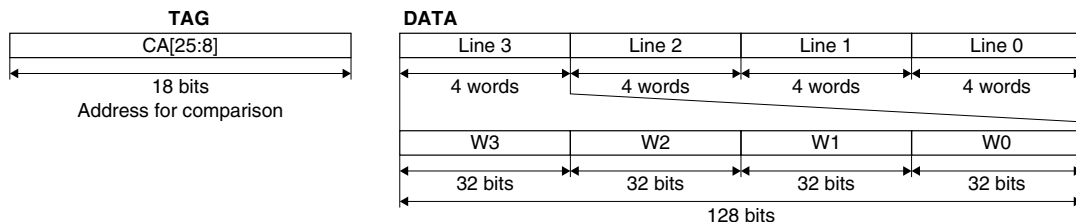


Figure 11.2.3 Frame Configuration

LRU Section

Configured with 4-Ways, the CCU has four frames of data assigned the same entry number. If nothing is hit, it is needed to select one of the four Ways to replace with, in which case, the LRU section stores the Way number.

11.3 Cache Settings and Operations

11.3.1 Cache Enable

At initial reset, the caching function is disabled.

Set up as follows to use caching.

When the instruction cache is used

Set the IC/CCU_CFG register to 1.

When the data cache is used

Set the DC/CCU_CFG register to 1.

Turning IC and DC back to 0 flushes and clears all data cached.

Note: Be sure to disable the instruction and data caches before executing the halt or slp instruction.

11.3.2 Selecting Area to Be Cached

The CCU caches access to one area (an external memory) out of Areas 14 to 22. Select a target area for caching in the instruction cache in ARIC[2:0]/CCU_AREA register and one for caching in the data cache in ARDC[2:0]/CCU_AREA register, respectively.

Table 11.3.2.1 Selecting Area to Be Cached

ARIC[2:0]/ARDC[2:0]	Areas to be cached
0x7	Area 22 (0x80000000 to 0xffffffff)
0x6	Area 21 (0x40000000 to 0x7fffffff)
0x5	Area 20 (0x20000000 to 0x3fffffff)
0x4	Area 19 (0x10000000 to 0x1fffffff)
0x3	Area 18 (0x0c000000 to 0x0fffffff)
0x2	Area 17 (0x08000000 to 0x0bffffff)
0x1	Areas 15 and 16 (0x04000000 to 0x07ffffff)
0x0	Area 14 (0x03000000 to 0x03ffffff)

(Default: 0x0)

The CCU only caches access to the range of 64MB starting at the top of the selected area. The areas after the leading 64MB space in Areas 19 to 22 are mirror areas.

11.3.3 Comparing Addresses and Cache Hit/Mishit

Among addresses output by the SRAMC, two bits at A[7:6] represent the entry number (frame offset). A[25:8] is deemed as a comparison address and compared with the address for comparison (CA[25:8]) stored in the TAG section containing four Ways under the entry selected in A[7:6].

11 CACHE CONTROLLER (CCU)

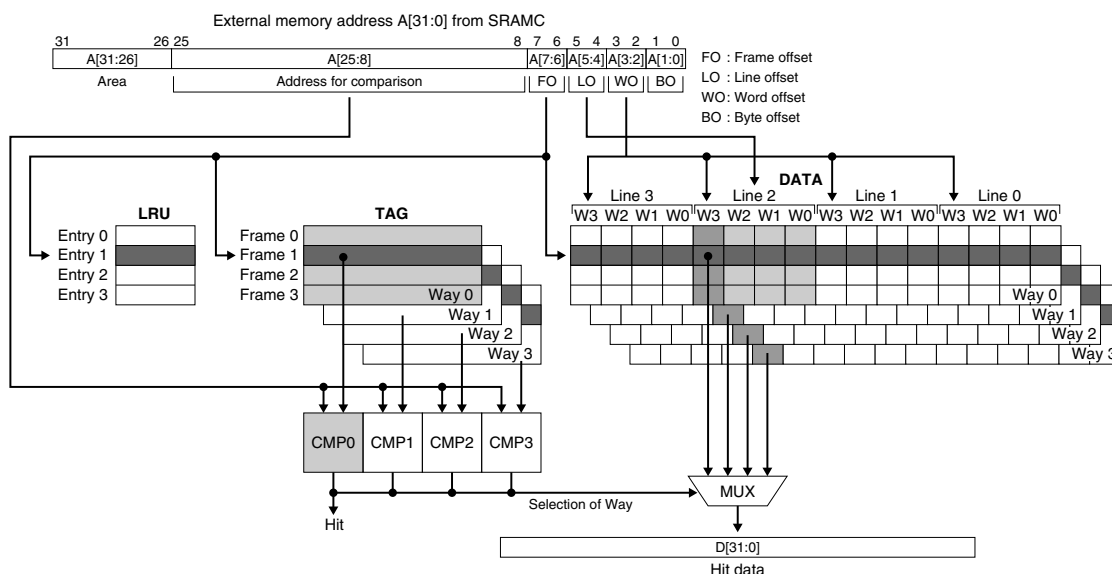


Figure 11.3.3.1 Cache-wise Operation

The following describes operation up to cache hit/mishit judgement.

1. Generates entry numbers (0–3) from address A[7:6] output by the SRAMC.
2. Reads information for four Ways from the TAG section in the selected entry. Reads word data for four Ways at the same time from the Data section that are indicated by the line offset (A[5:4]) and the word offset (A[3:2]).
3. Compares CA[25:8] in the TAG section in each Way against A[25:8].
4. It is judged as Hit if a matching Way is found in Step 3 and at the same time data in the relevant line are valid (cached from the external memory). At this stage, which Way is hit is determined.
If no matching Way is found in Step 3, it is judged as mishit.

For example, if Way 0 is hit at A[7:6] = 0b01, A[5:4] = 0b10 and A[3:2] = 0b11, reading/writing is performed from/ to Way 0 - Frame 1 - Line 2 - W3 in the cache memory.

11.3.4 Reading Operation

The following describes operations for cases where any data are hit or not hit in reading.

When any data is hit

The 32-bit data hit are transferred to the CPU and at the same time the LRU information of the relevant entry is updated.

When any data is mishit

No instructions or data in the cache are transferred to the CPU.

If no data is hit, the CCU updates (refills) the replace Way (the Way accessed earliest) obtained from the current LRU information. Refill operation takes place in units of one line (containing four words each). The CCU reads four words including the subject instruction or data from the external memory and writes the words to the relevant frame/line in the cache memory. At the same time, it transfers the subject instruction or data to the CPU and updates LRU information. The line cached is enabled in the CCU, after which instructions/data in the same line are read from the cache memory. Other lines within the same frame are disabled until it is refilled through access to the relevant addresses.

11.3.5 Writing Operation

Writing only supports the write through mode.

The write through mode enables the relevant external memory to be written at the same time when the cache memory is written in response to a request for writing to the data address in the cache. Thus, data matching is always ensured (if not changed by the DMAC). Also, in order to prevent the processing speed from dropping due to write through operation, the CCU incorporates a one-word write buffer. To enable the write buffer, set WBEN/CCU_CFG register to 1 (default). When WBEN is set to 0, the write buffer is disabled. In this case the CPU is placed into wait status until the data writing to the external memory has completed.

When any data is hit

A write cycle is issued to both the cache and the external memory. Additionally, LRU information is updated.

When any data is mishit

When any data is mishit, the cache memory is not written, and only the external memory is written. If this is the case, LRU information is not updated since nothing is written to the cache. No refilling is performed, either.

When the write buffer is enabled, writing data to the external memory is performed in two steps, first the data is written to the write buffer and then the external memory is updated. WBEMPTY/CCU_WB_STAT register and WEFINISH/CCU_WB_STAT register are provided to check the write buffer status and whether the buffered data has been actually written to the external memory or not.

WBEMPTY is set to 0 by writing data to the write buffer and is set to 1 when the buffered data is read out for writing to the external memory. At the same time the WBEMPTY is set to 1, WEFINISH goes 0 to indicate that the data is being written to the external memory. WEFINISH is set to 1 upon completion of writing to the external memory. The write buffer improves writing speed as the CPU does not need to wait for completion of writing to the external memory. However, read the above flags to check if the data has been written to, especially when data is written to a low-speed external device.

11.3.6 Flush

Flushing refers to nullifying all data in the cache.

To flush the instruction cache, set IC/CCU_CFG register to 0 (to disable the instruction cache). To flush the data cache, set DC/CCU_CFG register to 0 (to disable the data cache).

Set these bits back to 1 to enable the caches again.

Note that the cache is flushed several cycles after writing 0 to IC or DC. Before resuming caching operation, check the status bit to ensure that flushing is completed. Check ICS/CCU_STAT register for the operating status of the instruction cache, and DCS/CCU_STAT register for that of the data cache. If these bits are 1, then the cache is in operation and the flushing is not completed. When the flushing has completed and cache stops the operation, then these bits turns to 0.

11.4 Cache Lock with Interrupt Level Specified

The cache memory size is not large, therefore, frequently generated refilling may cause system performance degradation depending on the program. To avoid this, the CCU has a feature to lock the cache against refilling with interrupt handler routines of the specified levels. LKPRI[7:0]/CCU_LK register is provided for this setting.

By setting an LKPRI[7:0] bit to 1, the priority level of interrupts to disabled refilling can be selected. Each LKPRI[7:0] bit corresponds to an interrupt level, for example, LKPRI0 corresponds to interrupt level 0 (IL[2:0] = 0) and LKPRI7 corresponds to interrupt level 7 (IL[2:0] = 7). If the interrupt level in IL[2:0] (set by the interrupt occurred) and an LKPRI[7:0] bit that has been set to 1 are matched, the cache will be locked after a lapse of several cycles. After this point, the CCU will not refill the cache until the IL[2:0] value is altered to the interrupt level of an LKPRI[7:0] bit that has been set to 0.

Example 1: When LKPRI1 is set to 1 and other bits are set to 0 (LKPRI[7:0] = 0x02, the current IL[2:0] value is 0)

The CCU can refill until a level 1 interrupt occurs. When IL[2:0] is set to 1 due to occurrence of a level 1 interrupt, the cache is locked. Therefore, the handler routine for the interrupt occurred is executed at the store location in the memory (when not hit to the cached data). Executing the reti instruction at the end of the interrupt handler routine reset IL[2:0] to 0 and releases the cache lock status.

11 CACHE CONTROLLER (CCU)

If a higher priority interrupt occurs while an interrupt handler routine is being executed, the cache lock is released as IL[2:0] is altered. In this case, the cache will be locked again after the handler routine for the higher priority interrupt ends by the reti instruction as IL[2:0] returns to level 1.

Example 2: When all the LKPRI[7:0] bits to 1 (0xff)

The cache is locked immediately after this setting, then no refilling will occur. When no cached data is hit, the program is executed at the store location in the memory. If LKPRI[7:0] is altered and the bit corresponding to the interrupt level set in IL[2:0] is reset to 0, the cache lock is released.

Read ICLKS/CCU_STAT register and DCLKS/CCU_STAT register to check whether the instruction and data caches are locked or not. If the status bit is 1, the cache is locked. If the bit is 0, the cache is not locked.

To release the lock status, set all the LKPRI[7:0] bits to 0.

11.5 Caching Operation during Debugging

In debugging mode, the automatic lock function works on both the instruction cache and the data cache.

To execute the program in debugging mode with the same timings and performance as normal operating mode, set SBRK/CCU_CFG register to 0 and use only the hardware PC break function for suspending program execution.

Setting SBRK to 1 (default) allows use of both hardware PC break and software PC break. However, execution timings and performance will not be the same as those in normal operating mode.

Note: When SBRK is set to 0, a software PC break point cannot be set in the a target area for caching.

11.6 Cache Data Integrity

The CCU does not support a snooping function (for maintaining the data in the cache memory to match those in the external memory). The cache and the external memory are maintained in synch if reading/writing is only executed in the C33 PE Core. When data are transferred to the area subject to caching via DMAC or when data are written to the program area subject to caching by the CPU, flush the cache or otherwise ensure data integrity using software.

11.7 Control Register Details

Table 11.7.1 List of CCU Registers

Address	Register name	Function
0x302300	CCU_CFG	Cache Configuration Register
0x302304	CCU_AREA	Cacheable Area Select Register
0x302308	CCU_LK	Cache Lock Register
0x30230c	CCU_STAT	Cache Status Register
0x302318	CCU_WB_STAT	Cache Write Buffer Status Register
0x302360	CCU_CCLKDV	CCLK Division Ratio Select Register

The following describes each CCU register. These are all 32-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

Cache Configuration Register (CCU_CFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Cache Configuration Register (CCU_CFG)	0x302300 (32 bits)	D31-9	–	reserved	–	–	–	0 when being read.
		D8	WBEN	Write buffer enable	1 Enable 0 Disable	1	R/W	
		D7-4	–	reserved	–	–	–	0 when being read.
		D3	–	reserved	–	–	–	Do not set to 1.
		D2	SBRK	Software break enable	1 Enable 0 Disable	1	R/W	
		D1	IC	Instruction cache enable	1 Enable 0 Disable	0	R/W	
		D0	DC	Data cache enable	1 Enable 0 Disable	0	R/W	

D[31:9] Reserved

D8 WBEN: Write Buffer Enable Bit

Enables the write buffer.

1 (R/W): Enabled (default)

0 (R/W): Disabled

When WBEN is set to 1 (default), the write buffer is enabled. When the write buffer is enabled, writing data to the external memory is performed in two steps, first the data is written to the write buffer and then the external memory is updated.

When WBEN is set to 0, the write buffer is disabled. In this case the CPU is placed into wait status until the data writing to the external memory has completed.

D[7:3] Reserved (Do not set D3 to 1.)

D2 SBRK: Software Break Enable Bit

Enables the software PC break function during debugging.

1 (R/W): Enabled (default)

0 (R/W): Disabled

Note: When SBRK is set to 0, a software PC break point cannot be set in the a target area for caching.

D1 IC: Instruction Cache Enable Bit

Enables the instruction cache.

1 (R/W): Enabled

0 (R/W): Disable (default)

By setting IC to 1, addresses 0x1f800 to 0x1fbff (1KB) in Area 0 are set for use as the instruction cache, after which the cache is used for fetching instructions from the specified area.

Setting IC to 0 flushes the instruction cache and clears all cache data. Note that the cache is flushed several cycles after writing 0 to IC. Before resuming caching operation, check ICS/CCU_STAT register to ensure that flushing is completed.

D0 DC: Data Cache Enable Bit

Enables the data cache.

1 (R/W): Enabled

0 (R/W): Disable (default)

By setting DC to 1, addresses 0x1fc00 to 0x1ffff (1KB) in Area 0 are set for use as the data cache, after which the cache is used for reading data from the specified area. Data is written in the write through mode.

Setting DC to 0 flushes the data cache and clears all cache data. Note that the cache is flushed several cycles after writing 0 to DC. Before resuming caching operation, check DCS/CCU_STAT register to ensure that flushing is completed.

Note: Be sure to disable the instruction and data caches before executing the halt or slp instruction.

Cacheable Area Select Register (CCU_AREA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Cacheable Area Select Register (CCU_AREA)	0x302304 (32 bits)	D31-7	—	reserved	—	—	—	0 when being read.	
		D6-4	ARIC[2:0]	Instruction cache area select	ARIC[2:0]	Area	0x0	R/W	
					0x7	Area 22			
					0x6	Area 21			
					0x5	Area 20			
					0x4	Area 19			
					0x3	Area 18			
					0x2	Area 17			
					0x1	Areas 15 & 16			
					0x0	Area 14			
D3	—	reserved	—	—	—	0 when being read.			
D2-0	ARDC[2:0]	Data cache area select	ARDC[2:0]	Area	0x0	R/W			
			0x7	Area 22					
			0x6	Area 21					
			0x5	Area 20					
			0x4	Area 19					
			0x3	Area 18					
			0x2	Area 17					
			0x1	Areas 15 & 16					
			0x0	Area 14					

D[31:7] Reserved

D[6:4] ARIC[2:0]: Instruction Cache Area Select Bits

Selects the area to read the instruction from via the instruction cache. (See Table 11.7.2)

D3 Reserved**D[2:0] ARDC[2:0]: Data Cache Area Select Bits**

Selects the area to read the data from via the data cache.

Table 11.7.2 Selecting Area to Be Cached

ARIC[2:0]/ARDC[2:0]	Areas to be cached
0x7	Area 22 (0x80000000 to 0xffffffff)
0x6	Area 21 (0x40000000 to 0x7fffffff)
0x5	Area 20 (0x20000000 to 0x3fffffff)
0x4	Area 19 (0x10000000 to 0x1fffffff)
0x3	Area 18 (0x0c000000 to 0x0fffffff)
0x2	Area 17 (0x08000000 to 0x0bffffff)
0x1	Areas 15 and 16 (0x04000000 to 0x07ffffff)
0x0	Area 14 (0x03000000 to 0x03ffffff)

(Default: 0x0)

The CCU only caches access to the range of 64MB starting at the top of the selected area. The areas after the leading 16MB space in Areas 19 to 22 are mirror areas.

Cache Lock Register (CCU_LK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Cache Lock Register (CCU_LK)	0x302308 (32 bits)	D31–8	–	reserved	–	–	–	0 when being read.
		D7	LKPRI7	Interrupt level 7 cache-lock enable	1 Lock 0 Unlock	0	R/W	
		D6	LKPRI6	Interrupt level 6 cache-lock enable	1 Lock 0 Unlock	0	R/W	
		D5	LKPRI5	Interrupt level 5 cache-lock enable	1 Lock 0 Unlock	0	R/W	
		D4	LKPRI4	Interrupt level 4 cache-lock enable	1 Lock 0 Unlock	0	R/W	
		D3	LKPRI3	Interrupt level 3 cache-lock enable	1 Lock 0 Unlock	0	R/W	
		D2	LKPRI2	Interrupt level 2 cache-lock enable	1 Lock 0 Unlock	0	R/W	
		D1	LKPRI1	Interrupt level 1 cache-lock enable	1 Lock 0 Unlock	0	R/W	
		D0	LKPRI0	Interrupt level 0 cache-lock enable	1 Lock 0 Unlock	0	R/W	

D[31:8] Reserved**D[7:0] LKPRI[7:0]: Interrupt Level [7:0] Cache-Lock Enable Bits**

Selects the interrupt levels to lock the cache (the interrupt levels of the interrupt handler routines to disable refilling).

1 (R/W): Lock the cache when IL[2:0] = specified interrupt level

0 (R/W): Release the cache lock when IL[2:0] = specified interrupt level (default)

By setting an LKPRI[7:0] bit to 1, the priority level of interrupts to disabled refilling can be selected. Each LKPRI[7:0] bit corresponds to an interrupt level, for example, LKPRI0 corresponds to interrupt level 0 (IL[2:0] = 0) and LKPRI7 corresponds to interrupt level 7 (IL[2:0] = 7). If the interrupt level in IL[2:0] (set by the interrupt occurred) and an LKPRI[7:0] bit that has been set to 1 are matched, the cache will be locked after a lapse of several cycles. After this point, the CCU will not refill the cache until the IL[2:0] value is altered to the interrupt level of an LKPRI[7:0] bit that has been set to 0.

Cache Status Register (CCU_STAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Cache Status Register (CCU_STAT)	0x30230c (32 bits)	D31–4	–	reserved	–	–	–	0 when being read.
		D3	ICLKS	Instruction cache lock status	1 Locked 0 Not locked	X	R	
		D2	DCLKS	Data cache lock status	1 Locked 0 Not locked	X	R	
		D1	ICS	Instruction cache operating status	1 Active 0 Inactive	X	R	
		D0	DCS	Data cache operating status	1 Active 0 Inactive	X	R	

D[31:4] Reserved

D3 ICLKS: Instruction Cache Lock Status Bit

Indicates the lock status of the instruction cache. (Default: undefined)

- 1 (R): Locked
0 (R): Not locked

Read ICLKS to check whether the instruction cache is locked or not when the cache lock function is enabled.

D2 DCLKS: Data Cache Lock Status Bit

Indicates the lock status of the data cache. (Default: undefined)

- 1 (R): Locked
0 (R): Not locked

Read DCLKS to check whether the data cache is locked or not when the cache lock function is enabled.

D1 ICS: Instruction Cache Operating Status Bit

Indicates the operating status of the instruction cache. (Default: undefined)

- 1 (R): Active
0 (R): Inactive

Setting IC/CCU_CFG register to 1 activates the instruction cache and sets ICS to 1. Setting IC to 0 flushes the instruction cache and disables caching. Note that the instruction cache is flushed several cycles after writing 0 to IC. When flushing finishes, ICS is reset to 0. To resume caching after flushing, check that flushing has completed by reading ICS.

D0 DCS: Data Cache Operating Status Bit

Indicates the operating status of the data cache. (Default: undefined)

- 1 (R): Active
0 (R): Inactive

Setting DC/CCU_CFG register to 1 activates the data cache and sets DCS to 1. Setting DC to 0 flushes the data cache and disables caching. Note that the data cache is flushed several cycles after writing 0 to DC. When flushing finishes, DCS is reset to 0. To resume caching after flushing, check that flushing has completed by reading DCS.

Cache Write Buffer Status Register (CCU_WB_STAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Cache Write Buffer Status Register (CCU_WB_STAT)	0x302318 (32 bits)	D31-10	–	reserved	–	–	–	0 when being read.
		D9	WEFINISH	Write-finish flag	1 Finished 0 Writing	1	R	
		D8	WBEMPTY	Write buffer empty flag	1 Empty 0 Full	1	R	
		D7-0	–	reserved	–	X	–	

D[31:10] Reserved**D9 WEFINISH: Write-Finish Flag Bit**

Indicates whether the data writing from the write buffer to the external memory has finished or not.

- 1 (R): Finished (default)
0 (R): During writing

When the write buffer is enabled, writing data to the external memory is performed in two steps, first the data is written to the write buffer and then the external memory is updated. WEFINISH is set to 0 when data is loaded to the write buffer and is set to 1 upon completion of writing to the external memory.

D8 WBEMPTY: Write Buffer Empty Flag Bit

Indicates the write buffer status.

- 1 (R): Empty (default)
0 (R): Full

WBEMPTY is set to 0 by writing data to the write buffer and is set to 1 when the buffered data is read out for writing to the external memory. The write buffer improves writing speed as the CPU does not need to wait for completion of writing to the external memory. However, read the above flags to check if the data has been written to, especially when data is written to a low-speed external device.

D[7:0] Reserved

CCLK Division Ratio Select Register (CCU_CCLKDV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CCLK Division Ratio Select Register (CCU_CCLKDV)	0x302360 (32 bits)	D31–2 D1–0	– CLK_DOWN[1:0]	reserved CCLK division ratio select	– CLK_DOWN[1:0] Division ratio	– 0x0	– R/W	0 when being read. Source clock: MCLK
					0x3 1/8			
					0x2 1/4			
					0x1 1/2			
					0x0 1/1			

D[31:2] Reserved

D[1:0] CLK_DOWN[1:0]: CCLK Division Ratio Select Bits

Selects the division ratio to set the CCLK clock speed for operating the C33 PE Core and CCU. To reduce current consumption, operate the C33 PE Core and CCU using the slowest possible clock speed.

Table 11.7.4 CCLK Division Ratio Selection

CLK_DOWN[1:0]	Division ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

For more information on CCLK, see the “Clock Management Unit (CMU)” chapter.

12 Interrupt Controller (ITC)

12.1 ITC Module Overview

The interrupt controller (ITC) honors interrupt requests from the peripheral modules and outputs the interrupt request, interrupt level and vector number signals to the C33 PE Core according to the priority and interrupt levels. The features of the ITC module are listed below.

- Supports the following 27 maskable interrupt systems:
 - Port input interrupt (four systems)
 - DMAC interrupt (four systems)
 - 16-bit audio PWM timer (T16P) interrupt (one system)
 - 16-bit PWM timer (T16A5) interrupt (two systems)
 - LCDC interrupt (one systems)
 - 8-bit timer (T8) interrupt (four systems)
 - USI interrupt (one system)
 - USIL interrupt (one system)
 - FSIO interrupt (two systems)
 - A/D converter (ADC10) interrupt (one system)
 - RTC interrupt (one system)
 - Remote controller (REMC) interrupt (one system)
 - I²S interrupt (one system)
 - GE interrupt (two systems)
 - USB interrupt (one system)
- Supports seven interrupt levels (1 to 7) to prioritize the interrupt sources.

The ITC enables the interrupt level (priority) for determining the processing sequence when multiple interrupts occur simultaneously to be set for each interrupt system separately.

Each interrupt system includes one or more interrupt causes. Settings to enable or disable interrupts for different causes are performed by the respective peripheral module registers.

For specific information on interrupt causes and their control, refer to the descriptions of the peripheral module.

Figure 12.1.1 shows the structure of the interrupt system.

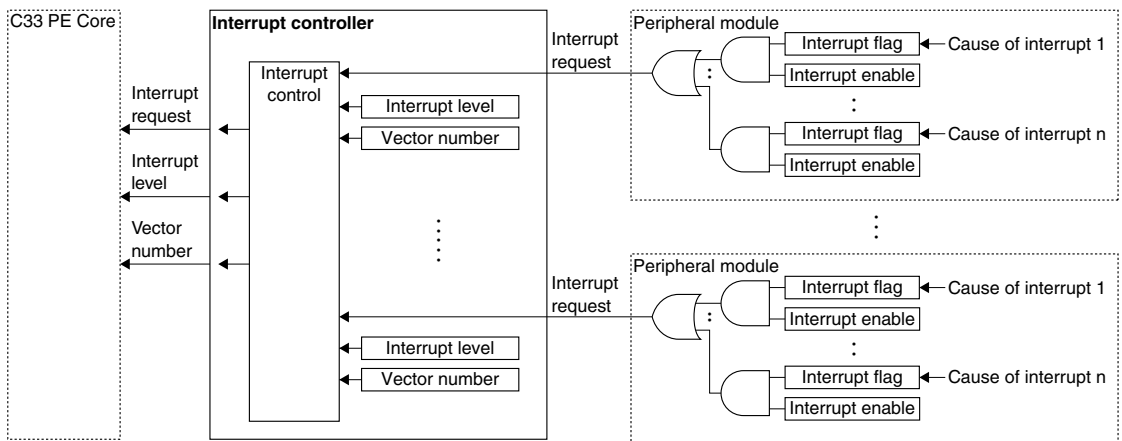


Figure 12.1.1 Interrupt System

12.2 Vector Table

The vector table contains the vectors to the interrupt handler routines (handler routine start address) that will be read by the C33 PE Core to execute the handler when an interrupt occurs.

Table 12.2.1 shows the vector table of the S1C33L26.

Table 12.2.1 Vector Table

Vector No.	Vector address	Interrupt name	Cause of interrupt	Priority
0 (0x0)	TTBR + 0x0	Reset	<ul style="list-style-type: none"> • Low input to the #RESET pin • Watchdog timer overflow *2 	1
1 (0x1)	TTBR + 0x4	Reserved	–	–
2 (0x2)	TTBR + 0x8	ext exception	ext instruction (illegal use)	4
3 (0x3)	TTBR + 0xc	Undefined instruction exception	Undefined instruction	3
4 (0x4)	TTBR + 0x10	Reserved	–	–
5 (0x5)	TTBR + 0x14			
6 (0x6)	TTBR + 0x18			
–	(0x60000)	Debugging exception	brk instruction, etc.	5
7 (0x7)	TTBR + 0x1c	NMI	<ul style="list-style-type: none"> • Low input to the #NMI pin • Watchdog timer overflow *2 	6
8 (0x8)	TTBR + 0x20	Reserved	–	–
9 (0x9)	TTBR + 0x24			
10 (0xa)	TTBR + 0x28			
11 (0xb)	TTBR + 0x2c			
12 (0xc)	TTBR + 0x30	Software exception 0	int instruction	High *1 ↑
13 (0xd)	TTBR + 0x34	Software exception 1	int instruction	
14 (0xe)	TTBR + 0x38	Software exception 2	int instruction	
15 (0xf)	TTBR + 0x3c	Software exception 3	int instruction	
16 (0x10)	TTBR + 0x40	Port input interrupt 0	FPT0–3 input (rising/falling edge or high/low level)	
17 (0x11)	TTBR + 0x44	Port input interrupt 1	FPT4–7 input (rising/falling edge or high/low level)	
18 (0x12)	TTBR + 0x48	Port input interrupt 2	FPT8–B input (rising/falling edge or high/low level)	
19 (0x13)	TTBR + 0x4c	Port input interrupt 3	FPTC–F input (rising/falling edge or high/low level)	
20 (0x14)	TTBR + 0x50	DMAC Ch.0/2 interrupt	End of DMA transfer	
21 (0x15)	TTBR + 0x54	DMAC Ch.1/3 interrupt	End of DMA transfer	
22 (0x16)	TTBR + 0x58	DMAC Ch.4/6 interrupt	End of DMA transfer	
23 (0x17)	TTBR + 0x5c	DMAC Ch.5/7 interrupt	End of DMA transfer	
24 (0x18)	TTBR + 0x60	16-bit audio PWM timer (T16P) interrupt	<ul style="list-style-type: none"> • Compare A/B • Buffer empty 	
25 (0x19)	TTBR + 0x64	16-bit PWM timer (T16A5) Ch.0 interrupt	<ul style="list-style-type: none"> • Compare A/B • Capture A/B • Capture A/B overwrite 	
26 (0x1a)	TTBR + 0x68	16-bit PWM timer (T16A5) Ch.1 interrupt	<ul style="list-style-type: none"> • Compare A/B • Capture A/B • Capture A/B overwrite 	
27 (0x1b)	TTBR + 0x6c	LCDC interrupt	Beginning of a frame	
28 (0x1c)	TTBR + 0x70	Reserved	–	
29 (0x1d)	TTBR + 0x74	8-bit timer (T8) Ch.0/4 interrupt	Timer underflow	
30 (0x1e)	TTBR + 0x78	8-bit timer (T8) Ch.1/5 interrupt	Timer underflow	
31 (0x1f)	TTBR + 0x7c	8-bit timer (T8) Ch.2/6 interrupt	Timer underflow	
32 (0x20)	TTBR + 0x80	8-bit timer (T8) Ch.3/7 interrupt	Timer underflow	
33 (0x21)	TTBR + 0x84	USI interrupt	<ul style="list-style-type: none"> • Transmit buffer empty • Receive buffer full • Receive error 	
34 (0x22)	TTBR + 0x88	FSIO Ch.0 interrupt	<ul style="list-style-type: none"> • Transmit buffer empty • Receive buffer full • Receive error 	
35 (0x23)	TTBR + 0x8c	A/D converter (ADC10) interrupt	<ul style="list-style-type: none"> • Conversion completion • Conversion result overwrite 	
36 (0x24)	TTBR + 0x90	RTC interrupt	1/512 second, 1/256 second, 1/128 second, 1/64 second, 1 second, 1 minute, or 1 hour cycles	
37 (0x25)	TTBR + 0x94	Reserved	–	
38 (0x26)	TTBR + 0x98	FSIO Ch.1 interrupt	<ul style="list-style-type: none"> • Transmit buffer empty • Receive buffer full • Receive error 	
39 (0x27)	TTBR + 0x9c	USIL interrupt	<ul style="list-style-type: none"> • Transmit buffer empty • Receive buffer full • Receive error 	

Vector No.	Vector address	Interrupt name	Cause of interrupt	Priority
40 (0x28)	TTBR + 0xa0	Remote controller (REMC) interrupt	<ul style="list-style-type: none"> • Data length counter underflow • Input rising edge detected • Input falling edge detected 	↓ Low *1
41 (0x29)	TTBR + 0xa4	I ² S interrupt	I ² S FIFO whole/half/one empty	
42 (0x2a)	TTBR + 0xa8	GE complete interrupt	End of command list execution	
43 (0x2b)	TTBR + 0xac	GE error interrupt	<ul style="list-style-type: none"> • Calculation error • Drawing error 	
44 (0x2c)	TTBR + 0xb0	USB interrupt	USB operation, bus and FIFO statuses	

*1 When the same interrupt level is set

*2 Either reset or NMI can be selected as the watchdog timer interrupt with software.

Vector numbers 16 to 44 are assigned to the maskable interrupts supported by the S1C33L26.

Vector table base address

The S1C33L26 allows the base (starting) address of the vector table to be set using the TTBR register. “TTBR” indicated in Table 12.2.1 means the value that is set to this register. Set the TTBR register in the initial routine executed after booting. Bits 9 to 0 of the TTBR register are fixed at 0. Therefore, the vector table starting address always begins with a 1 KB boundary address.

12.3 Control of Maskable Interrupts

12.3.1 Interrupt Control Bits in Peripheral Modules

The peripheral module that generates an interrupt includes an interrupt enable bit and an interrupt flag for each interrupt cause. The interrupt flag is set to 1 when the cause of interrupt occurs. By setting the interrupt enable bit to 1 (interrupt enabled), the flag state will be sent to the ITC as an interrupt request signal, generating an interrupt request to the C33 PE Core.

The corresponding interrupt enable bits should be set to 0 for those causes for which interrupts are not desired. In this case, although the interrupt flag is set to 1 if the interrupt cause occurs, the interrupt request signal sent to the ITC will not be asserted.

The interrupt flag set to 1 must be reset in the interrupt handler routine after the interrupt has occurred. The ITC will generate the same interrupt again once the interrupt handler routine has been ended by the reti instruction with the interrupt flag still set to 1, since it detects interrupt requests using the signal level.

For specific information on causes of interrupts, interrupt flags, and interrupt enable bits, refer to the respective peripheral module descriptions.

12.3.2 ITC Interrupt Request Processing

On receiving an interrupt signal from a peripheral module, the ITC sends the interrupt request, interrupt level, and vector number signals to the C33 PE Core.

Vector numbers are determined by the ITC internal hardware for each interrupt cause, as shown in Table 12.2.1.

The interrupt level is a value used by the C33 PE Core to compare with the IL bits (PSR). This interrupt level is used in the C33 PE Core to disable subsequently occurring interrupts with the same or lower level. (See Section 12.3.3.)

The default ITC settings are level 0 for all maskable interrupts. Interrupt requests are not accepted by the C33 PE Core if the level is 0.

The ITC includes control bits (INT_LV[2:0]/ITC_XXX_LV register) for selecting the interrupt level, and the level can be set to between 1 (low) and 7 (high) interrupt levels for each interrupt vector.

12 INTERRUPT CONTROLLER (ITC)

If interrupt requests are input to the ITC simultaneously from two or more peripheral modules, the ITC outputs the interrupt request with the highest priority to the C33 PE Core in accordance with the following conditions.

1. The interrupt with the highest interrupt level takes precedence.
2. If multiple interrupt requests are input with the same interrupt level, the interrupt with the lowest vector number takes precedence.

The other interrupts occurring at the same time are held until all interrupts with higher priority levels have been accepted by the C33 PE Core.

If an interrupt cause with higher priority occurs while the ITC is outputting an interrupt request signal to the C33 PE Core (before being accepted by the C33 PE Core), the ITC alters the vector number and interrupt level signals to the setting information on the more recent interrupt. The previously occurring interrupt is held. The held interrupt is canceled and no interrupt is generated if the interrupt flag in the peripheral module is reset with software.

Table 12.3.2.1 Interrupt Level Registers

Hardware interrupt	Interrupt level register
Port input interrupt 0 (FPT0–3 interrupt)	ITC_FPT03_LV (0x300210)
Port input interrupt 1 (FPT4–7 interrupt)	ITC_FPT47_LV (0x300211)
Port input interrupt 2 (FPT8–B interrupt)	ITC_FPT8B_LV (0x300212)
Port input interrupt 3 (FPTC–F interrupt)	ITC_FPTCF_LV (0x300213)
DMAC Ch.0/2 interrupt	ITC_DMA02_LV (0x300214)
DMAC Ch.1/3 interrupt	ITC_DMA13_LV (0x300215)
DMAC Ch.4/6 interrupt	ITC_DMA46_LV (0x300216)
DMAC Ch.5/7 interrupt	ITC_DMA57_LV (0x300217)
16-bit audio PWM timer (T16P) interrupt	ITC_T16P_LV (0x300218)
16-bit PWM timer (T16A5) Ch.0 interrupt	ITC_T16A0_LV (0x300219)
16-bit PWM timer (T16A5) Ch.1 interrupt	ITC_T16A1_LV (0x30021a)
LCDC interrupt	ITC_LCDC_LV (0x30021b)
8-bit timer (T8) Ch.0/4 interrupt	ITC_T804_LV (0x30021d)
8-bit timer (T8) Ch.1/5 interrupt	ITC_T815_LV (0x30021e)
8-bit timer (T8) Ch.2/6 interrupt	ITC_T826_LV (0x30021f)
8-bit timer (T8) Ch.3/7 interrupt	ITC_T837_LV (0x300220)
USI interrupt	ITC_USI_LV (0x300221)
FSIO Ch.0 interrupt	ITC_FSIO0_LV (0x300222)
A/D converter (ADC10) interrupt	ITC_ADC10_LV (0x300223)
RTC interrupt	ITC_RTC_LV (0x300224)
FSIO Ch.1 interrupt	ITC_FSIO1_LV (0x300226)
USIL interrupt	ITC_USIL_LV (0x300227)
Remote controller (REMC) interrupt	ITC_REMC_LV (0x300228)
I ² S interrupt	ITC_I2S_LV (0x300229)
GE complete interrupt	ITC_GECOM_LV (0x30022a)
GE error interrupt	ITC_GEERR_LV (0x30022b)
USB interrupt	ITC_USB_LV (0x30022c)

12.3.3 Interrupt Processing by the C33 PE Core

A Maskable interrupt to the C33 PE Core occurs when all of the following conditions are met:

- The interrupt is enabled by the interrupt control bit inside the peripheral module.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the C33 PE Core has been set to 1.
- The cause of interrupt that has occurred has a higher interrupt level than the value set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

If an interrupt cause that has been enabled in the peripheral module occurs, the corresponding interrupt flag is set to 1, and this state is maintained until it is reset by the program. This means that the interrupt cause is not cleared even if the conditions listed above are not met when the interrupt cause occurs. An interrupt occurs if the above conditions are met.

If multiple maskable interrupt causes occur simultaneously, the interrupt cause with the highest interrupt level and lowest vector number becomes the subject of the interrupt request to the C33 PE Core. Interrupts with lower levels are held until the above conditions are subsequently met.

The C33 PE Core samples interrupt requests for each cycle. On accepting an interrupt request, the C33 PE Core switches to interrupt processing immediately after execution of the current instruction has been completed.

Interrupt processing involves the following steps:

- (1) The PSR and current program counter (PC) values are saved to the stack.
- (2) The PSR IE bit is reset to 0 (disabling subsequent maskable interrupts).
- (3) The PSR IL bits are set to the received interrupt level. (The NMI does not affect the IL bits.)
- (4) The vector for the interrupt occurred is loaded to the PC to execute the interrupt handler routine.

When an interrupt is received, (2) prevents subsequent maskable interrupts. Setting the IE bit to 1 in the interrupt handler routine allows handling of multiple interrupts. In this case, since IL is changed by (3), only an interrupt with a higher level than that of the currently processed interrupt will be accepted. Ending interrupt handler routines using the `reti` instruction returns the PSR to the state before the interrupt has occurred. The program resumes processing following the instruction being executed at the time the interrupt occurred.

12.4 NMI

In the S1C33L26, a low level input to the #NMI pin or the watchdog timer can generate a non-maskable interrupt (NMI). The vector number for NMI is 7, with the vector address set to the vector table's starting address + 28 bytes. This interrupt takes precedence over other interrupts and is unconditionally accepted by the C33 PE Core.

For detailed information on generating NMI by the watchdog timer, see the “Watchdog Timer (WDT)” chapter.

12.5 Software Exception

A software exception can be generated by use of “`int imm2`” instruction of the C33 PE Core. A software exception number (0 to 3) is specified with `imm2` of the operand.

12.6 HALT and SLEEP Mode Cancellation

HALT mode is cleared by the following signals, which start the CPU.

- Interrupt request signal sent to the CPU from the ITC
- NMI signal output by the watchdog timer or input to the #NMI pin
- Debug interrupt signal
- Reset signal output by the watchdog timer or input to the #RESET pin

SLEEP mode is cleared by the following signals, which start the CPU.

- Port input interrupt or RTC interrupt request signal sent from the GPIO or RTC
- NMI signal input to the #NMI pin
- Reset signal input to the #RESET pin

Notes:

- If the CPU is able to receive interrupts when HALT or SLEEP mode has been cleared by an interrupt request for the CPU from the ITC, processing branches to the interrupt handler routine immediately after cancellation. In all other cases, the program is executed following the `halt` or `slp` instruction.

- HALT or SLEEP mode clearing due to interrupt requests cannot be masked (prohibited) using ITC interrupt level settings. When using a cause of interrupt to clear HALT or SLEEP mode, the interrupt enable bit corresponding to the cause of interrupt must be set to 1 (interrupt enabled).

For more information, see the “Power Saving” section in Appendix.

12.7 Control Register Details

Table 12.7.1 List of ITC Registers

Address	Register name		Function
0x300210	ITC_FPT03_LV	FPT0–3 Interrupt Level Register	Set FPT0–3 interrupt levels
0x300211	ITC_FPT47_LV	FPT4–7 Interrupt Level Register	Set FPT4–7 interrupt levels
0x300212	ITC_FPT8B_LV	FPT8–B Interrupt Level Register	Set FPT8–B interrupt levels
0x300213	ITC_FPTCF_LV	FPTC–F Interrupt Level Register	Set FPTC–F interrupt levels
0x300214	ITC_DMA02_LV	DMAC Ch.0 & 2 Interrupt Level Register	Set DMAC Ch.0 and 2 interrupt levels
0x300215	ITC_DMA13_LV	DMAC Ch.1 & 3 Interrupt Level Register	Set DMAC Ch.1 and 3 interrupt levels
0x300216	ITC_DMA46_LV	DMAC Ch.4 & 6 Interrupt Level Register	Set DMAC Ch.4 and 6 interrupt levels
0x300217	ITC_DMA57_LV	DMAC Ch.5 & 7 Interrupt Level Register	Set DMAC Ch.5 and 7 interrupt levels
0x300218	ITC_T16P_LV	T16P Interrupt Level Register	Set T16P interrupt level
0x300219	ITC_T16A0_LV	T16A5 Ch.0 Interrupt Level Register	Set T16A5 Ch.0 interrupt level
0x30021a	ITC_T16A1_LV	T16A5 Ch.1 Interrupt Level Register	Set T16A5 Ch.1 interrupt level
0x30021b	ITC_LCDC_LV	LCDC Interrupt Level Register	Set LCDC interrupt level
0x30021d	ITC_T804_LV	T8 Ch.0 & 4 Interrupt Level Register	Set T8 Ch.0 and 4 interrupt levels
0x30021e	ITC_T815_LV	T8 Ch.1 & 5 Interrupt Level Register	Set T8 Ch.1 and 5 interrupt levels
0x30021f	ITC_T826_LV	T8 Ch.2 & 6 Interrupt Level Register	Set T8 Ch.2 and 6 interrupt levels
0x300220	ITC_T837_LV	T8 Ch.3 & 7 Interrupt Level Register	Set T8 Ch.3 and 7 interrupt levels
0x300221	ITC_USI_LV	USI Interrupt Level Register	Set USI interrupt level
0x300222	ITC_FSIO0_LV	FSIO Ch.0 Interrupt Level Register	Set FSIO Ch.0 interrupt level
0x300223	ITC_ADC10_LV	ADC10 Interrupt Level Register	Set ADC10 interrupt level
0x300224	ITC_RTC_LV	RTC Interrupt Level Register	Set RTC interrupt level
0x300226	ITC_FSIO1_LV	FSIO Ch.1 Interrupt Level Register	Set FSIO Ch.1 interrupt level
0x300227	ITC_USIL_LV	USIL Interrupt Level Register	Set USIL interrupt level
0x300228	ITC_REMC_LV	REMC Interrupt Level Register	Set REMC interrupt level
0x300229	ITC_I2S_LV	I ² S Interrupt Level Register	Set I ² S interrupt level
0x30022a	ITC_GECOM_LV	GE Complete Interrupt Level Register	Set GE complete interrupt level
0x30022b	ITC_GEERR_LV	GE Error Interrupt Level Register	Set GE error interrupt level
0x30022c	ITC_USB_LV	USB Interrupt Level Register	Set USB interrupt level

The ITC registers are described in detail below. These are 8-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

Interrupt Level Registers (ITC_XXX_LV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Register (ITC_XXX_LV)	0x300210	D7–3	–	reserved	–	–	–	0 when being read.
	0x30022c	D2–0 (8 bits)	INT_LV[2:0]	Interrupt level	1 to 7	0x0	R/W	

D[7:3] Reserved

D[2:0] INT_LV[2:0]: Interrupt Level Bits

Sets the interrupt level (1 to 7). (Default: 0x0)

The C33 PE Core does not accept interrupts with a level set lower than the PSR IL value.

The ITC uses the interrupt level when multiple interrupt requests occur simultaneously.

If multiple interrupt requests enabled by the interrupt enable bit occur simultaneously, the ITC sends the interrupt request with the highest level set by the ITC_XXX_LV registers (0x300210 to 0x30022c) to the C33 PE Core.

If multiple interrupt requests with the same interrupt level occur simultaneously, the interrupt with the lowest vector number is processed first. The other interrupts are held until all interrupts of higher priority have been accepted by the C33 PE Core.

If an interrupt requests of higher priority occurs while the ITC outputs an interrupt request signal to the C33 PE Core (before acceptance by the C33 PE Core), the ITC alters the vector number and interrupt level signals to the setting details of the most recent interrupt. The immediately preceding interrupt is held.

Note: Make sure that the PSR IE bit is set to 0 before setting the interrupt level registers (ITC_XXX_LV).

13 DMA Controller (DMAC)

13.1 DMAC Module Overview

The S1C33L26 incorporates a DMA controller (DMAC) capable of controlling eight table DMA channels.

The table DMA transfers data according to the control information programmed in the RAM.

The following shows the features of the DMAC.

- Number of channels Maximum eight channels
- Control information Programmable in the DSTRAM, IVRAM (Area 3) or an external RAM (16-byte control information + 16-byte backup data per channel)
- Dual-address transfer IVRAM (Area 3), an external memory or an internal peripheral module area (from address 0x300000) can be specified as the transfer source and destination.
 1. Data transfer within IVRAM
 2. Data transfer between IVRAM and an external memory
 3. Data transfer between IVRAM and an internal peripheral module area
 4. Data transfer between an external memory and an external memory
 5. Data transfer between an external memory and an internal peripheral module area
 6. Data transfer within internal peripheral module area

* Note that Area 0 (including IVRAM relocated to Area 0) cannot be specified as the transfer source and destination.
- Transfer data size 8 bits, 16 bits or 32 bits
- 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, with 12-bit transfer counter)
- Transfer address control The source and/or destination addresses can be fixed or incremented in units of the transfer data size upon completion of transfer.
- Trigger 1. Software trigger via register control
2. Hardware trigger by interrupt source modules (USI, FSIO, I2S, T16P, T16A5, ADC10, USB)
- Pointer transfer Transfer data can be specified using the specified source as a pointer.
Transfer data = *(Pointer base address + *(Source address))
- Interrupt End-of-transfer interrupt
- Others - Auto-reload function for the identical DMA transfers without resetting
- A DMA pause (temporary standby) function of low-priority channel DMA by high-priority trigger

13.2 DMAC Operating Clock

The DMAC operates with BCLK supplied from the CMU. BCLK does not stop in normal mode and in HALT mode by default. It can be stopped in HALT mode using a CMU control register. For more information on BCLK, see the "Clock Management Unit (CMU)" chapter. In SLEEP mode, the CMU stops supplying BCLK to the DMAC.

13.3 Programming Control Information

The DMAC operates according to the transfer conditions specified with control information. The control information must be programmed in DSTRAM, IVRAM (Area 3) or an external RAM. The control information size is 4 words per channel. When using the auto-reload function, each channel needs additional 4 words for storing reload data (control information resetting data). The auto-reload function resets control information, which is updated during data transfer, with the reload data after a DMA transfer has finished. Therefore, a consecutive 256-byte space is required for the control table to use eight DMA channels.

The following explains how to set the base address of the control table and the contents of control information.

13.3.1 Setting the Base Address

The RAM area beginning with the specified base address is allocated to the control table. The base address is the start address of the control information for Ch.0 and can be specified using TBL_BASE[31:0]/DMAC_TBL_BASE register. TBL_BASE[9:0] is fixed at 0 regardless of the contents written, therefore the base address is always set to 1,024-byte boundary address. The initial value of TBL_BASE[31:0] is 0x80000 (DSTRAM).

Base + 0xf0	Ch.7 auto-reload data area
Base + 0xe0	Ch.7 control table
Base + 0xd0	Ch.6 auto-reload data area
Base + 0xc0	Ch.6 control table
Base + 0xb0	Ch.5 auto-reload data area
Base + 0xa0	Ch.5 control table
Base + 0x90	Ch.4 auto-reload data area
Base + 0x80	Ch.4 control table
Base + 0x70	Ch.3 auto-reload data area
Base + 0x60	Ch.3 control table
Base + 0x50	Ch.2 auto-reload data area
Base + 0x40	Ch.2 control table
Base + 0x30	Ch.1 auto-reload data area
Base + 0x20	Ch.1 control table
Base + 0x10	Ch.0 auto-reload data area
Base	Ch.0 control table

Figure 13.3.1.1 Control Table Map

Note: The control table must be placed on DSTRAM, IVRAM (Area 3) or an external RAM. IRAM (Area 0) and BBRAM cannot be used to store control information.

13.3.2 Control Information

The address to store control information is determined by the base address and a channel number.

Start address of channel = base address + (channel number × 32 [bytes])

Note: The control information must be written only when the channel to be configured does not start a DMA transfer. If a DMA transfer starts when the control information is being written, proper transfer cannot be performed. Reading the control information can always be performed.

The contents of control information in each channel are shown in the table below.

Table 13.3.2.1 Control Information

Word	Address	Bit	Name	Function	Setting	
1st word (32 bits)	Ch.0: Base + 0x0	D31–24	–	reserved	–	
	Ch.1: Base + 0x20	D23–12	TC[11:0]	Transfer counter	TC[11:0]	Count
	Ch.2: Base + 0x40				0xff	4,095
	Ch.3: Base + 0x60				:	:
	Ch.4: Base + 0x80				0x1	1
	Ch.5: Base + 0xa0				0x0	4,096
	Ch.6: Base + 0xc0					
Ch.7: Base + 0xe0	D11	ST	Source type	1 Pointer	0 Data	

Word	Address	Bit	Name	Function	Setting	
1st word (32 bits)	Ch.0: Base + 0x0 Ch.1: Base + 0x20 Ch.2: Base + 0x40 Ch.3: Base + 0x60 Ch.4: Base + 0x80 Ch.5: Base + 0xa0 Ch.6: Base + 0xc0 Ch.7: Base + 0xe0	D10–8	UNIT[2:0]	Transfer data unit	UNIT[2:0]	Data unit
					0x7–0x3	reserved
					0x2	32 bits
					0x1	16 bits
		D7–6	SRINC[1:0]	Source address control	SRINC[1:0]	Address
					0x3–0x2	reserved
		D5–4	DSINC[1:0]	Destination address control	DSINC[1:0]	Address
					0x3–0x2	reserved
D3	CHEN	Channel enable	1	Enable	0	Disable
D2	TM	Transfer mode	1	Successive	0	Single
D1	RELOAD	Auto-reload enable	1	Enable	0	Disable
D0	PTW	Pointer bit width	1	8 bits	0	16 bits
2nd word (32 bits)	Ch.0: Base + 0x4 Ch.1: Base + 0x24 Ch.2: Base + 0x44 Ch.3: Base + 0x64 Ch.4: Base + 0x84 Ch.5: Base + 0xa4 Ch.6: Base + 0xc4 Ch.7: Base + 0xe4	D31–0	SRADR[31:0]	Source address/source data pointer	0x0 to 0xffffffff	
3rd word (32 bits)	Ch.0: Base + 0x8 Ch.1: Base + 0x28 Ch.2: Base + 0x48 Ch.3: Base + 0x68 Ch.4: Base + 0x88 Ch.5: Base + 0xa8 Ch.6: Base + 0xc8 Ch.7: Base + 0xe8	D31–0	DSADR[31:0]	Destination address	0x0 to 0xffffffff	
4th word (32 bits)	Ch.0: Base + 0xc Ch.1: Base + 0x2c Ch.2: Base + 0x4c Ch.3: Base + 0x6c Ch.4: Base + 0x8c Ch.5: Base + 0xac Ch.6: Base + 0xcc Ch.7: Base + 0xec	D31–16	PTBASE [31:16]	Pointer base address (high-order 16 bits)	0x0 to 0xffff (PTBASE[31:0] = 0x0 to 0xffff0000)	
		D15–0	PTBASE [15:0]	Fix at 0 (Pointer base address low-order 16 bits)	0x0	
5th word (32 bits)	Ch.0: Base + 0x10 Ch.1: Base + 0x30 Ch.2: Base + 0x50 Ch.3: Base + 0x70 Ch.4: Base + 0x90 Ch.5: Base + 0xb0 Ch.6: Base + 0xd0 Ch.7: Base + 0xf0	D31–0	RELOAD0 [31:0]	Reload data 0	(Same contents as 1st word)	
6th word (32 bits)	Ch.0: Base + 0x14 Ch.1: Base + 0x34 Ch.2: Base + 0x54 Ch.3: Base + 0x74 Ch.4: Base + 0x94 Ch.5: Base + 0xb4 Ch.6: Base + 0xd4 Ch.7: Base + 0xf4	D31–0	RELOAD1 [31:0]	Reload data 1	(Same contents as 2nd word)	
7th word (32 bits)	Ch.0: Base + 0x18 Ch.1: Base + 0x38 Ch.2: Base + 0x58 Ch.3: Base + 0x78 Ch.4: Base + 0x98 Ch.5: Base + 0xb8 Ch.6: Base + 0xd8 Ch.7: Base + 0xf8	D31–0	RELOAD2 [31:0]	Reload data 2	(Same contents as 3rd word)	
8th word (32 bits)	Ch.0: Base + 0x1c Ch.1: Base + 0x3c Ch.2: Base + 0x5c Ch.3: Base + 0x7c Ch.4: Base + 0x9c Ch.5: Base + 0xbc Ch.6: Base + 0xdc Ch.7: Base + 0xfc	D31–0	RELOAD3 [31:0]	Reload data 3	(Same contents as 4th word)	

TC[11:0]: Transfer counter (D[23:12]/1st word)

Set the number of times for unit data transfers to be executed. Writing 0x1 to 0xfff sets the transfer count to 1 to 4,095 and writing 0x0 sets it to 4,096. After a transfer of the data unit specified in UNIT[2:0] is completed, the transfer counter is decremented.

ST: Source type (D11/1st word)

Selects whether the memory contents on the specified source address are used as data or pointers.

ST = 0: Data

The DMAC transfers the data stored in the source address to the destination address.

ST = 1: Pointer

The DMAC uses the specified source address as a pointer and determines the address in which transfer data is stored as follows:

Transfer data = *(Pointer base address + *(Source address))

Example: When base address (PTBASE[31:16]) = 0x85 (i.e., address 0x850000), source address (SRADR[31:0]) = 0xfc000, and the contents of address 0xfc000 = 0x2

The data stored in address 0x850002 is transferred.

UNIT[2:0]: Transfer data unit (D[10:8]/1st word)

Sets the data size for the transfer unit.

Table 13.3.2.2 Transfer Data Unit

UNIT[2:0]	Transfer data unit
0x7–0x3	Reserved
0x2	32 bits
0x1	16 bits
0x0	8 bits

SRINC[1:0]: Source address control (D[7:6]/1st word)

Sets the control method for the source address after a unit data transfer.

Table 13.3.2.3 Source Address Control

SRINC[1:0]	Source address control
0x3–0x2	Reserved
0x1	Increment
0x0	Fixed

SRINC[1:0] = 0x0: 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[1:0] = 0x1: Address increment

After a transfer of a data unit specified with UNIT[2:0] is completed, the source address is incremented for the transferred data unit. The address that has been incremented during transfer does not return to the initial value.

DSINC[1:0]: Destination address control (D[5:4]/1st word)

Sets the control method for the destination address after a unit data transfer.

Table 13.3.2.4 Destination Address Control

DSINC[1:0]	Destination address control
0x3–0x2	Reserved
0x1	Increment
0x0	Fixed

DSINC[1:0] = 0x0: 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[1:0] = 0x1: Address increment

After a transfer of a data unit specified with UNIT[2:0] is completed, the destination address is incremented for the transferred data unit. The address that has been incremented during transfer does not return to the initial value.

CHEN: Channel enable (D3/1st word)

Enables or disables DMA transfers in each channel.

CHEN = 0: Transfer disabled

DMA transfers in the channel with this bit set to 0 are disabled.

CHEN = 1: Transfer enabled

DMA transfers in the channel are enabled. When triggered in this status, the DMAC starts DMA transfers in that channel (unless the transfer pausing function is operating, triggered by a high-priority channel).

CHEN is cleared upon completion of the current transfer. When the RELOAD bit (D1/1st word) is set to 0, this disables subsequent DMA transfers. When RELOAD is 1, CHEN will be replaced with RELOAD03 (D3/reload data 0) by auto-reloading. Therefore, if RELOAD03 is set to 1, CHEN will be set to 1 again enabling the DMAC to accept the subsequent DMA trigger without setting via software.

TM: Transfer mode (D2/1st word)

Sets the transfer mode (single transfer mode, successive transfer mode).

TM = 0: Single transfer mode

In this mode, a transfer operation invoked by one trigger is completed after transferring one data unit of the size set in UNIT[2:0]. 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.

TM = 1: Successive transfer mode

In this mode, one trigger performs data transfer a number of times as set by the transfer counter. The transfer counter is decremented each time a unit data is transferred, and successive transfers end when the counter reaches 0.

RELOAD: Auto-reload enable (D1/1st word)

Enables or disables the auto-reload function. The auto-reload function resets the initial value of control information set in the auto-reload area (configured the same as the four words in the control table) in the control table as soon as the transfer counter reaches 0. With this function, you can execute transfers with the new conditions without resetting the initial value in the DMAC interrupt handler routine.

RELOAD = 0: Auto-reload disabled

Setting this bit to 0 disables the auto-reload function. Furthermore, the CHEN bit is set to 0 when the transfer counter reaches 0, disabling subsequent DMA transfers. The control table retains the contents when the transfer counter reaches 0.

RELOAD = 1: Auto-reload enabled

Setting this bit to 1 enables the auto-reload function. When the transfer counter reaches 0, the control information stored in the auto-reload area is reset on the control table. The next trigger executes DMA transfer with the new conditions set in the reload data.

Note: Control information (initial value) for the auto-reload area should be prepared in the application program.

PTW: Pointer bit width (D0/1st word)

Sets the pointer size when pointer is selected for the source type (ST).

- **PTW = 0: 16 bits**

DMAC performs a 16-bit read from the specified source address to obtain the pointer.

- **PTW = 1: 8 bits**

DMAC performs an 8-bit read from the specified source address to obtain the pointer.

13 DMA CONTROLLER (DMAC)

SRADR[31:0]: Source address (D[31:0]/2nd word)

Set the start address of the transfer source (or the pointer to the transfer source). This setting is updated according to the setting of SRINC[1:0].

DSADR[31:0]: Destination address (D[31:0]/3rd word)

Set the start address of the transfer destination. This setting is updated according to the setting of DSINC[1:0].

PTBASE[31:16]: Pointer base address (D[31:16]/4th word)

Set the pointer base address (see ST) when pointer is selected for the source type. The low-order 16 bits of the base address are fixed at 0x0 (D[15:0] is ignored).

13.3.3 Auto-Reload Data

As shown in Figure 13.3.1.1, a RAM area is allocated to an auto-reloading data area for each channel along with the control table. When the auto-reload function is enabled by setting RELOAD (D1/1st word) to 1, the contents of the auto-reloading data area will be reset on the control table when the transfer counter reaches 0, enabling transfers with the new conditions to be executed without setting of the conditions in the DMAC interrupt handler routine. The four words (32 bytes) in the auto-reloading area are handled as exactly the same bit configuration as the four words on the control table. The auto-reloading area can be used as a control information buffer. Before using a DMAC channel with the auto-reload function, write the first transfer conditions with RELOAD set to 1 to the control table and the second transfer conditions to the auto-reloading area. The control information written to the auto-reloading area is loaded to the control table upon completion of the first data transfer and it will control the second data transfer. If the auto-reload function is not used, the control table must be reset to the subsequent transfer conditions in the DMAC interrupt handler routine.

The address of the auto-reload data area can be calculated from the equation below.

$$\text{Start address of auto-reloading data area} = \text{base address} + (\text{channel number} \times 32) + 16$$

13.4 DMAC Invocation

The triggers by which DMA is invoked have the following two causes:

1. Software trigger via register control
2. Hardware trigger due to a cause of interrupt in internal peripheral modules

Enabling DMAC

Each DMAC channel enters ready-to-operate status by setting DMAON_x/DMAC_CH_EN register to 1. When DMAON_x is 0 (default), the DMA channel does not accept triggers even if the control information enables transfers.

Enabling DMA transfers

Writing 1 to the CHEN bit (D3/1st word) on the control table enables DMA transfers in that channel, making it ready to accept triggers.

DMAC invocation by a software trigger

Any DMAC channel can be invoked via software. In order to invoke DMA transfer using Ch._x, write 1 to TRG_x/DMAC_TRG_FLG register.

TRG_x retains 1 until the DMA request is accepted and then it is reset to 0 by the hardware. TRG_x is also set to 1 by a hardware trigger.

DMAC invocation by a cause of interrupt in internal peripheral modules

To respective channels of the DMAC, hardware trigger sources (causes of interrupt in peripheral modules) shown in Table 13.4.1 are assigned, which can be selected with the TRG_SEL_x[1:0]/DMAC_TRG_SEL register.

Table 13.4.1 DMAC Trigger Source

Channel	Control bits	Setting	Trigger source	Channel priority
Ch.7	TRG_SEL7[1:0]	0x3	A/D converter (ADC10) conversion completion	Low ↑
		0x2	Reserved	
		0x1	USIL transmit buffer empty	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.6	TRG_SEL6[1:0]	0x3	USB interrupt	
		0x2	Reserved	
		0x1	USIL receive buffer full	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.5	TRG_SEL5[1:0]	0x3	16-bit PWM timer (T16A5) Ch.x compare/capture A *	
		0x2	FSIO Ch.1 transmit buffer empty	
		0x1	Reserved	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.4	TRG_SEL4[1:0]	0x3	16-bit PWM timer (T16A5) Ch.x compare/capture B *	
		0x2	FSIO Ch.1 receive buffer full	
		0x1	Reserved	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.3	TRG_SEL3[1:0]	0x3	16-bit PWM timer (T16A5) Ch.x compare/capture A *	
		0x2	FSIO Ch.0 transmit buffer empty	
		0x1	USI transmit buffer empty	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.2	TRG_SEL2[1:0]	0x3	16-bit PWM timer (T16A5) Ch.x compare/capture B *	
		0x2	FSIO Ch.0 receive buffer full	
		0x1	USI receive buffer full	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.1	TRG_SEL1[1:0]	0x3	USB interrupt	
		0x2	Port input interrupt 0–3	
		0x1	I ² S R channel FIFO empty	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.0	TRG_SEL0[1:0]	0x3	A/D converter (ADC10) conversion completion	↓ High
		0x2	16-bit audio PWM timer (T16P) buffer empty	
		0x1	I ² S L channel FIFO empty	
		0x0	Hardware trigger disabled (software trigger only)	

* Set the T16A5 channel for invoking the DMAC using DMASEL[1:0]/T16A_CTLx register.

(Default: 0x0)

At initial reset, TRG_SELx[1:0] in all channels are set to 0x0 (hardware trigger disabled). Note that software triggers are enabled regardless of the trigger source selected.

These trigger sources (causes of interrupt) are used in common for interrupt requests and DMAC invocation requests. When interrupts due to the cause used for a trigger is enabled and the interrupt level is set to 1 or more, an interrupt is also generated simultaneously with the trigger for the DMAC. When an interrupt vector and handler routine are located in IRAM, interrupt handling can be executed even during a DMA transfer. An instruction for accessing the transfer source/destination is not executed until the DMA transfer is completed. When only invoking the DMAC and not using an interrupt, set the interrupt enable bit to 0 (interrupt disabled).

DMA request generated during a DMA transfer

A low-priority DMA request generated during a DMA transfer is not accepted until the transfer currently being executed is completed (until the unit data transfer is completed in the single transfer mode or until the transfer counter reaches 0 in the successive transfer mode).

A DMA request for another high-priority channel that is generated during successive transfers in a channel is accepted after the transfer of the current data unit is completed. The current DMA transfer is suspended at that point, and is resumed after that high-priority DMA transfer generated later is completed.

DMA request when the channel is disabled to transfer

Triggers are disabled for a channel with the CHEN bit (D3/1st word) set to 0 (DMA transfer disabled). TRGx for the channel will not be set.

13.5 Operation of DMAC

The DMAC has two transfer modes (single and successive transfer modes), in each of which data transfer operates differently. The following describes the operation in each transfer mode.

13.5.1 Single Transfer Mode

The channels for which TM (D2/1st word) in control information is set to 0 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 in UNIT[2:0]. If 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 in the single transfer mode is shown by the flow chart in Figure 13.5.1.1.

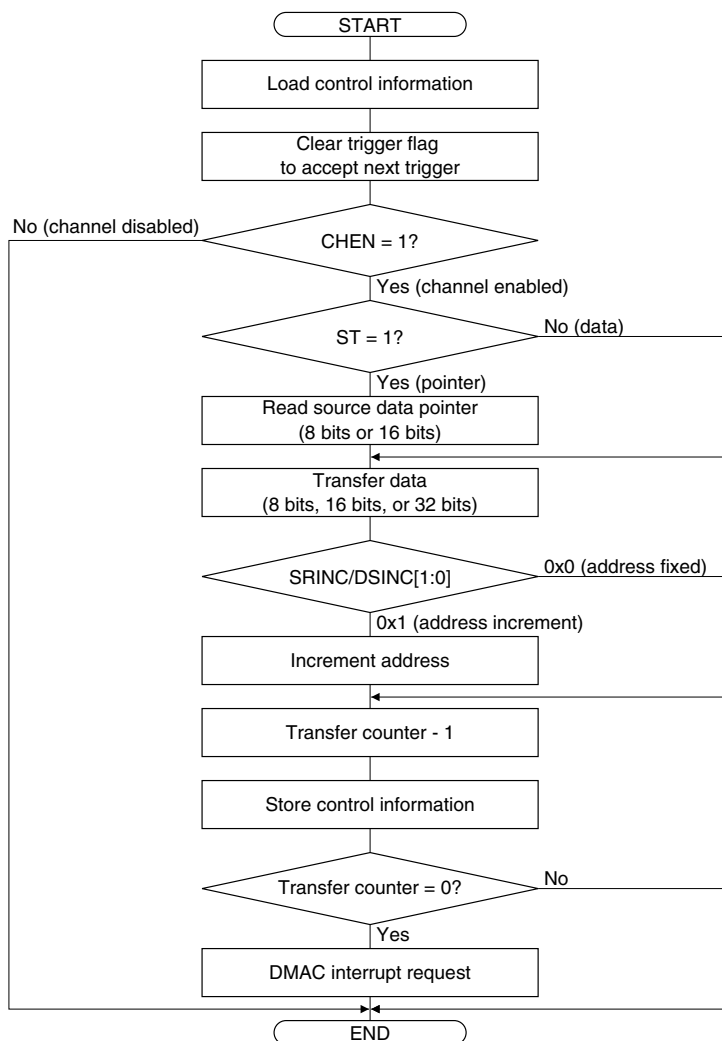


Figure 13.5.1.1 Operation Flow in Single Transfer Mode

- (1) When the DMAC accepts a trigger, it loads the control information of the channel into the DMAC module.
- (2) To allow the next trigger, the DMAC clears the trigger flag (TRGx/DMAC_TRG_FLG register).
- (3) The DMAC checks to see if CHEN is set to 1 (DMA transfer enabled). It abort data transfer if CHEN is set to 0.
- (4) If the source type specified in the control information is pointer (ST = 1), the DMAC read the contents of the specified source address to determine the pointer to the source data.
- (5) The DMAC reads the specified data unit from the source address into a buffer and then write it to the destination address.

The transfer status flag (RUNx/DMAC_RUN_STAT register) is set and retains 1 while data is being transferred.

- (6) According to the control information, the DMAC increments the source and/or destination addresses. The addresses are not changed if “address fixed” is specified. Also the transfer counter is decremented.
- (7) The DMAC writes the modified control information back to the control table.
- (8) The DMAC checks the transfer counter. If the value of the counter is not 0, the process is terminated here. Step (9) is not executed. Step (9) is executed if the transfer counter has reached 0.
- (9) The DMAC sets the end-of-transfer flag (ENDF_x/DMAC_END_FLG register) and clears the transfer status flag (RUN_x). If DMAIE_x/DMAC_IE register is set to 1 (end-of-transfer interrupt enabled), the DMAC outputs an interrupt request to the ITC.

This completes the single transfer process.

13.5.2 Successive Transfer Mode

The channels for which TM (D2/1st word) in control information is set to 1 operate in the 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 operation in the successive transfer mode is shown by the flow chart in Figure 13.5.2.1.

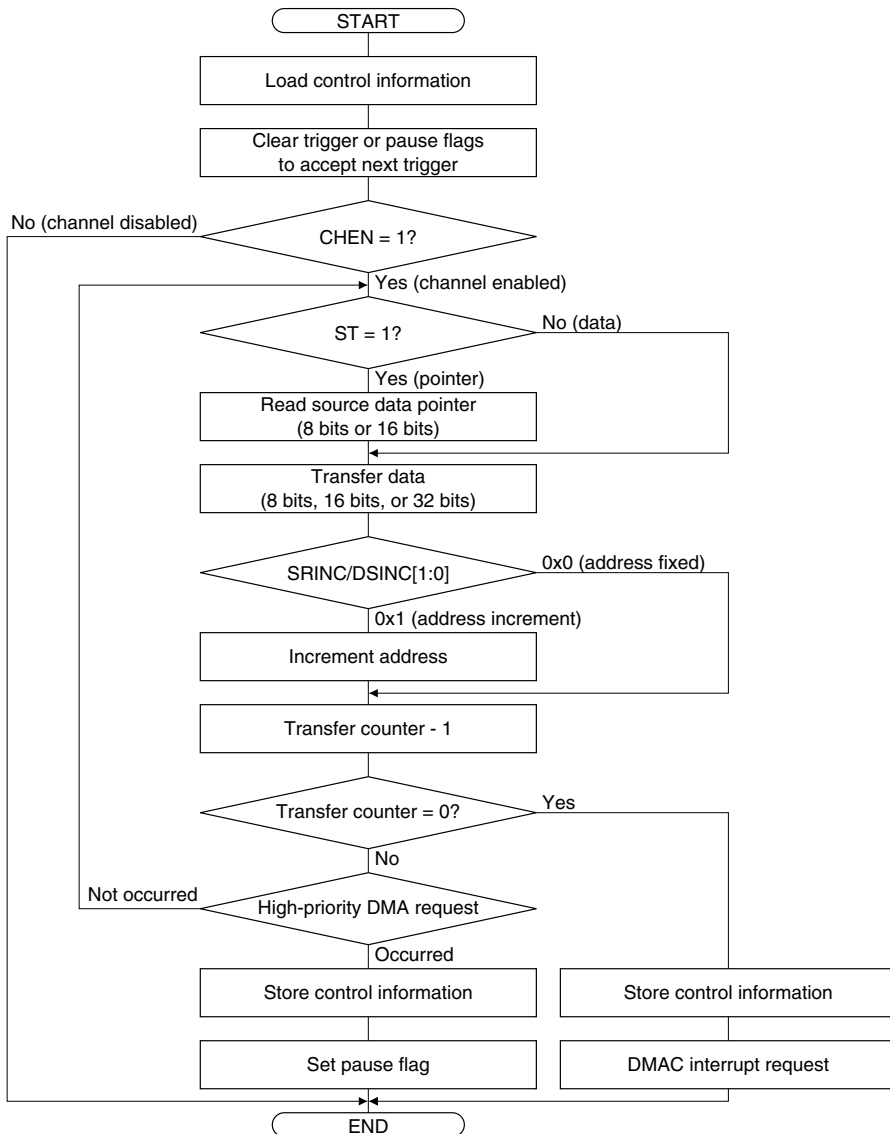


Figure 13.5.2.1 Operation Flow in Successive Transfer Mode

13 DMA CONTROLLER (DMAC)

- (1) When the DMAC accepts a trigger (or pause status is released), it loads the control information of the channel into the DMAC module.
- (2) To allow the next trigger, the DMAC clears the trigger flag (TRG_x/DMAC_TRG_FLG register) or the pause flag (PAUSE_x/DMAC_PAUSE_STAT register) according to the cause of the current DMA transfer.
- (3) The DMAC checks to see if CHEN is set to 1 (DMA transfer enabled). It abort data transfer if CHEN is set to 0.
- (4) If the source type specified in the control information is pointer (ST = 1), the DMAC read the contents of the specified source address to determine the pointer to the source data.
- (5) The DMAC reads the specified data unit from the source address into a buffer and then write it to the destination address.
The transfer status flag (RUN_x/DMAC_RUN_STAT register) is set and retains 1 while data is being transferred.
- (6) According to the control information, the DMAC increments the source and/or destination addresses. The addresses are not changed if “address fixed” is specified. Also the transfer counter is decremented.
- (7) The DMAC checks the transfer counter. It goes to Step (10) if the transfer counter has reaches 0.
- (8) The DMAC checks to see if any DMA request has been generated from other high-priority channels. If a high-priority trigger flag is set, the DMAC sets the pause flag (PAUSE_x) of the channel currently performing a transfer and suspends the transfer. The suspended DMA transfer will resume after other high-priority DMA transfers are completed.
- (9) If no DMA request has issued from high-priority channels, the DMAC returns to Step (4) to transfer the next data unit.
- (10) The DMAC sets the end-of-transfer flag (ENDF_x/DMAC_END_FLG register) and clears the transfer status flag (RUN_x). Also it writes the modified control information back to the control table. If DMAIE_x/DMAC_IE register is set to 1 (end-of-transfer interrupt enabled), the DMAC outputs an interrupt request to the ITC.

This completes the successive transfer process.

Suspending successive transfers due to other high-priority DMA request

Successive transfers can be temporarily suspended due to occurrence of a high-priority DMA request.

When a high-priority DMA request is generated, the channel performing a transfer saves control information required for resuming transfers (such as the current transfer count and the transfer source and destination addresses) as soon as the current data unit transfer is completed and then suspends transfers. At the same time, the pause flag (PAUSE_x/DMAC_PAUSE_STAT register) in the suspended channel is set.

After that, the higher-priority DMA transfer is executed. After the transfer is completed, the suspended DMA transfer is resumed by PAUSE_x that has been set. PAUSE_x is cleared when the DMA transfer is resumed.

- Notes:**
- Single transfers cannot be suspended.
 - The software triggered high-priority DMA request can not suspend any low-priority DMA transfer being performed.

13.6 DMAC Interrupt

The DMAC module includes a function for generating interrupts upon completion of a data transfer.

For the interrupts generated from trigger sources, see the descriptions of the peripheral modules.

End-of-transfer interrupt

This cause of interrupt occurs when a transfer in a channel has completed (when the transfer counter has reaches 0) and sets ENDF_x/DMAC_END_FLG register corresponding to the channel to 1.

To use this interrupt, set DMAIE_x/DMAC_IE register to 1. When DMAIE_x is set to 0 (default), interrupt requests for this interrupt cause are not sent to the interrupt controller (ITC).

If ENDF_x is set to 1 while DMAIE_x is set to 1 (interrupt enabled), the DMAC module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied.

To check the channel that has completed a data transfer, read ENDF_x in the interrupt handler routine.

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

- Notes:**
- To prevent interrupt recurrences, the DMAC module interrupt flag ENDF_x must be reset in the interrupt handler routine after a DMAC interrupt has occurred.
 - To prevent unwanted interrupts, ENDF_x should be reset before enabling DMAC interrupts with DMAIE_x. ENDF_x can be reset to 0 by writing 1.

13.7 Control Register Details

Table 13.7.1 List of DMAC Registers

Address	Register name		Function
0x302100	DMAC_CH_EN	DMAC Channel Enable Register	Enable DMAC channels
0x302104	DMAC_TBL_BASE	DMAC Control Table Base Address Register	Set control table base address
0x302108	DMAC_IE	DMAC Interrupt Enable Register	Enable/disable DMAC interrupts
0x30210c	DMAC_TRG_SEL	DMAC Trigger Select Register	Select trigger sources
0x302110	DMAC_TRG_FLG	DMAC Trigger Flag Register	Control software trigger and indicate trigger status
0x302114	DMAC_END_FLG	DMAC End-of-Transfer Flag Register	Indicate DMA completed channels
0x302118	DMAC_RUN_STAT	DMAC Running Status Register	Indicates running channel
0x30211c	DMAC_PAUSE_STAT	DMAC Pause Status Register	Indicate DMA suspended channels

The DMAC module registers are described in detail below. These are 32-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

DMAC Channel Enable Register (DMAC_CH_EN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
DMAC Channel Enable Register (DMAC_CH_EN)	0x302100 (32 bits)	D31–8	–	reserved	–	–	–	0 when being read.
		D7	DMAON7	DMAC Ch.7 enable	1 Enable 0 Disable	0	R/W	
		D6	DMAON6	DMAC Ch.6 enable	1 Enable 0 Disable	0	R/W	
		D5	DMAON5	DMAC Ch.5 enable	1 Enable 0 Disable	0	R/W	
		D4	DMAON4	DMAC Ch.4 enable	1 Enable 0 Disable	0	R/W	
		D3	DMAON3	DMAC Ch.3 enable	1 Enable 0 Disable	0	R/W	
		D2	DMAON2	DMAC Ch.2 enable	1 Enable 0 Disable	0	R/W	
		D1	DMAON1	DMAC Ch.1 enable	1 Enable 0 Disable	0	R/W	
		D0	DMAON0	DMAC Ch.0 enable	1 Enable 0 Disable	0	R/W	

D[31:8] Reserved

D[7:0] DMAON_x: DMAC Ch._x Enable Bit

Enables DMAC Ch._x to accept DMA triggers.

1 (R/W): Enabled

0 (R/W): Disabled/Forced termination (default)

To perform DMA transfer using DMAC Ch._x, write 1 to DMAON_x. When DMAON_x is 0, DMAC Ch._x does not accept triggers and data transfer cannot be started.

DMAC Control Table Base Address Register (DMAC_TBL_BASE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
DMAC Control Table Base Address Register (DMAC_TBL_BASE)	0x302104 (32 bits)	D31–10	TBL_BASE [31:10]	DMAC control table base address	0x0 to 0xffffc00 (1,024-byte boundary address within a RAM)	0x80000	R/W	
		D9–0	TBL_BASE [9:0]	Fixed at 0x0 (Cannot be altered.)			R	

D[31:0] TBL_BASE[31:0]: DMAC Control Table Base Address Bits

Sets a base address for the control table for writing control information and auto reload information.

The size of control information is 4 words (16 bytes) per channel. The area for auto-reloading also requires 4 words (16 bytes) per channel. Therefore, a consecutive 256-byte space is needed for the control table in order to support eight channels.

13 DMA CONTROLLER (DMAC)

The control table is secured in the RAM with the base address specified in these registers assumed to be the start address of the control information for Ch.0.

Since TBL_BASE[9:0] of this register is fixed at 0 regardless of the contents written, it is always set to 1,024-byte boundary address. The initial value of the register is 0x80000.

Base + 0xf0	Ch.7 auto-reload data area
Base + 0xe0	Ch.7 control table
Base + 0xd0	Ch.6 auto-reload data area
Base + 0xc0	Ch.6 control table
Base + 0xb0	Ch.5 auto-reload data area
Base + 0xa0	Ch.5 control table
Base + 0x90	Ch.4 auto-reload data area
Base + 0x80	Ch.4 control table
Base + 0x70	Ch.3 auto-reload data area
Base + 0x60	Ch.3 control table
Base + 0x50	Ch.2 auto-reload data area
Base + 0x40	Ch.2 control table
Base + 0x30	Ch.1 auto-reload data area
Base + 0x20	Ch.1 control table
Base + 0x10	Ch.0 auto-reload data area
Base	Ch.0 control table

Figure 13.7.1 Control Table Map

Note: The control table must be placed on DSTRAM, IVRAM (Area 3) or an external RAM. IRAM and BBRAM cannot be used to store control information.

DMAC Interrupt Enable Register (DMAC_IE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
DMAC Interrupt Enable Register (DMAC_IE)	0x302108 (32 bits)	D31-8	-	reserved	-	-	-	0 when being read.	
		D7	DMAIE7	DMAC Ch.7 interrupt enable	1 Enable	0 Disable	0	R/W	
		D6	DMAIE6	DMAC Ch.6 interrupt enable	1 Enable	0 Disable	0	R/W	
		D5	DMAIE5	DMAC Ch.5 interrupt enable	1 Enable	0 Disable	0	R/W	
		D4	DMAIE4	DMAC Ch.4 interrupt enable	1 Enable	0 Disable	0	R/W	
		D3	DMAIE3	DMAC Ch.3 interrupt enable	1 Enable	0 Disable	0	R/W	
		D2	DMAIE2	DMAC Ch.2 interrupt enable	1 Enable	0 Disable	0	R/W	
		D1	DMAIE1	DMAC Ch.1 interrupt enable	1 Enable	0 Disable	0	R/W	
		D0	DMAIE0	DMAC Ch.0 interrupt enable	1 Enable	0 Disable	0	R/W	

D[31:8] Reserved

D[7:0] DMAIE_x: DMAC Ch._x Interrupt Enable Bit

Enables or disables DMAC Ch._x interrupts.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Setting DMAIE_x to 1 enables the output of DMAC Ch._x interrupt requests to the ITC. Interrupts from Ch._x will not be generated if DMAIE_x is set to 0.

DMAC Trigger Select Register (DMAC_TRG_SEL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
DMAC Trigger Select Register (DMAC_TRG_SEL)	0x30210c (32 bits)	D31–16	–	reserved	–	–	–	0 when being read.
		D15–14	TRG_SEL7 [1:0]	Ch.7 trigger select	TRG_SEL7[1:0]	Trigger source 0x3 ADC complete 0x2 reserved 0x1 USIL Tx 0x0 No hard trigger	0x0	R/W
		D13–12	TRG_SEL6 [1:0]	Ch.6 trigger select	TRG_SEL6[1:0]	Trigger source 0x3 USB 0x2 reserved 0x1 USIL Rx 0x0 No hard trigger	0x0	R/W
		D11–10	TRG_SEL5 [1:0]	Ch.5 trigger select	TRG_SEL5[1:0]	Trigger source 0x3 T16A5 Ch.x A 0x2 FSIO Ch.1 Tx 0x1 reserved 0x0 No hard trigger	0x0	R/W
		D9–8	TRG_SEL4 [1:0]	Ch.4 trigger select	TRG_SEL4[1:0]	Trigger source 0x3 T16A5 Ch.x B 0x2 FSIO Ch.1 Rx 0x1 reserved 0x0 No hard trigger	0x0	R/W
		D7–6	TRG_SEL3 [1:0]	Ch.3 trigger select	TRG_SEL3[1:0]	Trigger source 0x3 T16A5 Ch.x A 0x2 FSIO Ch.0 Tx 0x1 USI Tx 0x0 No hard trigger	0x0	R/W
		D5–4	TRG_SEL2 [1:0]	Ch.2 trigger select	TRG_SEL2[1:0]	Trigger source 0x3 T16A5 Ch.x B 0x2 FSIO Ch.0 Rx 0x1 USI Rx 0x0 No hard trigger	0x0	R/W
		D3–2	TRG_SEL1 [1:0]	Ch.1 trigger select	TRG_SEL1[1:0]	Trigger source 0x3 USB 0x2 Port 0x1 I ² S R 0x0 No hard trigger	0x0	R/W
		D1–0	TRG_SEL0 [1:0]	Ch.0 trigger select	TRG_SEL0[1:0]	Trigger source 0x3 ADC complete 0x2 T16P 0x1 I ² S L 0x0 No hard trigger	0x0	R/W

D[31:16] Reserved

D[15:0] TRG_SELx[1:0]: Ch.x Trigger Select Bits

Selects a trigger source for each DMAC channel.

Table 13.7.2 DMAC Trigger Source

Channel	Control bits	Setting	Trigger source	Channel priority
Ch.7	TRG_SEL7[1:0]	0x3	A/D converter (ADC10) conversion completion	Low ↑
		0x2	Reserved	
		0x1	USIL transmit buffer empty	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.6	TRG_SEL6[1:0]	0x3	USB interrupt	
		0x2	Reserved	
		0x1	USIL receive buffer full	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.5	TRG_SEL5[1:0]	0x3	16-bit PWM timer (T16A5) Ch.x compare/capture A *	
		0x2	FSIO Ch.1 transmit buffer empty	
		0x1	Reserved	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.4	TRG_SEL4[1:0]	0x3	16-bit PWM timer (T16A5) Ch.x compare/capture B *	
		0x2	FSIO Ch.1 receive buffer full	
		0x1	Reserved	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.3	TRG_SEL3[1:0]	0x3	16-bit PWM timer (T16A5) Ch.x compare/capture A *	
		0x2	FSIO Ch.0 transmit buffer empty	
		0x1	USI transmit buffer empty	
		0x0	Hardware trigger disabled (software trigger only)	

13 DMA CONTROLLER (DMAC)

Channel	Control bits	Setting	Trigger source	Channel priority
Ch.2	TRG_SEL2[1:0]	0x3	16-bit PWM timer (T16A5) Ch.x compare/capture B *	↓ High
		0x2	FSIO Ch.0 receive buffer full	
		0x1	USI receive buffer full	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.1	TRG_SEL1[1:0]	0x3	USB interrupt	
		0x2	Port input interrupt 0–3	
		0x1	I ² S R channel FIFO empty	
		0x0	Hardware trigger disabled (software trigger only)	
Ch.0	TRG_SEL0[1:0]	0x3	A/D converter (ADC10) conversion completion	
		0x2	16-bit audio PWM timer (T16P) buffer empty	
		0x1	I ² S L channel FIFO empty	
		0x0	Hardware trigger disabled (software trigger only)	

* Set the T16A5 channel for invoking the DMAC using DMASEL[1:0]/T16A_CTLx register. (Default: 0x0)

At initial reset, TRG_SELx[1:0] in all channels are set to 0x0 (hardware trigger disabled). Note that software triggers are enabled regardless of the trigger source selected.

DMAC Trigger Flag Register (DMAC_TRG_FLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
DMAC Trigger Flag Register (DMAC_TRG_FLG)	0x302110 (32 bits)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	TRG7	Ch.7 software trigger/trigger status	1 (W)	0 (W)	0	R/W	
		D6	TRG6	Ch.6 software trigger/trigger status			0	R/W	
		D5	TRG5	Ch.5 software trigger/trigger status	Soft trigger	Ignored	0	R/W	
		D4	TRG4	Ch.4 software trigger/trigger status			0	R/W	
		D3	TRG3	Ch.3 software trigger/trigger status	(R)	Not triggered	0	R/W	
		D2	TRG2	Ch.2 software trigger/trigger status			0	R/W	
		D1	TRG1	Ch.1 software trigger/trigger status	Triggered	Not triggered	0	R/W	
		D0	TRG0	Ch.0 software trigger/trigger status			0	R/W	

D[31:8] Reserved

D[7:0] TRGx: Ch.x Software Trigger/Trigger Status Bit

Invokes a DMA of the specified channel by software trigger. Also indicates trigger status in respective channels, including hardware trigger.

1 (W): Software trigger

0 (W): Ignored

1 (R): Triggered

0 (R): Not triggered (default)

To use software trigger to start a Ch.x DMA transfer, write 1 to TRGx. In the case of a hardware trigger, the DMA transfer starts after TRGx is set to 1.

Among DMAC channels, Ch.0 is assigned the highest priority, which goes down in the ascending order of channels numbers. Therefore, when there are multiple settings of TRGx, channels with lower channel numbers are processed before the higher-number channels. Lower-priority channels are kept pending until all DMA transfers in higher-priority channels are completed, and TRGx also retains 1. The above applies to cases where another trigger is generated during a DMA transfer. That is, regardless of the order of trigger generation, a DMA request from the highest-priority channel is accepted as soon as the current DMA transfer is completed or suspended.

After the DMAC accepts a trigger, the DMA transfer of the channel starts. At the same time TRGx is cleared, allowing the channel to be re-triggered.

Note that acceptance of the trigger does not start a DMA transfer if CHEN (D3/1st word) in control information is set to 0.

If DMAONx/DMAC_CH_EN register is set to 0 (forced termination), TRGx that has been set is cleared and the pending DMA request is canceled.

DMAC End-of-Transfer Flag Register (DMAC_END_FLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
DMAC End-of-Transfer Flag Register (DMAC_END_FLG)	0x302114 (32 bits)	D31-8	–	reserved	–	–	–	0 when being read.	
		D7	ENDF7	Ch.7 end-of-transfer flag	1 Finished	0 Not finished	0	R/W	Reset by writing 1.
		D6	ENDF6	Ch.6 end-of-transfer flag	1 Finished	0 Not finished	0	R/W	
		D5	ENDF5	Ch.5 end-of-transfer flag	1 Finished	0 Not finished	0	R/W	
		D4	ENDF4	Ch.4 end-of-transfer flag	1 Finished	0 Not finished	0	R/W	
		D3	ENDF3	Ch.3 end-of-transfer flag	1 Finished	0 Not finished	0	R/W	
		D2	ENDF2	Ch.2 end-of-transfer flag	1 Finished	0 Not finished	0	R/W	
		D1	ENDF1	Ch.1 end-of-transfer flag	1 Finished	0 Not finished	0	R/W	
		D0	ENDF0	Ch.0 end-of-transfer flag	1 Finished	0 Not finished	0	R/W	

D[31:8] Reserved

D[7:0] ENDFx: Ch.x End-of-Transfer Flag Bit

Indicates the channel that has finished transfers.

- 1 (R): Finished
- 0 (R): Not finished (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

If the transfer counter in DMA transfer reaches 0, the DMAC sets ENDF_x indicating that transfers are finished. At the same time, an interrupt request is output to the ITC if DMAIE_x/DMAC_IE is set to 1 (interrupt enabled).

Read this register in the DMAC interrupt handler routine and check which channel has finished transfers. Also, in preparation for next interrupts, write 1 to ENDF_x for resetting it.

In a channel with DMAIE_x is set to 0 (interrupt disabled), an interrupt is not generated even if ENDF_x is set.

DMAC Running Status Register (DMAC_RUN_STAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
DMAC Running Status Register (DMAC_RUN_STAT)	0x302118 (32 bits)	D31-8	–	reserved	–	–	–	0 when being read.	
		D7	RUN7	Ch.7 running status	1 Running	0 Idle/paused	0	R	
		D6	RUN6	Ch.6 running status	1 Running	0 Idle/paused	0	R	
		D5	RUN5	Ch.5 running status	1 Running	0 Idle/paused	0	R	
		D4	RUN4	Ch.4 running status	1 Running	0 Idle/paused	0	R	
		D3	RUN3	Ch.3 running status	1 Running	0 Idle/paused	0	R	
		D2	RUN2	Ch.2 running status	1 Running	0 Idle/paused	0	R	
		D1	RUN1	Ch.1 running status	1 Running	0 Idle/paused	0	R	
		D0	RUN0	Ch.0 running status	1 Running	0 Idle/paused	0	R	

D[31:8] Reserved

D[7:0] RUNx: Ch.x Running Status Bit

Indicates whether the channel is performing a DMA transfer or not.

- 1 (R): Performing a DMA transfer
- 0 (R): Idle/paused (default)

RUN_x is set to 1 when DMAC Ch._x starts a DMA transfer and reset to 0 upon completion of the transfer operation. Also this bit reverts to 0 when the transfer is suspended due to a high-priority DMA request.

When modifying control information after a data transfer or forced termination, check this bit to ensure that the transfer operation is actually completed.

DMAC Pause Status Register (DMAC_PAUSE_STAT)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
DMAC Pause Status Register (DMAC_PAUSE_STAT)	0x30211c (32 bits)	D31-8	-	reserved	-			-	-	0 when being read.	
		D7	PAUSE7	Ch.7 pause status	1	Paused	0	Not paused	0	R	
		D6	PAUSE6	Ch.6 pause status	1	Paused	0	Not paused	0	R	
		D5	PAUSE5	Ch.5 pause status	1	Paused	0	Not paused	0	R	
		D4	PAUSE4	Ch.4 pause status	1	Paused	0	Not paused	0	R	
		D3	PAUSE3	Ch.3 pause status	1	Paused	0	Not paused	0	R	
		D2	PAUSE2	Ch.2 pause status	1	Paused	0	Not paused	0	R	
		D1	PAUSE1	Ch.1 pause status	1	Paused	0	Not paused	0	R	
		D0	PAUSE0	Ch.0 pause status	1	Paused	0	Not paused	0	R	

D[31:8] Reserved**D[7:0] PAUSEx: Ch.x Paused Status Bit**

Indicates whether the successive transfer operation is suspended due to a high-priority DMA transfer or not.

1 (R): Suspended

0 (R): Status other than suspension (default)

When a DMA request is generated that has higher priority than that of the channel in operation, the channel performing a transfer saves control information required for resuming transfers (such as the current transfer count and the transfer source and destination addresses) as soon as the current data transfer is completed and then suspends transfers. In this case, PAUSE_x is also set to 1, indicating that the channel has suspended a transfer. After that, the high-priority DMA transfer is executed. After the transfer is completed, suspended DMA transfers are resumed. At this time, the DMAC checks PAUSE_x and TRG_x/DMAC_TRG_FLG register, and processes the channels with their bits set, starting with one with the highest-priority (with the channel with the lowest number).

When the DMAC resumes DMA transfers that have been suspended, PAUSE_x is cleared.

14 8-bit Timers (T8)

14.1 T8 Module Overview

The S1C33L26 incorporates an eight-channel 8-bit timer module (T8).

The features of T8 are listed below.

- 8-bit presetable down counter with an 8-bit reload data register for setting the preset value
- The count clock is selectable from 15 clocks output from the prescaler.
- Generates the USI/USIL operating clocks (transfer clock source) and A/D trigger signal from the counter underflow signals.
- Generates an underflow interrupt signal to the interrupt controller (ITC).
- Any desired time intervals and serial transfer rates can be programmed by selecting an appropriate count clock and preset value.
- Ch.0 to Ch.3 support fine mode to minimize transfer rate errors.

Figure 14.1.1 shows the T8 configuration.

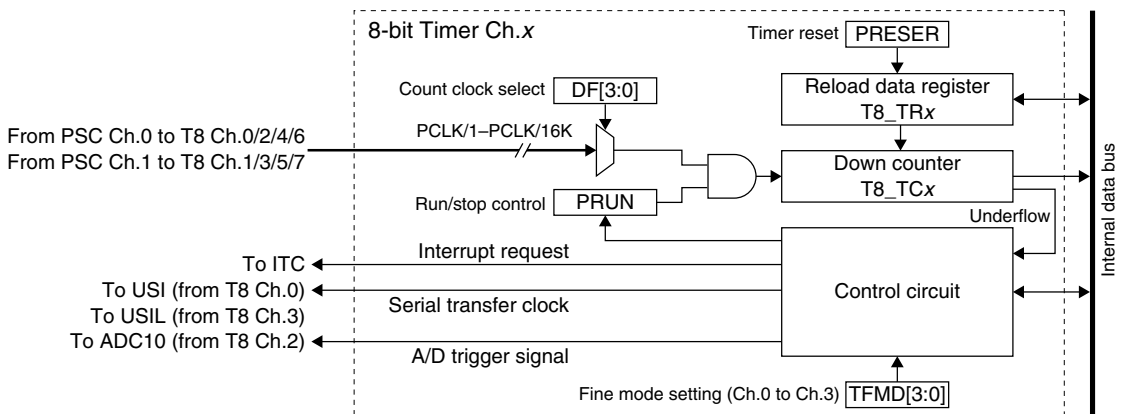


Figure 14.1.1 T8 Configuration (one channel)

T8 consists of an 8-bit presetable down counter and an 8-bit reload data register holding the preset value. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signals are used to generate an interrupt, USI clocks, and an A/D trigger signal. The underflow cycle can be programmed by selecting the prescaler clock and reload data, enabling the application program to obtain time intervals and serial transfer speeds as required. Ch.0 to Ch.3 support fine mode to minimize transfer rate errors.

Note: Eight channels of T8 module have the same functions except for the control register addresses and fine mode control bit. The description in this chapter applies to all channels otherwise unless specified. The 'x' in the register name refers to the channel number (0 to 7).

Example: T8_CTLx register

Ch.0: T8_CTL0 register

Ch.1: T8_CTL1 register

Ch.2: T8_CTL2 register

:

Ch.7: T8_CTL7 register

14.2 Count Clock

The count clock is selected by DF[3:0]/T8_CLKx register from the 15 types generated by the prescaler dividing the PCLK* clock into 1/1 to 1/16K.

* T8 Ch.0, 2, 4, and 6 use the PSC Ch.0 output clocks generated from PCLK1. T8 Ch.1, 3, 5, and 7 use the PSC Ch.1 output clocks generated from PCLK2. The descriptions in this chapter use PCLK as PCLK1 and PCLK2.

Table 14.2.1 Count Clock (PCLK Division Ratio) Selection

DF[3:0]	Division ratio	DF[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

- Notes:**
- The prescaler must run before T8 can operate.
 - Make sure the counter is halted before setting the count clock.

For detailed information on the prescaler control, see the “Prescaler (PSC)” chapter.

14.3 Count Mode

T8 features two count modes: repeat mode and one-shot mode. These modes are selected using TRMD/T8_CTLx register.

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets T8 to repeat mode.

In this mode, once the count starts, the timer continues running until stopped by the application program. When the counter underflows, the timer presets the reload data register value into the counter and continues the count. Thus, the timer periodically outputs an underflow pulse. T8 should be set to this mode to generate periodic interrupts or A/D triggers at desired intervals or to generate a serial transfer clock.

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets T8 to one-shot mode.

In this mode, the timer stops automatically as soon as the counter underflows. This means only one interrupt can be generated after the timer starts. Note that the timer presets the reload data register value to the counter, then stops after an underflow has occurred. T8 should be set to this mode to set a specific wait time.

Note: Make sure the counter is halted before setting the count mode.

14.4 Reload Data Register and Underflow Cycle

The reload data register T8_TRx is used to set the initial value for the down counter.

The initial counter value set in the reload data register is preset to the down counter if T8 is reset or the counter underflows. If T8 is started after resetting, the timer counts down from the reload value (initial value). This means that the reload value and the input clock frequency determine the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the specified wait time, the intervals between periodic interrupts or A/D triggers, and the programmable serial interface transfer clock.

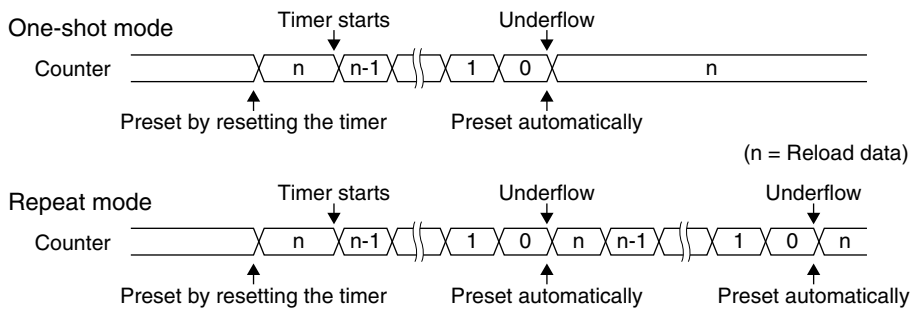


Figure 14.4.1 Preset Timing

The underflow cycle can be calculated as follows:

$$\text{Underflow interval} = \frac{\text{TR} + 1}{\text{clk_in}} [\text{s}] \quad \text{Underflow cycle} = \frac{\text{clk_in}}{\text{TR} + 1} [\text{Hz}]$$

clk_in: Count clock (prescaler output clock) frequency [Hz]

TR: Reload data (0–255)

14.5 Timer Reset

T8 is reset by writing 1 to PRESER/T8_CTLx register. The reload data is preset and the counter is initialized.

14.6 RUN/STOP Control

Make the following settings before starting T8.

- (1) Select the count clock (prescaler output clock). See Section 14.2.
- (2) Set the count mode (one-shot or repeat). See Section 14.3.
- (3) Calculate the initial counter value and set it to the reload data register. See Section 14.4.
- (4) Reset the timer to preset the counter to the initial value. See Section 14.5.
- (5) When using timer interrupts, set the interrupt level and enable interrupts for the relevant timer channel. See Section 14.9.

To start T8, write 1 to PRUN/T8_CTLx register.

The timer starts counting down from the initial value or from the current counter value if no initial value was preset. When the counter underflows, the timer outputs an underflow pulse and presets the counter to the initial value. An interrupt request is sent simultaneously to the interrupt controller (ITC).

If one-shot mode is set, the timer stops counting.

If repeat mode is set, the timer continues counting from the reloaded initial value.

Write 0 to PRUN to stop T8 via the application program. The counter stops counting and retains the current counter value until either the timer is reset or restarted. To restart the count from the initial value, the timer should be reset before writing 1 to PRUN.

When the timer is reset during running, the timer loads the reload register value to the counter and continues counting.

14 8-BIT TIMERS (T8)

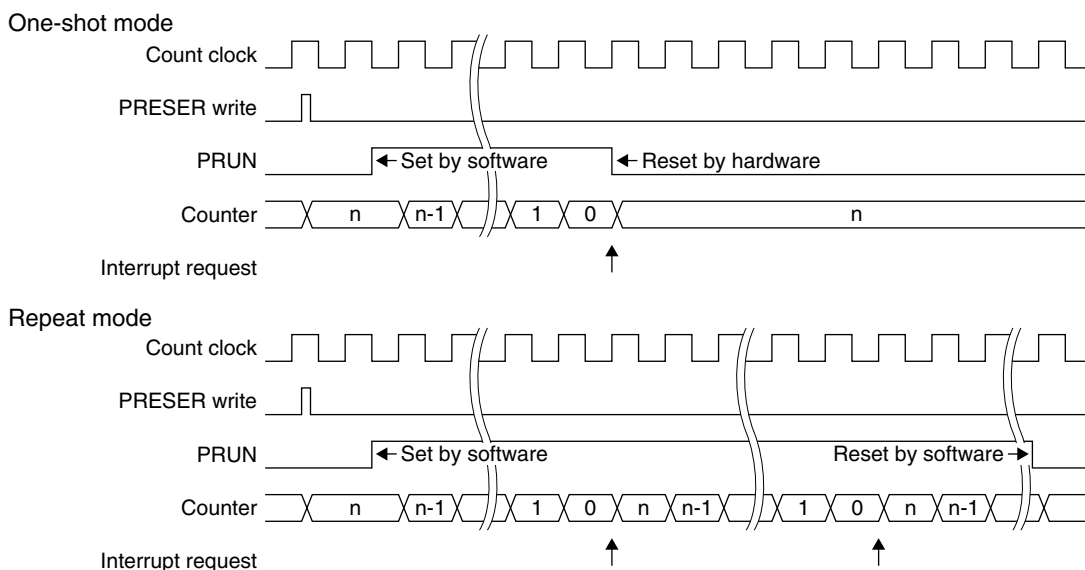


Figure 14.6.1 Count Operation

14.7 T8 Output Signals

T8 outputs underflow pulses when the counter underflows.

These pulses are used for timer interrupt requests.

These pulses are also used to generate the serial transfer clock for the internal serial interface or the A/D trigger signal.

The clock generated is sent to the internal peripheral module, as shown below.

T8 Ch.0 output clock → USI

T8 Ch.3 output clock → USIL

T8 Ch.2 output clock → A/D converter

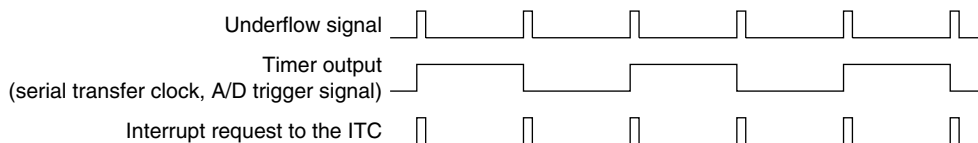


Figure 14.7.1 Timer Output Clock

14.8 Fine Mode (Ch.0 to Ch.3)

Ch.0 to Ch.3 support fine mode to minimize transfer rate errors.

T8 can output a programmable clock signal for use as the USI serial transfer clock. The timer output clock can be set to the required frequency by selecting the appropriate prescaler output clock and reload data. Note that errors may occur, depending on the transfer rate. Fine mode extends the output clock cycle by delaying the underflow pulse from the counter. This delay can be specified with the TFMD[3:0]/T8_CTLx register.

TFMD[3:0] specifies the delay pattern to be inserted into a 16 underflow period. Inserting one delay extends the output clock cycle by one count clock cycle. This setting delays the interrupt timing in the same way.

Table 14.8.1 Delay Patterns Specified by TFMD[3:0]

TFMD[3:0]	Underflow number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0x0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	D
0x2	-	-	-	-	-	-	-	D	-	-	-	-	-	-	-	D
0x3	-	-	-	-	-	-	-	D	-	-	-	D	-	-	-	D
0x4	-	-	-	D	-	-	-	D	-	-	-	D	-	-	-	D
0x5	-	-	-	D	-	-	-	D	-	-	-	D	-	D	-	D
0x6	-	-	-	D	-	D	-	D	-	-	-	D	-	D	-	D
0x7	-	-	-	D	-	D	-	D	-	D	-	D	-	D	-	D
0x8	-	D	-	D	-	D	-	D	-	D	-	D	-	D	-	D
0x9	-	D	-	D	-	D	-	D	-	D	-	D	-	D	D	D
0xa	-	D	-	D	-	D	D	D	-	D	-	D	-	D	D	D
0xb	-	D	-	D	-	D	D	D	-	D	D	D	-	D	D	D
0xc	-	D	D	D	-	D	D	D	-	D	D	D	-	D	D	D
0xd	-	D	D	D	-	D	D	D	-	D	D	D	D	D	D	D
0xe	-	D	D	D	D	D	D	D	-	D	D	D	D	D	D	D
0xf	-	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

D: Indicates the insertion of a delay cycle.

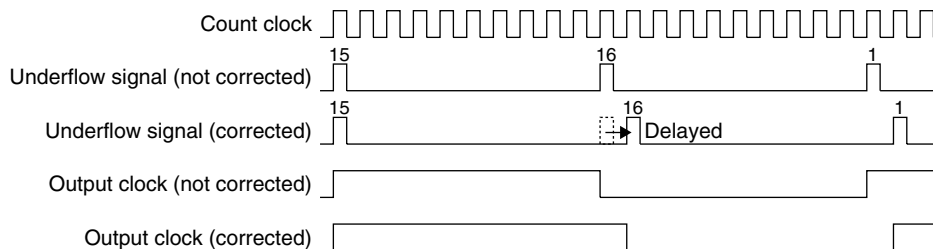


Figure 14.8.1 Delay Cycle Insertion in Fine Mode

At initial reset, TFMD[3:0] is set to 0x0, preventing insertion of delay cycles.

14.9 T8 Interrupts

T8 outputs an interrupt request to the interrupt controller (ITC) when the counter underflows.

Timer underflow interrupt

When the counter underflows, the interrupt flag T8IF/T8_INTx register, which is provided for each channel in the T8 module, is set to 1. At the same time, an interrupt request is sent to the ITC if T8IE/T8_INTx register has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied.

If T8IE is set to 0 (interrupt disabled, default), no interrupt request will be sent to the ITC.

For specific information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

- Notes:**
- The T8 module interrupt flag T8IF must be reset in the interrupt handler routine after a T8 interrupt has occurred to prevent recurring interrupts.
 - Reset T8IF before enabling T8 interrupts with T8IE to prevent occurrence of unwanted interrupt. T8IF is reset by writing 1.

14 8-BIT TIMERS (T8)

- T8 outputs four interrupt signals for Ch.0/4, Ch.1/5, Ch.2/6, and Ch.3/7 to the ITC. When using T8 interrupts, read out the interrupt flag in the T8 module as part of the interrupt handler routine and check which channel generates the interrupt.

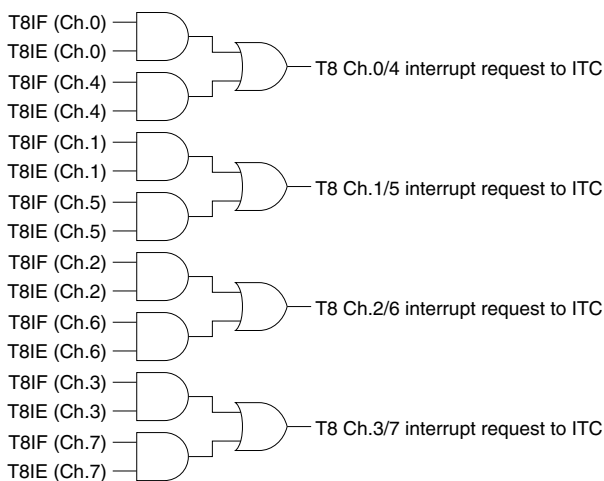


Figure 14.9.1 T8 Interrupt Circuit

14.10 Control Register Details

Table 14.10.1 List of 8-bit timer Registers

Address	Register name		Function
0x301100	T8_CLK0	T8 Ch.0 Input Clock Select Register	Select prescaler output clock
0x301102	T8_TR0	T8 Ch.0 Reload Data Register	Set reload data
0x301104	T8_TC0	T8 Ch.0 Counter Data Register	Counter data
0x301106	T8_CTL0	T8 Ch.0 Control Register	Set timer mode and start/stop timer
0x301108	T8_INT0	T8 Ch.0 Interrupt Control Register	Control interrupt
0x301110	T8_CLK1	T8 Ch.1 Input Clock Select Register	Select prescaler output clock
0x301112	T8_TR1	T8 Ch.1 Reload Data Register	Set reload data
0x301114	T8_TC1	T8 Ch.1 Counter Data Register	Counter data
0x301116	T8_CTL1	T8 Ch.1 Control Register	Set timer mode and start/stop timer
0x301118	T8_INT1	T8 Ch.1 Interrupt Control Register	Control interrupt
0x301120	T8_CLK2	T8 Ch.2 Input Clock Select Register	Select prescaler output clock
0x301122	T8_TR2	T8 Ch.2 Reload Data Register	Set reload data
0x301124	T8_TC2	T8 Ch.2 Counter Data Register	Counter data
0x301126	T8_CTL2	T8 Ch.2 Control Register	Set timer mode and start/stop timer
0x301128	T8_INT2	T8 Ch.2 Interrupt Control Register	Control interrupt
0x301130	T8_CLK3	T8 Ch.3 Input Clock Select Register	Select prescaler output clock
0x301132	T8_TR3	T8 Ch.3 Reload Data Register	Set reload data
0x301134	T8_TC3	T8 Ch.3 Counter Data Register	Counter data
0x301136	T8_CTL3	T8 Ch.3 Control Register	Set timer mode and start/stop timer
0x301138	T8_INT3	T8 Ch.3 Interrupt Control Register	Control interrupt
0x301140	T8_CLK4	T8 Ch.4 Input Clock Select Register	Select prescaler output clock
0x301142	T8_TR4	T8 Ch.4 Reload Data Register	Set reload data
0x301144	T8_TC4	T8 Ch.4 Counter Data Register	Counter data
0x301146	T8_CTL4	T8 Ch.4 Control Register	Set timer mode and start/stop timer
0x301148	T8_INT4	T8 Ch.4 Interrupt Control Register	Control interrupt
0x301150	T8_CLK5	T8 Ch.5 Input Clock Select Register	Select prescaler output clock
0x301152	T8_TR5	T8 Ch.5 Reload Data Register	Set reload data
0x301154	T8_TC5	T8 Ch.5 Counter Data Register	Counter data
0x301156	T8_CTL5	T8 Ch.5 Control Register	Set timer mode and start/stop timer
0x301158	T8_INT5	T8 Ch.5 Interrupt Control Register	Control interrupt
0x301160	T8_CLK6	T8 Ch.6 Input Clock Select Register	Select prescaler output clock
0x301162	T8_TR6	T8 Ch.6 Reload Data Register	Set reload data
0x301164	T8_TC6	T8 Ch.6 Counter Data Register	Counter data
0x301166	T8_CTL6	T8 Ch.6 Control Register	Set timer mode and start/stop timer
0x301168	T8_INT6	T8 Ch.6 Interrupt Control Register	Control interrupt
0x301170	T8_CLK7	T8 Ch.7 Input Clock Select Register	Select prescaler output clock
0x301172	T8_TR7	T8 Ch.7 Reload Data Register	Set reload data

Address	Register name		Function
0x301174	T8_TC7	T8 Ch.7 Counter Data Register	Counter data
0x301176	T8_CTL7	T8 Ch.7 Control Register	Set timer mode and start/stop timer
0x301178	T8_INT7	T8 Ch.7 Interrupt Control Register	Control interrupt

The 8-bit timer registers are described in detail below. These are 16-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

T8 Ch.x Input Clock Select Registers (T8_CLKx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
T8 Ch.x Input Clock Select Register (T8_CLKx)	0x301100 0x301170 (16 bits)	D15–4	–	reserved	T8 clock division ratio select (Prescaler output clock)	–	–	0 when being read. Source clock = PCLK1 (Ch.0/2/4/6) or PCLK2 (Ch.1/3/5/7)		
		D3–0	DF[3:0]	DF[3:0]		Division ratio	0x0		R/W	
						0xf	reserved			
						0xe	1/16384			
						0xd	1/8192			
						0xc	1/4096			
						0xb	1/2048			
						0xa	1/1024			
						0x9	1/512			
						0x8	1/256			
						0x7	1/128			
						0x6	1/64			
						0x5	1/32			
						0x4	1/16			
			0x3	1/8						
			0x2	1/4						
			0x1	1/2						
			0x0	1/1						

D[15:4] Reserved

D[3:0] DF[3:0]: T8 Input Clock Division Ratio Select Bits

Selects a PCLK division ratio to generate the T8 count clock.

Table 14.10.2 Count Clock (PCLK Division Ratio) Selection

DF[3:0]	Division ratio	DF[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

* Source clock = PCLK1 (for T8 Ch.0/2/4/6) or PCLK2 (for T8 Ch.1/3/5/7)

(Default: 0x0)

Note: Make sure the counter is halted before setting the count clock.

T8 Ch.x Reload Data Registers (T8_TRx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.x Reload Data Register (T8_TRx)	0x301102 0x301172 (16 bits)	D15–8	–	reserved	T8 reload data TR7 = MSB TR0 = LSB	–	–	0 when being read.
		D7–0	TR[7:0]	TR[7:0]		0x0 to 0xff	0x0	

D[15:8] Reserved

D[7:0] TR[7:0]: T8 Reload Data Bits

Sets the counter initial value. (Default: 0x0)

The reload data set in this register is preset to the counter when the timer is reset or the counter underflows. If the timer is started after resetting, it counts down from the reload value (initial value). This means that the reload value and the input clock frequency determine the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the desired wait time, the intervals between periodic interrupts or A/D triggers, and the programmable serial interface transfer clock.

T8 Ch.x Counter Data Registers (T8_TCx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.x Counter Data Register (T8_TCx)	0x301104 0x301174 (16 bits)	D15-8	–	reserved	–	–	–	0 when being read.
		D7-0	TC[7:0]	T8 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

D[15:8] Reserved

D[7:0] TC[7:0]: T8 Counter Data Bits

The counter data can be read out. (Default: 0xff)

This register is read-only and cannot be written to.

T8 Ch.x Control Registers (T8_CTLx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T8 Ch.x Control Register (T8_CTLx)	0x301106 0x301176 (16 bits)	D15-12	–	reserved	–	–	–	0 when being read.	
		D11-8	TFMD[3:0]	Fine mode setup (Ch.0 to Ch.3)	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.	
		–	–	reserved (Ch.4 to Ch.7)	–	–	–	0 when being read.	
		D7-5	–	reserved	–	–	–	–	
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W		
		D3-2	–	reserved	–	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset 0 Ignored	0	W		
		D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W		

D[15:12] Reserved

D[11:8] TFMD[3:0]: Fine Mode Setup Bits (Ch.0 to Ch.3)

Corrects the transfer rate error. (Default: 0x0)

TFMD[3:0] specifies the delay pattern to be inserted into a 16 underflow period. Inserting one delay extends the output clock cycle by one count clock cycle. This setting delays the interrupt timing in the same way.

Table 14.10.3 Delay Patterns Specified by TFMD[3:0]

TFMD[3:0]	Underflow number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0x0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
0x1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	D
0x2	–	–	–	–	–	–	–	D	–	–	–	–	–	–	–	D
0x3	–	–	–	–	–	–	–	D	–	–	–	D	–	–	–	D
0x4	–	–	–	D	–	–	–	D	–	–	–	D	–	–	–	D
0x5	–	–	–	D	–	–	–	D	–	–	–	D	–	D	–	D
0x6	–	–	–	D	–	D	–	D	–	–	–	D	–	D	–	D
0x7	–	–	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x8	–	D	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x9	–	D	–	D	–	D	–	D	–	D	–	D	–	D	D	D
0xa	–	D	–	D	–	D	D	D	–	D	–	D	–	D	D	D
0xb	–	D	–	D	–	D	D	D	–	D	D	D	–	D	D	D
0xc	–	D	D	D	–	D	D	D	–	D	D	D	–	D	D	D
0xd	–	D	D	D	–	D	D	D	–	D	D	D	D	D	D	D
0xe	–	D	D	D	D	D	D	D	–	D	D	D	D	D	D	D
0xf	–	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

D: Indicates the insertion of a delay cycle.

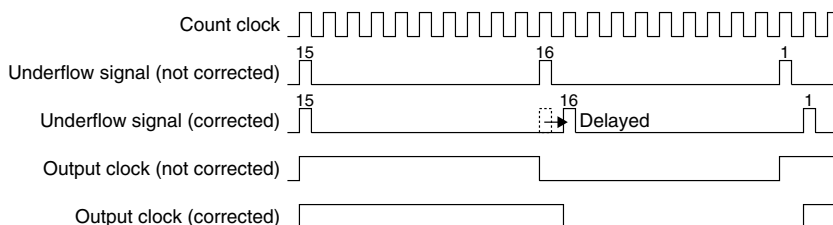


Figure 14.10.1 Delay Cycle Insertion in Fine Mode

D[11:8] Reserved (Ch.4 to Ch.7)

D[7:5] Reserved

D4 TRMD: Count Mode Select Bit

Selects the T8 count mode.

1 (R/W): One-shot mode

0 (R/W): Repeat mode (default)

Setting TRMD to 0 sets T8 to repeat mode. In this mode, once the count starts, the timer continues to run until stopped by the application program. When the counter underflows, the timer presets the counter to the reload data register value and continues the count. Thus, the timer periodically outputs an underflow pulse. Set T8 to this mode to generate periodic interrupts or A/D triggers at desired intervals or to generate a serial transfer clock.

Setting TRMD to 1 sets T8 to one-shot mode. In this mode, the timer stops automatically as soon as the counter underflows. This means only one interrupt can be generated after the timer starts. Note that the timer presets the counter to the reload data register value, then stops when an underflow occurs. Set T8 to this mode to set a specific wait time.

Note: Make sure the counter is halted before setting the count mode.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the timer.

1 (W): Reset

0 (W): Ignored

0 (R): Always 0 when read (default)

Writing 1 to this bit presets the counter to the reload data value.

D0 PRUN: Timer Run/Stop Control Bit

Controls the timer RUN/STOP.

1 (R/W): Run

0 (R/W): Stop (default)

The timer starts counting when PRUN is written as 1 and stops when written as 0. When the timer is stopped, the counter data is retained until reset or until the next RUN state.

T8 Ch.x Interrupt Control Registers (T8_INTx)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
T8 Ch.x Interrupt Control Register (T8_INTx)	0x301108 0x301178 (16 bits)	D15-9	–	reserved	–		–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable	0 Disable	0	R/W	
		D7-1	–	reserved	–		–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

D[15:9] Reserved

D8 T8IE: T8 Interrupt Enable Bit

Enables or disables interrupts caused by counter underflows for each channel.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting T8IE to 1 enables T8 interrupt requests to the ITC; setting to 0 disables interrupts.

D[7:1] Reserved

14 8-BIT TIMERS (T8)

D0 T8IF: T8 Interrupt Flag Bit

Indicates whether the cause of counter underflow interrupt has occurred or not.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

T8IF is the T8 interrupt flag that is set to 1 when the counter underflows.

T8IF is reset by writing 1.

15 16-bit PWM Timer (T16A5)

15.1 T16A5 Module Overview

The S1C33L26 includes a 16-bit PWM timer (T16A5) module with two timer channels.

The features of T16A5 are listed below.

- 16-bit up counter with a comparator and capture unit
- The count clock is selectable from 15 clocks output from the prescaler.
- Supports event counter function using an external clock.
- Includes a comparator that compares the counter value with two specified comparison values to generate interrupts and various output waveform including a PWM waveform.
- Includes a capture unit that captures counter values using two external trigger signals and generates interrupts.

Figure 15.1.1 shows the T16A5 configuration.

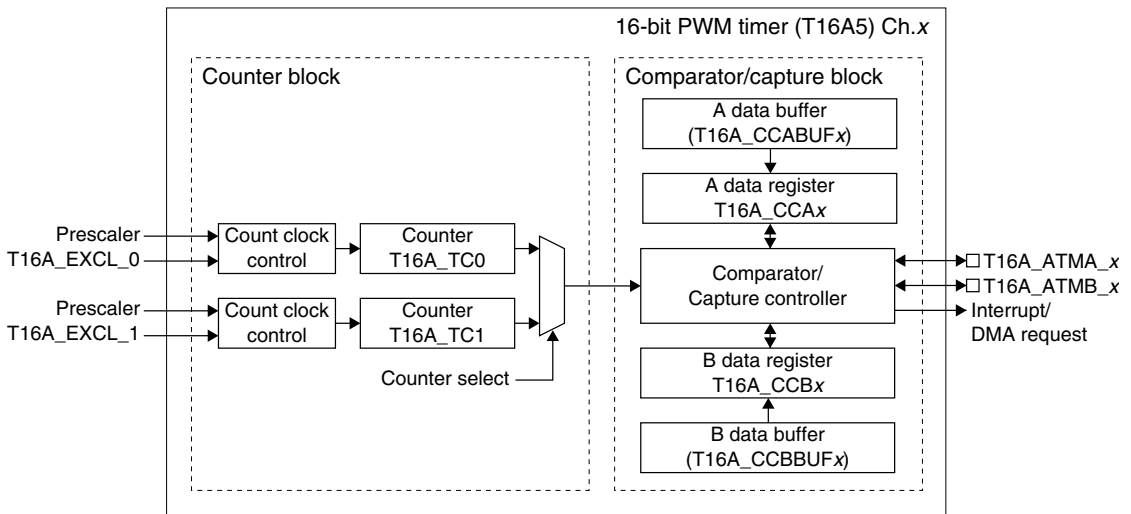


Figure 15.1.1 T16A5 Configuration (one channel)

The T16A5 module consists of a counter block and a comparator/capture block.

Counter block

The counter block includes a 16-bit up-counter that operates with a prescaler output clock, or the external count clock input from outside the IC. The 16-bit PWM timer (T16A5) allows software to run and stop the counter, and to reset the counter value (cleared to 0) as well as selection of the count clock. The counter can also be reset by the compare B signal output from the comparator/capture block.

Comparator/capture block

The comparator/capture block includes two systems (units A and B) of comparators that compare between the counter value and the specified comparison value and capture circuits that capture the counter value by an external trigger signal. Note, however, that the comparator and capture functions cannot be used at the same time in each system. One of the two functions must be selected by the software switch.

When using the comparator function, set the value(s) to be compared with the counter value to the compare A and/or compare B registers. When the counter reaches the value set in the compare A or compare B register, the comparator asserts the compare A or compare B signal. These signals can generate interrupts. Also the signals control the cycle time and duty ratio of the timer output signal allowing the timer to output a PWM or other waveform. In addition to these functions, the compare B signal is used to reset the counter.

15 16-BIT PWM TIMER (T16A5)

Comparison data can be read or written directly from/to the compare A and compare B registers. The compare buffers are separately provided to load data to the compare A and compare B registers automatically by the compare B signal. Software can select which of the compare register and buffer the comparison values are written to.

When the capture function is enabled, the compare A and compare B registers are used as the capture A and capture B registers, respectively. The capture A and capture B circuits can input a trigger signal individually, and the counter value is loaded to the respective capture register at the selected edge of the trigger signal.

The capturing operation can generate an interrupt, this make it possible to read the captured data in the interrupt handler routine. Also an overwrite interrupt can be generated for the error handling when the counter value is captured before reading the previous captured data.

Note: Both channels of the T16A5 module has the same functions except for the control register addresses. The description in this section applies to both channels of the T16A5 module unless otherwise specified. Letter 'x' in the register name refers to the channel number (0 or 1).

Example: T16A_CTLx register

Ch.0: T16A_CTL0 register

Ch.1: T16A_CTL1 register

15.2 T16A5 Input/Output Pins

Table 15.2.1 lists the input/output pins for the T16A5 module.

Table 15.2.1 List of T16A5 Pins

Pin name	I/O	Qty	Function
T16A_EXCL_0 (Ch.0) T16A_EXCL_1 (Ch.1)	I	2	T16A5 external clock input pin Inputs an external clock for the event counter function. The T16A_EXCL_0 pin can also be used as the WDT external clock input pin.
T16A_ATMA_0 (Ch.0) T16A_ATMA_1 (Ch.1)	I/O	2	T16A5 system A input/output pin Outputs timer generating signal in comparator mode. Inputs a counter-capture trigger signal in capture mode.
T16A_ATMB_0 (Ch.0) T16A_ATMB_1 (Ch.1)	I/O	2	T16A5 system B input/output pin Outputs timer generating signal in comparator mode. Inputs a counter-capture trigger signal in capture mode.

The T16A5 input/output pins (T16A_EXCL_x, T16A_ATMA_x, T16A_ATMB_x) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as T16A5 input/output pins.

For detailed information on pin function switching, see the "I/O Ports (GPIO)" chapter.

15.3 Count Clock

The count clock is selected by CLKS[3:0]/T16A_CTLx register from the 15 types generated by the prescaler (PSC Ch.0) dividing the PCLK1 clock into 1/1 to 1/16K and an external clock.

Table 15.3.1 Count Clock (PCLK1 Division Ratio) Selection

CLKS[3:0]	Division ratio	CLKS[3:0]	Division ratio
0xf	External clock	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

- Notes:**
- Make sure the counter is halted before setting the count clock.
 - When using an external clock, the external clock cycle must be at least two CPU operating clock cycles.

For controlling the prescaler, refer to the “Prescaler (PSC)” chapter.

15.4 T16A5 Operating Modes

T16A5 provides some operating modes to support various usages. This section describes the functions of each operating mode and how to enter the mode.

15.4.1 Comparator Mode and Capture Mode

The T16A_CCA_x and T16A_CCB_x registers that are embedded in the comparator/capture block can be set to comparator mode or capture mode, individually. The T16A_CCA_x register mode is selected using CCAMD/T16A_CCCTL_x register and the T16A_CCB_x register mode is selected using CCBMD/T16A_CCCTL_x register.

Comparator mode (CCAMD/CCBMD = 0, default)

The comparator mode compares the counter value and the comparison value set by software. It generates an interrupt and toggles the timer output signal level when the values are matched. The T16A_CCA_x and T16A_CCB_x registers function as the compare A and compare B registers that are used for loading comparison values in this mode.

The counter channel (Ch.0 or Ch.1) to be used can be selected using T16SEL[1:0]/T16A_CTL_x register. This selection enables the both channels output compare A and compare B signals in sync with one 16-bit counter.

When the counter reaches the value set in the compare A register during counting, the comparator asserts the compare A signal. At the same time the compare A interrupt flag is set and an interrupt signal is output to the ITC if the interrupt has been enabled.

When the counter reaches the value set in the compare B register, the comparator asserts the compare B signal. At the same time the compare B interrupt flag is set and an interrupt signal is output to the ITC if the interrupt is enabled. Furthermore, the counter is reset to 0.

Note: The intervals of the compare A and compare B interrupts must be longer than three count clock cycles. Otherwise, the second interrupt will be omitted by T16A5.

The compare A and compare B signals are also used to generate a timer output waveform. See Section 15.6, “Timer Output Control,” for more information.

To generate PWM waveform, the T16A_CCA_x and T16A_CCB_x registers must be both placed into comparator mode.

Compare buffers

Comparison data can be read or written directly from/to the compare registers. Comparison data for system A or B can also be written to the compare buffer so that it will be loaded to the compare A or compare B register by the compare B signal. The CBUFEN/T16A_CTL_x register is used to select whether comparison data is written to the compare register or buffer.

Setting CBUFEN to 0 (default) selects the compare registers. Setting it to 1 selects the compare buffers. Although the T16A_CCA_x and T16A_CCB_x registers are used to write compare data even if CBUFEN = 1, compare buffers will be accessed. Note that compare data is always read from the compare register regardless of whether the compare buffer is enabled or not.

Capture mode (CCAMD/CCBMD = 1)

The capture mode captures the counter value when an external event such as a key entry occurs (at the specified edge of the external input signal). In this mode, the T16A_CCA_x and/or T16A_CCB_x registers function as the capture A and/or capture B registers for loading the captured data. To input a counter capture trigger signal, the capture A circuit uses the T16A_ATMA_x pin and the capture B circuit uses the T16A_ATMB_x pin. The T16A_ATMA_x and T16A_ATMB_x pins are shared with the timer outputs. They are configured for input when the system A or B is set to capture mode.

The counter channel (Ch.0 or Ch.1) to be used can be selected using T16SEL[1:0]/T16A_CTLx register. This selection enables the both channels capture values of the same 16-bit counter.

The trigger edge of the input signal can be selected using the CAPATR[1:0]/T16A_CCCTLx register for capture A and CAPBTR[1:0]/T16A_CCCTLx register for capture B.

Table 15.4.1.1 Capture Trigger Edge Selection

CAPATR[1:0]/CAPBTR[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

When a specified trigger edge is input during counting, the current counter value is loaded to the capture register. At the same time the capture A or capture B interrupt flag is set and an interrupt signal is output to the ITC if the interrupt has been enabled. This interrupt can be used to read the captured data from the T16A_CCAx or T16A_CCBx register. For example, external event cycles and pulse widths can be measured from the difference between two captured counter values read.

If the captured data is overwritten by the next trigger when the capture A or capture B interrupt flag has already been set, the overwrite interrupt flag will be set. This interrupt can be used to execute an overwrite error handling. To avoid occurrence of unnecessary overwrite interrupt, the capture A or capture B interrupt flag must be reset after the captured data has been read from the T16A_CCAx or T16A_CCBx register.

- Notes:**
- The correct captured data may not be obtained if the captured data is read at the same time the next value is being captured. Read the capture register twice to check if the read data is correct as necessary.
 - To capture counter data properly, both the High and Low period of the T16A_ATMA_x/T16A_ATMB_x trigger signal must be longer than three count clock cycles.

The setting of CAPATR[1:0] or CAPBTR[1:0] is ineffective in comparator mode. No counter capturing operation will be performed, as the T16A_ATMA_x/T16A_ATMB_x pin is configured for output.

The capture mode cannot generate/output the timer signal, as no compare signal is generated.

15.4.2 Repeat Mode and One-Shot Mode

Each counter features two count modes: repeat mode and one-shot mode. The count mode is selected using TMMD/T16A_CTLx register.

Repeat mode (TMMD = 0, default)

Setting TMMD to 0 sets the counter to repeat mode.

In this mode, once the count starts, the counter continues running until stopped by the application program. If the counter is reset to 0 or returns to 0 due to a counter overflow, the counter continues the count. The counter should be set to this mode to generate periodic interrupts at desired intervals or to generate a timer output waveform.

One-shot mode (TMMD = 1)

Setting TMMD to 1 sets the counter to one-shot mode.

In this mode, the counter stops automatically as soon as the counter is reset or it overflows. The counter should be set to this mode to set a specific wait time or for pulse width measurement.

15.5 Counter Control

15.5.1 Counter Reset

The counter can be reset to 0 by writing 1 to PRESET/T16A_CTLx register.

Normally, the counter should be reset by writing 1 to this bit before starting the count.

The counter is reset by the hardware if the counter reaches the compare B register value after the count starts.

Note: The counter is reset in sync with the counter clock. It may take long time depending on the count clock selected. To make sure that the reset operation has finished, check if PRESET is set to 0 or BUSY/T16A_CTLx register is set to 0 (idle) by reading these bits.

15.5.2 Counter RUN/STOP Control

Make the following settings before starting the count operation.

- (1) Switch the input/output pin functions to be used for T16A5. Refer to the “I/O Port (GPIO)” chapter.
- (2) Select operating modes. See Section 15.4.
- (3) Select the clock source. See Section 15.3.
- (4) Configure the timer outputs. See Section 15.6.
- (5) If using interrupts, set the interrupt level and enable the T16A5 interrupts. See Section 15.7.
- (6) Reset the counter to 0. See Section 15.5.1.
- (7) Set comparison data (in comparator mode). See Section 15.4.1.

The T16A5 module provides PRUN/T16A_CTLx register to control the counter operation.

The counter starts counting when 1 is written to PRUN. Writing 0 to PRUN disables clock input and stops the count.

This control does not affect the counter data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.

If PRUN and PRESET are written as 1 simultaneously, the counter starts counting after reset.

- Notes:**
- Always make sure that BUSY/T16A_CTLx register is set to 0 (idle) before writing to the T16A_CTLx register.
 - Setting PRUN to 1 may not start the counter immediately as the counter starts counting in sync with the count clock. The counter operation should be checked by reading PRUN (check if it is set to 1) or BUSY (check if it is set to 0).

15.5.3 Reading Counter Values

The counter value can be read from T16ATC[15:0]/T16A_TCx register even if the counter is running. However, the counter value should be read at once using a 16-bit transfer instruction. If data is read twice using an 8-bit transfer instruction, the correct value may not be obtained due to occurrence of count up between readings.

Note: The counter value must be read from the T16A_TCx register of the channel selected using T16SEL[1:0]/T16A_CTLx register.

15.5.4 Timing Charts

Comparator mode

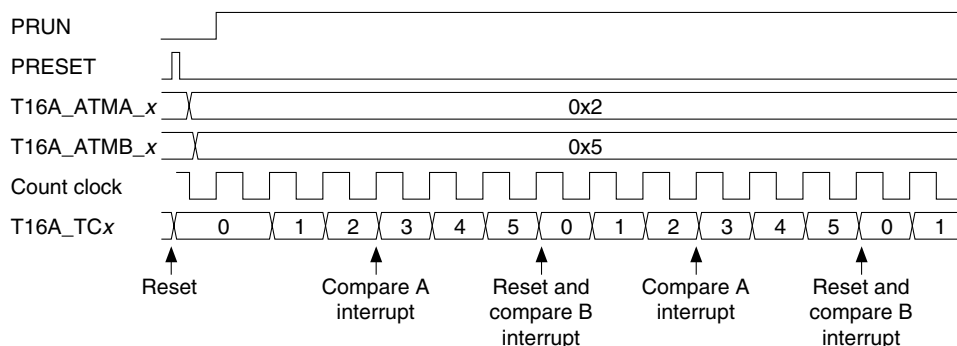


Figure 15.5.4.1 Operation Timing in Comparator Mode

Capture mode

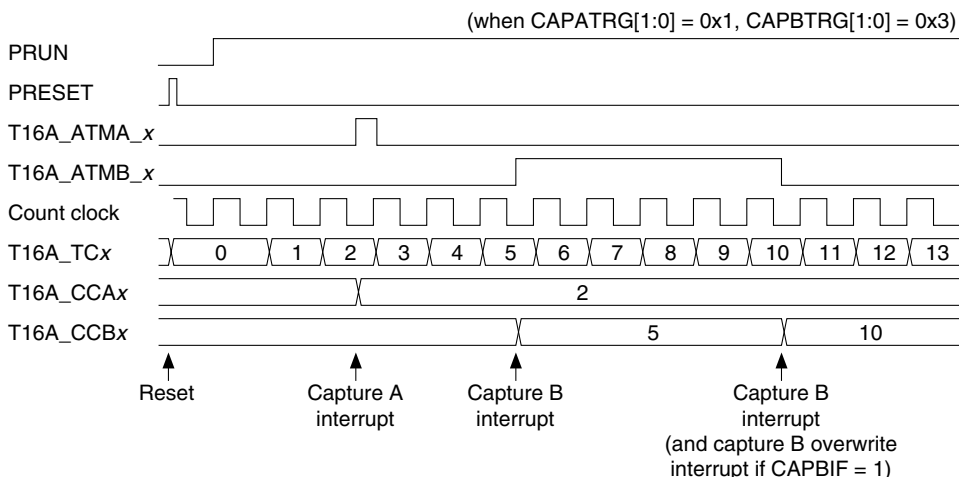


Figure 15.5.4.2 Operation Timing in Capture Mode

15.6 Timer Output Control

T16A5 in comparator mode can generate two TOUT signals using the compare A and compare B signals and can output them to external devices. Figure 15.6.1 shows the TOUT output circuit.

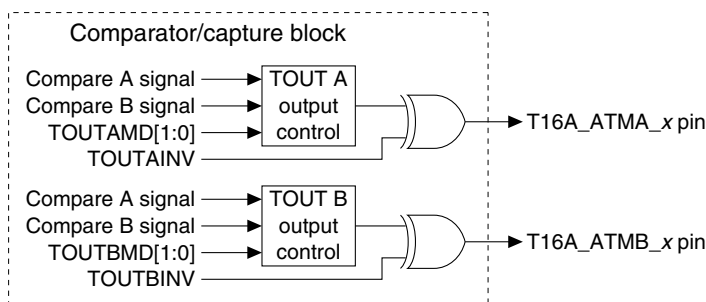


Figure 15.6.1 TOUT Output Circuit

T16A5 includes two TOUT output circuits and their signal generation and output can be controlled individually. Although the output circuit and register names use letters 'A' and 'B' to distinguish two systems, it does not mean that they correspond to compare A and B signals.

Note: The compare A and compare B signals can be generated from the counter value of another channel by setting T16SEL[1:0]/T16A_CTLx register.

TOUT output pins

The TOUT A signal is output from the T16A_ATMA_x pin and TOUT B signal is output from the T16A_ATMB_x pin. The T16A_ATMA_x and T16A_ATMB_x pins are shared with the capture trigger inputs. They are configured for output when the system A or B is set to comparator mode.

TOUT generation mode

TOUTAMD[1:0]/T16A_CCCTLx register (for system A) or TOUTBMD[1:0]/T16A_CCCTLx register (for system B) is used to set how the TOUT signal is changed by the compare A and compare B signals.

Table 15.6.1 TOUT Generation Mode

TOUTAMD[1:0]/ TOUTBMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTAMD[1:0] and TOUTBMD[1:0] are also used to turn the TOUT output on and off.

TOUT signal polarity selection

By default, an active high output signal is generated. This logic can be inverted using TOUTAINV/T16A_CCCTLx register (for system A) or TOUTBINV/T16A_CCCTLx register (for system B). Writing 1 to TOUTAINV/TOUTBINV causes the timer to generate an active low TOUT signal.

Resetting the counter sets the TOUT signal to the inactive level.

Figure 15.6.2 shows the TOUT output waveform.

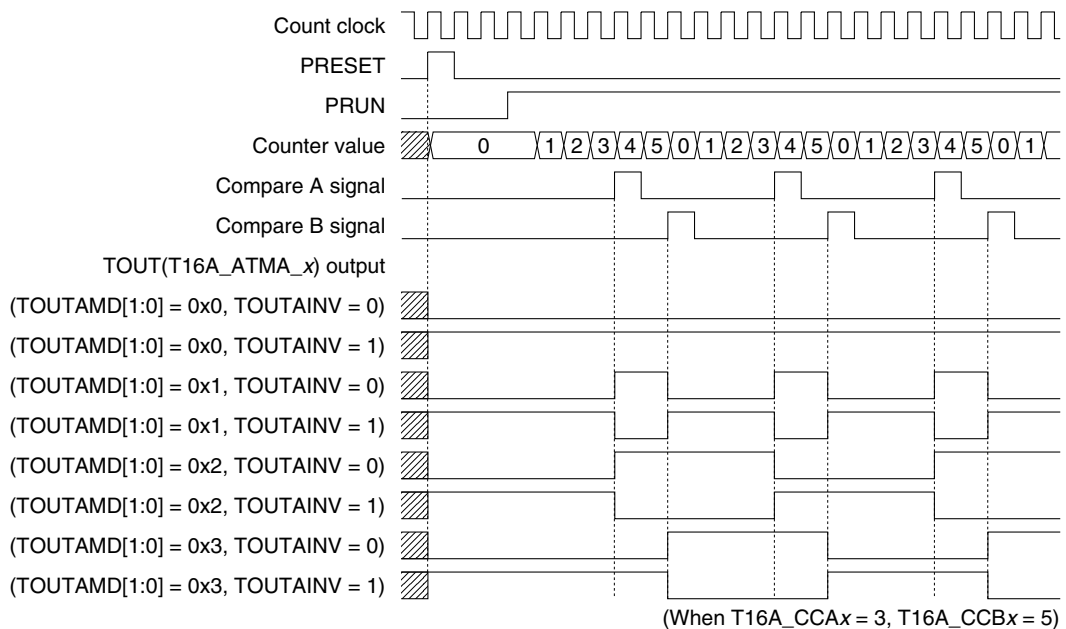


Figure 15.6.2 TOUT Output Waveform

15.7 T16A5 Interrupts and DMA

This section describes the T16A5 interrupts and invoking DMA.

For more information on interrupt processing and DMA transfer, see the “Interrupt Controller (ITC)” chapter and the “DMA Controller (DMAC)” chapter, respectively.

15.7.1 Interrupts

Each T16A5 channel can generate the following six kinds of interrupts:

- Compare A interrupt (in comparator mode)
- Compare B interrupt (in comparator mode)
- Capture A interrupt (in capture mode)
- Capture B interrupt (in capture mode)
- Capture A overwrite interrupt (in capture mode)
- Capture B overwrite interrupt (in capture mode)

A T16A5 channel outputs a single interrupt signal shared by the above interrupt causes to the interrupt controller (ITC). Read the interrupt flags in the T16A5 module to identify the interrupt cause that has been occurred.

Interrupts in comparator mode

Compare A interrupt

This interrupt request is generated when the counter matches the compare A register value during counting in comparator mode. It sets the interrupt flag CAIF/T16A_IFLGx register in the T16A5 module to 1.

To use this interrupt, set CAIE/T16A_IENx register to 1. If CAIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Compare B interrupt

This interrupt request is generated when the counter matches the compare B register value during counting in comparator mode. It sets the interrupt flag CBIF/T16A_IFLGx register in the T16A5 module to 1.

To use this interrupt, set CBIE/T16A_IENx register to 1. If CBIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Note: The intervals of the compare A and compare B interrupts must be four or more count clock cycles. Otherwise, the second interrupt will be omitted by T16A5.

Interrupts in capture mode

Capture A interrupt

This interrupt request is generated when the counter value is captured in the capture A register by an external trigger during counting in capture mode. It sets the interrupt flag CAPAIF/T16A_IFLGx register in the T16A5 module to 1.

To use this interrupt, set CAPAIE/T16A_IENx register to 1. If CAPAIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Capture B interrupt

This interrupt request is generated when the counter value is captured in the capture B register by an external trigger during counting in capture mode. It sets the interrupt flag CAPBIF/T16A_IFLGx register in the T16A5 module to 1.

To use this interrupt, set CAPBIE/T16A_IENx register to 1. If CAPBIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Capture A overwrite interrupt

This interrupt request is generated if the capture A register is overwritten by a new external trigger when the capture A interrupt flag CAPAIF has been set (a counter value has already been loaded to the capture A register). It sets the interrupt flag CAPAOWIF/T16A_IFLGx register in the T16A5 module to 1.

To use this interrupt, set CAPAOWIE/T16A_IENx register to 1. If CAPAOWIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

CAPAOWIF will be set if the capture A register is overwritten when CAPAIF has been set regardless of whether the capture A register has been read or not. Therefore, be sure to reset CAPAIF immediately after the capture A register is read.

Capture B overwrite interrupt

This interrupt request is generated if the capture B register is overwritten by a new external trigger when the capture B interrupt flag CAPBIF has been set (a counter value has already been loaded to the capture B register). It sets the interrupt flag CAPBOWIF/T16A_IFLGx register in the T16A5 module to 1.

To use this interrupt, set CAPBOWIE/T16A_IENx register to 1. If CAPBOWIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

CAPBOWIF will be set if the capture B register is overwritten when CAPBIF has been set regardless of whether the capture B register has been read or not. Therefore, be sure to reset CAPBIF immediately after the capture B register is read.

If the interrupt flag is set to 1 when the interrupt has been enabled, the T16A5 module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C33 PE Core interrupt conditions are satisfied.

For more information on interrupt control registers and the operation when an interrupt occurs, see the “Interrupt Controller (ITC)” chapter.

- Notes:**
- Reset the interrupt flag before enabling interrupts with the interrupt enable bit to prevent occurrence of unwanted interrupt. The interrupt flag is reset by writing 1.
 - After an interrupt occurs, the interrupt flag in the T16A5 module must be reset in the interrupt handler routine.

15.7.2 DMA Transfer

The causes of compare A/capture A and compare B/capture B interrupts can invoke a DMA. This allows continuous data transfer via the DMAC between memory and the compare/capture data register. The interrupt signal is output to both the ITC and DMAC. Therefore, DMA transfer can be performed without generating a T16A5 interrupt.

Two DMAC channels (Ch.2 and 3 or Ch.4 and 5) are available for each T16A5 channel. The DMAC channels to be used can be selected using DMASEL[1:0]/T16A_CTLx register.

For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

15.8 Control Register Details

Table 15.8.1 List of 16-bit PWM Timer (T16A5) Register

Address	Register name		Function
0x301180	T16A_CTL0	T16A5 Ch.0 Counter Control Register	Control counter
0x301182	T16A_TC0	T16A5 Ch.0 Counter Data Register	Counter data
0x301184	T16A_CCCTL0	T16A5 Ch.0 Comparator/Capture Control Register	Control comparator/capture block and TOUT
0x301186	T16A_CCA0	T16A5 Ch.0 Comparator/Capture A Data Register	Compare A/capture A data
0x301188	T16A_CCB0	T16A5 Ch.0 Comparator/Capture B Data Register	Compare B/capture B data
0x30118a	T16A_IEN0	T16A5 Ch.0 Comparator/Capture Interrupt Enable Register	Enable/disable T16A5 interrupts
0x30118c	T16A_IFLG0	T16A5 Ch.0 Comparator/Capture Interrupt Flag Register	Indicate T16A5 interrupt cause status
0x301190	T16A_CTL1	T16A5 Ch.1 Counter Control Register	Control counter
0x301192	T16A_TC1	T16A5 Ch.1 Counter Data Register	Counter data
0x301194	T16A_CCCTL1	T16A5 Ch.1 Comparator/Capture Control Register	Control comparator/capture block and TOUT
0x301196	T16A_CCA1	T16A5 Ch.1 Comparator/Capture A Data Register	Compare A/capture A data
0x301198	T16A_CCB1	T16A5 Ch.1 Comparator/Capture B Data Register	Compare B/capture B data
0x30119a	T16A_IEN1	T16A5 Ch.1 Comparator/Capture Interrupt Enable Register	Enable/disable T16A5 interrupts
0x30119c	T16A_IFLG1	T16A5 Ch.1 Comparator/Capture Interrupt Flag Register	Indicate T16A5 interrupt cause status

The 16-bit PWM timer (T16A5) registers are described in detail below. These are 16-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

T16A5 Ch.x Counter Control Registers (T16A_CTLx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T16A5 Ch.0 Counter Control Register (T16A_CTL0)	0x301180 (16 bits)	D15-14	–	reserved	–	–	–	0 when being read.	
		D13-12	DMASEL [1:0]	DMAC channel select	DMASEL[1:0]	DMAC channel	0x0	R/W	
					0x3	Ch.4/5			
					0x2	Ch.2/3			
					0x1	Ch.4/5			
					0x0	Ch.2/3			
		D11-8	CLKS[3:0]	Counter clock (division ratio) select	CLKS[3:0]	Division ratio	0x0	R/W	Source clock = PCLK1
					0xf	External clock			
					0xe	1/16384			
					0xd	1/8192			
					0xc	1/4096			
0xb	1/2048								
0xa	1/1024								
0x9	1/512								
0x8	1/256								
0x7	1/128								
D7	BUSY	Register writing status	1 Busy	0 Idle	0	R			
D6	–	reserved	–	–	–	–	0 when being read.		
D5-4	T16SEL [1:0]	Counter select	T16SEL[1:0]	Counter channel	0x0	R/W			
			0x3	Ch.1					
			0x2	Ch.0					
			0x1	Ch.1					
D3	CBUFEN	Compare buffer enable	1 Enable	0 Disable	0	R/W			
D2	TMMD	Count mode select	1 One-shot	0 Repeat	0	R/W			
D1	PRESET	Counter reset	1 Reset	0 Ignored	0	W	0 when being read.		
D0	PRUN	Counter run/stop control	1 Run	0 Stop	0	R/W			
T16A5 Ch.1 Counter Control Register (T16A_CTL1)	0x301190 (16 bits)	D15-14	–	reserved	–	–	–	0 when being read.	
		D13-12	DMASEL [1:0]	DMAC channel select	DMASEL[1:0]	DMAC channel	0x1	R/W	
					0x3	Ch.4/5			
					0x2	Ch.2/3			
					0x1	Ch.4/5			
					0x0	Ch.2/3			
		D11-8	CLKS[3:0]	Counter clock (division ratio) select	CLKS[3:0]	Division ratio	0x0	R/W	Source clock = PCLK1
					0xf	External clock			
					0xe	1/16384			
					0xd	1/8192			
					0xc	1/4096			
0xb	1/2048								
0xa	1/1024								
0x9	1/512								
0x8	1/256								
0x7	1/128								
D7	BUSY	Register writing status	1 Busy	0 Idle	0	R			
D6	–	reserved	–	–	–	–	0 when being read.		
D5-4	T16SEL [1:0]	Counter select	T16SEL[1:0]	Counter channel	0x1	R/W			
			0x3	Ch.1					
			0x2	Ch.0					
			0x1	Ch.1					
D3	CBUFEN	Compare buffer enable	1 Enable	0 Disable	0	R/W			
D2	TMMD	Count mode select	1 One-shot	0 Repeat	0	R/W			
D1	PRESET	Counter reset	1 Reset	0 Ignored	0	W	0 when being read.		
D0	PRUN	Counter run/stop control	1 Run	0 Stop	0	R/W			

D[15:14] Reserved

D[13:12] DMASEL[1:0]: DMAC Channel Select Bits

Selects the DMAC channels to be used for DMA transfer when a cause of compare A/B or capture A/B interrupt occurs.

Table 15.8.2 DMAC Channel Selection

DMASEL[1:0]	DMAC channels
0x3	Ch.4 and Ch.5
0x2	Ch.2 and Ch.3
0x1	Ch.4 and Ch.5
0x0	Ch.2 and Ch.3

(Default: 0x0/T16A5 Ch.0, 0x1/T16A5 Ch.1)

D[11:8] CLKS[3:0]: Counter Clock Select Bits

Selects the counter clock from the 15 different prescaler (PSC Ch.0) output clocks (PCLK1 division ratio) and an external clock (T16A_EXCL_x input clock).

Table 15.8.3 Counter Clock (PCLK1 Division Ratio) Selection

CLKS[3:0]	Division ratio	CLKS[3:0]	Division ratio
0xf	External clock	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

- Note:**
- Make sure the counter is halted before setting the count clock.
 - When using an external clock, the external clock cycle must be at least two CPU operating clock cycles.

D7 BUSY: Register Writing Status Bit

Indicates the T16A5 register writing status.

- 1 (R): Busy
0 (R): Idle (default)

BUSY goes 1 when data is written to the T16A_CTLx, T16A_CCAx, or T16A_CCBx register and it reverts to 0 upon completion of the writing operation.

Note: Make sure that BUSY is set to 0 before writing to these registers.

D6 Reserved**D[5:4] T16SEL[1:0]: Counter Select Bits**

Selects the counter channel.

Table 15.8.4 Counter Channel Selection

T16SEL[1:0]	Counter channel
0x3	Ch.1
0x2	Ch.0
0x1	Ch.1
0x0	Ch.0

(Default: 0x0/T16A5 Ch.0, 0x1/T16A5 Ch.1)

A timer channel (comparator/capture block) allows use of the counter in another channel. This enables the comparator/capture blocks to compare and capture values of the same counter.

From the T16A_TCx register, values of the counter channel selected by T16SEL[1:0] are read out.

D3 CBUFEN: Compare Buffer Enable Bit

Enables or disables writing to the compare buffer.

- 1 (R/W): Enabled
0 (R/W): Disabled (default)

When CBUFEN is set to 1, compare data is written via the compare data buffer. The buffer contents are loaded into the compare A and compare B registers by the compare B signal.

15 16-BIT PWM TIMER (T16A5)

When CBUFEN is set to 0, compare data is written directly to the compare A and compare B registers. Compare data is always read from compare A and compare B registers.

D2 TMMD: Count Mode Select Bit

Selects the count mode.

1 (R/W): One-shot mode

0 (R/W): Repeat mode (default)

Setting TMMD to 0 sets the counter to repeat mode. In this mode, once the count starts, the counter continues counting until stopped by the application program.

Setting TMMD to 1 sets the counter to one-shot mode. In this mode, the counter stops counting automatically as soon as the counter is reset by the compare B signal as well as stopped via software.

D1 PRESET: Counter Reset Bit

Resets the counter.

1 (W): Reset

0 (W): Ignored

0 (R): Normally 0 when read out (default)

Writing 1 to this bit resets the counter to 0.

D0 PRUN: Counter Run/Stop Control Bit

Starts/stops the count.

1 (W): Run

0 (W): Stop

1 (R): Counting

0 (R): Stopped (default)

The counter starts counting when PRUN is written as 1 and stops when written as 0. The counter data is retained even if the counter is stopped.

T16A5 Ch.x Counter Data Registers (T16A_TCx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A5 Ch.x Counter Data Register (T16A_TCx)	0x301182 0x301192 (16 bits)	D15-0	T16ATC [15:0]	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	

D[15:0] T16ATC[15:0]: Counter Data Bits

Counter data can be read out. (Default: 0x0)

The counter value can be read out even if the counter is running. However, the counter value should be read at once using a 16-bit transfer instruction. If data is read twice using an 8-bit transfer instruction, the correct value may not be obtained due to occurrence of count up between readings.

Note: The counter value must be read from the T16A_TCx register of the channel selected using T16SEL[1:0]/T16A_CTLx register.

T16A5 Ch.x Comparator/Capture Control Registers (T16A_CCCTLx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T16A5 Ch.x Comparator/ Capture Control Register (T16A_CCCTLx)	0x301184 0x301194 (16 bits)	D15–14	CAPBTRG [1:0]	Capture B trigger select	CAPBTRG[1:0]	Trigger edge	0x0	R/W	
		0x3			↑ and ↓				
		0x2	↓						
		0x1	↑						
		0x0	None						
	D13–12	TOUTBMD [1:0]	TOUT B mode select	TOUTBMD[1:0]	Mode	0x0	R/W		
	0x3			cmp B: ↑ or ↓					
		0x2	cmp A: ↑ or ↓						
		0x1	cmp A: ↑, B: ↓						
		0x0	Off						
D11–10	–	reserved	–	–	–	–	0 when being read.		
D9	TOUTBINV	TOUT B invert	1	Invert	0	Normal	0	R/W	
D8	CCBMD	T16A_CCB register mode select	1	Capture	0	Comparator	0	R/W	
D7–6	CAPATR [1:0]	Capture A trigger select	CAPATR[1:0]	Trigger edge	0x0	R/W			
							0x3	↑ and ↓	
	0x2	↓							
	0x1	↑							
	0x0	None							
D5–4	TOUTAMD [1:0]	TOUT A mode select	TOUTAMD[1:0]	Mode	0x0	R/W			
							0x3	cmp B: ↑ or ↓	
	0x2	cmp A: ↑ or ↓							
	0x1	cmp A: ↑, B: ↓							
	0x0	Off							
D3–2	–	reserved	–	–	–	–	0 when being read.		
D1	TOUTAINV	TOUT A invert	1	Invert	0	Normal	0	R/W	
D0	CCAMD	T16A_CCA register mode select	1	Capture	0	Comparator	0	R/W	

D[15:14] CAPBTRG[1:0]: Capture B Trigger Select Bits

Selects the trigger edge(s) of the external signal (T16A_ATMB_x input) at which the counter value is captured in the capture B register.

Table 15.8.5 Capture B Trigger Edge Selection

CAPBTRG[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

CAPBTRG[1:0] are control bits for capture mode and are ineffective in comparator mode.

D[13:12] TOUTBMD[1:0]: TOUT B Mode Select Bits

Configures how the TOUT B signal waveform (T16A_ATMB_x output) is changed by the compare A and compare B signals. These bits are also used to turn the TOUT B output on and off.

Table 15.8.6 TOUT B Generation Mode

TOUTBMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTBMD[1:0] are control bits for comparator mode and are ineffective in capture mode.

D[11:10] Reserved

D9 TOUTBINV: TOUT B Invert Bit

Selects the TOUT B signal (T16A_ATMB_x output) polarity.

1 (R/W): Inverted (active low)

0 (R/W): Normal (active high) (default)

Writing 1 to TOUTBINV generates an active low signal (off level = high) for the TOUT B output. When TOUTBINV is 0, an active high signal (off level = low) is generated.

TOUTBINV is a control bit for comparator mode and is ineffective in capture mode.

D8 CCBMD: T16A_CCB Register Mode Select Bit

Selects the T16A_CCB_x register function (comparator mode or capture mode).

1 (R/W): Capture mode

0 (R/W): Comparator mode (default)

Writing 1 to CCBMD configures the T16A_CCB_x register as the capture B register (capture mode) to which the counter data will be loaded by the external trigger signal. When CCBMD is 0, the T16A_CCB_x register functions as the compare B register (comparator mode) for writing a comparison value to generate the compare B signal.

D[7:6] CAPATRG[1:0]: Capture A Trigger Select Bits

Selects the trigger edge(s) of the external signal (T16A_ATMA_x input) at which the counter value is captured in the capture A register.

Table 15.8.7 Capture A Trigger Edge Selection

CAPATRG[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

CAPATRG[1:0] are control bits for capture mode and are ineffective in comparator mode.

D[5:4] TOUTAMD[1:0]: TOUT A Mode Select Bits

Configures how the TOUT A signal waveform (T16A_ATMA_x output) is changed by the compare A and compare B signals. These bits are also used to turn the TOUT A output on and off.

Table 15.8.8 TOUT A Generation Mode

TOUTAMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTAMD[1:0] are control bits for comparator mode and are ineffective in capture mode.

D[3:2] Reserved**D1 TOUTAINV: TOUT A Invert Bit**

Selects the TOUT A signal (T16A_ATMA_x output) polarity.

1 (R/W): Inverted (active low)

0 (R/W): Normal (active high) (default)

Writing 1 to TOUTAINV generates an active low signal (off level = high) for the TOUT A output. When TOUTAINV is 0, an active high signal (off level = low) is generated.

TOUTAINV is a control bit for comparator mode and is ineffective in capture mode.

D0 CCAMD: T16A_CCA Register Mode Select Bit

Selects the T16A_CCA_x register function (comparator mode or capture mode).

1 (R/W): Capture mode

0 (R/W): Comparator mode (default)

Writing 1 to CCAMD configures the T16A_CCA_x register as the capture A register (capture mode) to which the counter data will be loaded by the external trigger signal. When CCAMD is 0, the T16A_CCA_x register functions as the compare A register (comparator mode) for writing a comparison value to generate the compare A signal.

T16A5 Ch.x Comparator/Capture A Data Registers (T16A_CCAx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A5 Ch.x Comparator/Capture A Data Register (T16A_CCAx)	0x301186 0x301196 (16 bits)	D15-0	CCA[15:0]	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CCA[15:0]: Compare/Capture A Data Bits

In comparator mode (CCAMD/ T16A_CCCTLx register = 0)

Sets a compare A data, which will be compared with the counter value, through this register.

When CBUFEN/T16A_CTLx register is set to 0, accessing to this register directly read/write from/to the compare A register.

When CBUFEN is set to 1, writing to this register loads the data to the compare A buffer. Compare data is always read from this register. The buffer contents are loaded into the compare A register when the counter is reset by the compare B signal.

The data set is compared with the counter data. When the counter reaches the comparison value set, the compare A signal is asserted and a cause of compare A interrupt occurs. Furthermore, the TOUT output waveform changes when TOUTAMD[1:0]/T16A_CCCTLx register or TOUTBMD[1:0]/T16A_CCCTLx register is set to 0x2 or 0x1. These processes do not affect the counter data and the count up operation.

In capture mode (CCAMD = 1)

When the counter value is captured at the external trigger signal (T16A_ATMA_x input) edge selected using CAPATR[1:0]/T16A_CCCTLx register, the captured value is loaded to this register. At the same time a capture A interrupt can be generated, thus the captured counter value can be read out in the interrupt handler.

T16A5 Ch.x Comparator/Capture B Data Registers (T16A_CCBx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A5 Ch.x Comparator/Capture B Data Register (T16A_CCBx)	0x301188 0x301198 (16 bits)	D15-0	CCB[15:0]	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CCB[15:0]: Compare/Capture B Data Bits

In comparator mode (CCBMD/ T16A_CCCTLx register = 0)

Sets a compare B data, which will be compared with the counter value, through this register.

When CBUFEN/T16A_CTLx register is set to 0, accessing to this register directly read/write from/to the compare B register.

When CBUFEN is set to 1, writing to this register loads the data to the compare B buffer. Compare data is always read from this register. The buffer contents are loaded into the compare B register when the counter is reset by the compare B signal.

The data set is compared with the counter data. When the counter reaches the comparison value set, the compare B signal is asserted and a cause of compare B interrupt occurs. The counter is reset to 0. Furthermore, the TOUT output waveform changes when TOUTAMD[1:0]/T16A_CCCTLx register or TOUTBMD[1:0]/T16A_CCCTLx register is set to 0x3 or 0x1.

In capture mode (CCBMD = 1)

When the counter value is captured at the external trigger signal (T16A_ATMB_x input) edge selected using CAPBTRG[1:0]/T16A_CCCTLx register, the captured value is loaded to this register. At the same time a capture B interrupt can be generated, thus the captured counter value can be read out in the interrupt handler.

T16A5 Ch.x Comparator/Capture Interrupt Enable Registers (T16A_IENx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
T16A5 Ch.x Comparator/ Capture Interrupt Enable Register (T16A_IENx)	0x30118a 0x30119a (16 bits)	D15–6	–	reserved		–	–	–	0 when being read.	
		D5	CAPBOWIE	Capture B overwrite interrupt enable	1	Enable	0	Disable	0	R/W
		D4	CAPAOWIE	Capture A overwrite interrupt enable	1	Enable	0	Disable	0	R/W
		D3	CAPBIE	Capture B interrupt enable	1	Enable	0	Disable	0	R/W
		D2	CAPAIE	Capture A interrupt enable	1	Enable	0	Disable	0	R/W
		D1	CBIE	Compare B interrupt enable	1	Enable	0	Disable	0	R/W
		D0	CAIE	Compare A interrupt enable	1	Enable	0	Disable	0	R/W

D[15:6] Reserved

D5 CAPBOWIE: Capture B Overwrite Interrupt Enable Bit

Enables or disables capture B overwrite interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting CAPBOWIE to 1 enables capture B overwrite interrupt requests to the ITC. Setting it to 0 disables interrupts.

D4 CAPAOWIE: Capture A Overwrite Interrupt Enable Bit

Enables or disables capture A overwrite interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting CAPAOWIE to 1 enables capture A overwrite interrupt requests to the ITC. Setting it to 0 disables interrupts.

D3 CAPBIE: Capture B Interrupt Enable Bit

Enables or disables capture B interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting CAPBIE to 1 enables capture B interrupt requests to the ITC. Setting it to 0 disables interrupts.

D2 CAPAIE: Capture A Interrupt Enable Bit

Enables or disables capture A interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting CAPAIE to 1 enables capture A interrupt requests to the ITC. Setting it to 0 disables interrupts.

D1 CBIE: Compare B Interrupt Enable Bit

Enables or disables compare B interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting CBIE to 1 enables compare B interrupt requests to the ITC. Setting it to 0 disables interrupts.

D0 CAIE: Compare A Interrupt Enable Bit

Enables or disables compare A interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting CAIE to 1 enables compare A interrupt requests to the ITC. Setting it to 0 disables interrupts.

T16A5 Ch.x Comparator/Capture Interrupt Flag Registers (T16A_IFLGx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
T16A5 Ch.x Comparator/ Capture Interrupt Flag Register (T16A_IFLGx)	0x30118c 0x30119c (16 bits)	D15–6	–	reserved		–	–	–	0 when being read.		
		D5	CAPBOWIF	Capture B overwrite interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	CAPAOWIF	Capture A overwrite interrupt flag			0		R/W		
		D3	CAPBIF	Capture B interrupt flag			0		R/W		
		D2	CAPAIF	Capture A interrupt flag			0		R/W		
		D1	CBIF	Compare B interrupt flag			0		R/W		
		D0	CAIF	Compare A interrupt flag			0		R/W		

D[15:6] Reserved**D5 CAPBOWIF: Capture B Overwrite Interrupt Flag Bit**

Indicates whether the cause of capture B overwrite interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

CAPBOWIF is a T16A5 interrupt flag that is set to 1 when the capture B register is overwritten. CAPBOWIF is reset by writing 1.

D4 CAPAOWIF: Capture A Overwrite Interrupt Flag Bit

Indicates whether the cause of capture A overwrite interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

CAPAOWIF is a T16A5 interrupt flag that is set to 1 when the capture A register is overwritten. CAPAOWIF is reset by writing 1.

D3 CAPBIF: Capture B Interrupt Flag Bit

Indicates whether the cause of capture B interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

CAPBIF is a T16A5 interrupt flag that is set to 1 when the counter value is captured in the capture B register. CAPBIF is reset by writing 1.

D2 CAPAIF: Capture A Interrupt Flag Bit

Indicates whether the cause of capture A interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

CAPAIF is a T16A5 interrupt flag that is set to 1 when the counter value is captured in the capture A register. CAPAIF is reset by writing 1.

D1 CBIF: Compare B Interrupt Flag Bit

Indicates whether the cause of compare B interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

CBIF is a T16A5 interrupt flag that is set to 1 when the counter reaches the value set in the compare B register. CBIF is reset by writing 1.

D0 CAIF: Compare A Interrupt Flag Bit

Indicates whether the cause of compare A interrupt has occurred or not.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

CAIF is a T16A5 interrupt flag that is set to 1 when the counter reaches the value set in the compare A register.

CAIF is reset by writing 1.

16 16-bit Audio PWM Timer (T16P)

16.1 T16P Module Overview

The S1C33L26 incorporate a 16-bit audio PWM timer (T16P) that generates PWM pulses from PCM data. The pulses generated can be directly output to a low pass filter that eliminates quantization noise to shape the output signal into sound waveform. A monophonic audio output system can be implemented simply without an external D/A converter.

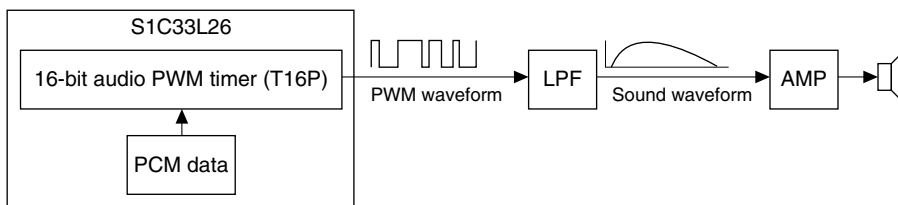


Figure 16.1.1 Audio Output Circuit Using T16P

If no audio output feature is required, T16P can be used as a general-purpose 16-bit timer. The following lists the main functions of T16P.

- Supports 8-bit and 16-bit PCM data with varied sample rates: 8, 16, 22.05, 32, 44.1, and 48 kHz.
- Supports both signed and unsigned PCM data.
- Supports split mode; 16-bit audio data can be split into 10 bits + 6 bits, 9 bits + 7 bits, or 8 bits + 8 bits.
- Supports fine mode to improve the precision of the pulse width.
- Includes a digital volume control unit.
- Programmable count clocks using the prescaler or an external clock
- Built-in two 16-bit data buffers for setting pulse widths (duty cycles) and pulse periods
- Can generate three different types of interrupts and invoke a DMA.

Figure 16.1.2 shows the T16P configuration.

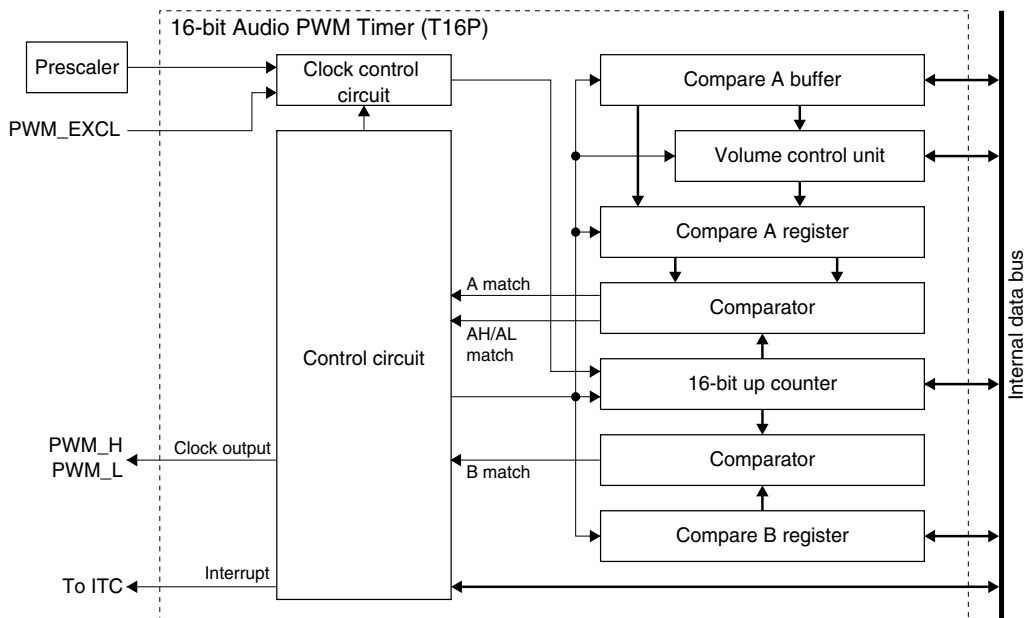


Figure 16.1.2 16-bit Audio PWM Timer (T16P) Configuration

T16P consists of a 16-bit up-counter and two 16-bit compare data buffers/registers.

The 16-bit counter can be reset to 0 via software and counts up using a prescaler output clock or an external clock input from the PWM_EXCL pin.

The compare A buffer is used to store data (PCM data). The stored data is loaded to the compare A register and compared with the counter value to determine the output pulse width. The volume control unit multiplies the PCM data stored in the compare A buffer by the specified volume level set via software before loading to the compare A register. This makes it possible to adjust the volume level to 1/64 through 127/64 as well as muting.

The compare B buffer is used to store data to determine a pulse period. The stored data is loaded to the compare B register and compared with the counter value.

When the counter value reaches the compare data, the timer output signal is inverted to generate PWM waveform.

16.2 T16P Input/Output Pins

Table 16.2.1 lists the input/output pins for the T16P module.

Table 16.2.1 List of T16P Pins

Pin name	I/O	Qty	Function
PWM_EXCL	I	1	T16P external clock input pin Inputs an external clock as the count clock.
PWM_H	O	1	PWM signal output pin In a split mode: Outputs the PWM signal generated from the high-order PCM data bits. In normal mode: Outputs the PWM signal generated from the PCM data.
PWM_L	O	1	PWM signal output pin In a split mode: Outputs the PWM signal generated from the low-order PCM data bits. In normal mode: Fixed at the initial output level (or not used).

The T16P input/output pins (PWM_EXCL, PWM_H, PWM_L) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as T16P input/output pins.

For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

16.3 Setting T16P Operating Conditions

Make the following settings before starting T16P.

1. Select the count clock. (See Section 16.3.1.)
2. Configure the PCM data (resolution, signed/unsigned data format). (See Section 16.3.2.)
3. Select the operating modes (split mode, fine mode). (See Section 16.3.3.)
4. Set the PWM output condition (initial signal level). (See Section 16.3.4.)
5. Set the initial volume level. (See Section 16.4.4.)
6. Set interrupt and/or DMA conditions. (See Section 16.5.)
7. Reset T16P. (See Section 16.4.1.)
8. Configure the T16P input/output pins. (See Section 16.2.)
9. Start T16P. (See Section 16.4.2.)

16.3.1 Count Clock

Either an internal clock or an external clock can be selected as the count clock using CLKSEL/T16P_CTL register. When CLKSEL is set to 0 (default), an internal clock is used; when set to 1, the external clock input to the PWM_EXCL pin is used.

When using an external clock, the external clock cycle must be at least two CPU operating clock cycles.

When an internal clock is used, it can be selected using CLKDIV[3:0]/T16P_CLK register from the 13 types generated by the prescaler (PSC Ch.0) dividing the PCLK1 clock into 1/1 to 1/4,096.

Table 16.3.1.1 Internal Clock (PCLK1 Division Ratio) Selections

CLKDIV[3:0]	Division ratio	CLKDIV[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	Reserved	0x6	1/64
0xd	Reserved	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

Note: Make sure the counter is halted before setting the count clock.

For controlling PSC Ch.0, refer to the “Prescaler (PSC)” chapter.

16.3.2 PCM Data Configuration

The resolution and data format must be specified for manipulating PCM data.

Data resolution

T16P supports 8-bit and 16-bit PCM data. Use RESSEL/T16P_CTL register to select the resolution. When RESSEL is set to 1 (default), 16-bit resolution is selected; when set to 0, 8-bit resolution is selected.

- Notes:**
- 16-bit audio data must be written to address 0x301200 (CMPA[15:0]/T16P_A register) in 16-bit size.
 - 8-bit audio data must be written to address 0x301201 (CMPA[15:8]/T16P_A register) in 8-bit size. Furthermore, select 8 bits + 8 bits split mode to use 8-bit audio data (RESSEL = 0). The PWM pulse will be output from the PWM_H pin and the PWM_L pin is fixed at the level set by INITOL. In this case, the PWM_L pin can be used for a GPIO or other function.

Data format

T16P supports signed and unsigned PCM data. Use SGNSEL/T16P_CTL register to select the data format. When SGNSEL is set to 1 (default), signed data format is selected; when set to 0, unsigned data format is selected.

Note: When signed audio data is selected, CMPA15/T16P_A register is treated as the sign bit for both 16-bit and 8-bit audio data.

16.3.3 Operating Mode Selection

Split mode

When 16-bit PCM data is used, it can be manipulated by splitting into two data units. Use SPLTMD[1:0]/T16P_CTL register to select a split mode.

Table 16.3.3.1 Split Mode Selection

SPLTMD[1:0]	Split mode
0x3	10 bits + 6 bits split mode
0x2	9 bits + 7 bits split mode
0x1	8 bits + 8 bits split mode
0x0	16 bits normal mode

(Default: 0x0)

When a split mode is selected, the split high-order bits (10, 9, or 8 high-order bits) of the compare A data and the low-order bits (6, 7, or 8 low-order bits) are compared with the counter data and the two comparison results generate two PWM output signals. The PWM signal generated from the high-order data bits is output from the PWM_H pin and another generated from the low-order data bits is output from the PWM_L pin. When a split mode or 8-bit PCM data resolution is selected, compare A interrupts cannot be generated.

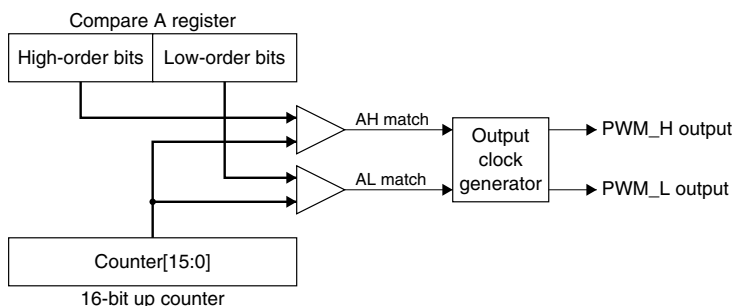


Figure 16.3.3.1 Split Mode

When normal mode is selected (SPLTMD[1:0] = 0x0), 16-bit PCM (compare A) data is compared with the 16-bit counter data and the PWM signal generated is output from the PWM_H pin. If PWM_L output function is enabled, the PWM_L pin is fixed at the initial output level. If this mode and 16-bit PCM data resolution are selected, compare A interrupts can be generated when the counter reaches the compare A data.

Note: When using T16P as a 16-bit timer, set SPLTMD[1:0] to 0x0 (16 bits normal mode).

When using T16P for audio output, set SPLTMD[1:0] to 0x1 (8 bits + 8 bits), 0x2 (9 bits + 7 bits), or 0x3 (10 bits + 6 bits) split mode according to the audio sampling rate.

Fine mode

Normally, compare A data is compared with the counter data at the rising edge of the count clock. When T16P is set to fine mode, the comparisons are performed at both rising and falling edges of the count clock. At this time the compare A data is halved when compared.

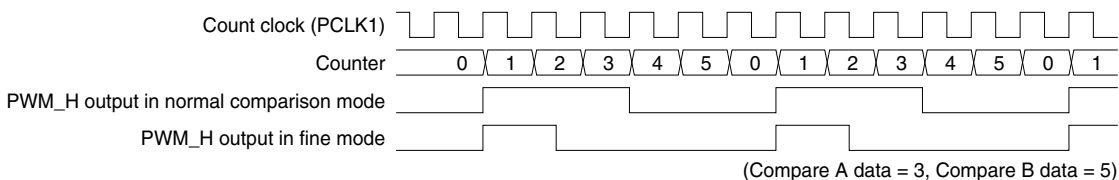


Figure 16.3.3.2 Fine Mode

The fine mode improves the precision of the pulse width. Note, however, that the PCLK1/1 clock can only be used as the count clock in this mode. CLKSEL and CLKDIV[3:0] settings are ineffective.

Set SELFM/T16P_CTL register to 1 to set T16P to fine mode.

The fine mode does not affect the pulse period that is determined with compare B data.

Note: When using A match interrupts while T16P is placed into fine mode, the maximum value of CMPB[15:0] is limited to $2^{15} - 1$ (= 32,767) and the CMPA[15:0] programmable range is limited to 0 to $(2 \times \text{CMPB}[15:0] - 1)$.

However, there is no such limitation when T16P is used only for generating PWM pulses with A match interrupt disabled.

16.3.4 PWM Output Condition Settings

Initial output level

The PWM_H and PWM_L output pins go to the initial output level when the pin function is switched for T16P before starting T16P or when T16P is stopped or reset. Use INITOL/T16P_CTL register to select the initial output level.

When INITOL is 0 (default), the initial output level is low. When INITOL is set to 1, the initial output level is set to high.

Note: Before the pin function is switched for T16P, be sure to set INITOL and then reset the T16P (set PRESET to 1).

16.4 Control and T16P Operations

16.4.1 Resetting T16P

Writing 1 to PRESET/T16P_CTL register resets T16P. The following operations are performed when PRESET is set to 1.

- The counter (CNT_DATA[15:0]/T16P_CNT_DATA register) is reset to 0x0.
- The B match counter is reset to 0x0.
- The initial volume level (VOLSEL[6:0]/T16P_VOL_CTL register) is loaded into the volume control circuit.
- The compare A and B buffers/registers (CMPA[15:0]/T16P_A register, CMPB[15:0]/T16P_B register) are reset to 0x0.
- The buffer empty flag (BUFEF/T16P_INT register) is set to 1. (No interrupt occurs.)
- All other interrupt flags are reset to 0 and interrupt requests are canceled.
- DMA request is canceled if it has been issued.
- The PWM outputs go to the initial output level set by INITOL/T16P_CTL register.

Note: Be sure to reset T16P before the GPIO pins are switched to the PWM_H and PWM_L pins, and before setting PRUN/T16P_RUN register to 1 to start T16P.

16.4.2 Run/Stop Control

To start T16P, write 1 to PRUN/T16P_RUN register.

T16P must be reset (write 1 to PRESET/T16P_CTL register) before writing 1 to PRUN. Resetting the T16P sets the buffer empty flag to 1, but neither an interrupt request nor a DMA request is issued at this point even if the buffer empty interrupt is enabled. Writing 1 to PRUN enables T16P to issue buffer empty interrupts and DMA requests, so that the first audio data can be sent to the buffer in the interrupt handler routine or DMA.

Note: Writing 1 to PRUN does not actually start T16P, because the buffer is still empty. T16P will start after the buffer is filled by an interrupt or DMA.

To stop T16P being run, write 0 to PRUN. The compare data buffers/registers and counter retain the value at stop. The PWM output is fixed at the level set by INITOL. Note that T16P may not stop counting until B match conditions occur (BCNT[3:0] + 1) times.

16.4.3 Setting Compare Data

Compare A buffer

The compare A buffer (CMPA[15:0]/T16P_A register) is used to specify output pulse widths (duty cycles). Set output audio data to this buffer. The buffer data is loaded to the compare A register when the timer starts counting or when a B match occurs specified number of times, and is compared with the counter value. The output signal level is inverted at the beginning of a pulse period and when the counter reaches the compare data stored in the compare A register. This operation converts audio data set to the compare A buffer into a pulse width.

When the data written to the compare A buffer is loaded to the compare A register, the buffer empty interrupt flag (BUFEF/T16P_INT register) is set to 1 and an interrupt occurs if buffer empty interrupts are enabled. Also this cause of interrupt can invoke a DMA transfer. By using this interrupt or DMA transfer, the next output data can be set to the compare A buffer.

When the counter reaches the compare A data, the A match interrupt flag (INTAF/T16P_INT register) is set to 1 and an interrupt occurs if A match interrupts are enabled. This type of interrupts does not occur in split mode or when 8-bit PCM data resolution is selected.

The pulse width set by compare A data is as follows:

In normal comparison mode (SELFM/T16P_CTL register = 0)

Output pulse width = CMPA × Count clock cycle

(CMPA: CMPA[15:0] in normal mode, CMPA[15:n] or CMPA[(n-1):0] in split mode)

In fine mode (SELFM = 1)

Output pulse width = CMPA × PCLK1 cycle × 1/2

(CMPA: CMPA[15:0] in normal mode, CMPA[15:n] or CMPA[(n-1):0] in split mode)

8-bit audio data should be written to CMPA[15:8] in 8-bit size.

Compare B buffer and B match counter

The compare B buffer (CMPB[15:0]/T16P_B register) is used to specify pulse periods. The buffer data is loaded to the compare B register and is compared with the counter value. The output signal level is inverted when the counter reaches the compare data stored in the compare B register (B match). When a B match occurs the counter is reset to 0x0 to start the next pulse period. This operation generates a pulse period according to the compare B data specified.

When the counter reaches the compare B data, the B match interrupt flag (INTBF/T16P_INT register) is set to 1 and an interrupt occurs if B match interrupts are enabled.

The T16P controller includes the B match counter (BCNT[3:0]/T16P_CTL register) to set the sampling rate. Set BCNT[3:0] to 0 to 15. When a B match occurs (BCNT[3:0] + 1) times, the compare A and B buffer data are loaded into the compare A and B registers to start new sampling period.

The pulse period set by compare B data is as follows:

$$\text{Output pulse period} = (\text{CMPB}[15:0] + 1) \times \text{Count clock cycle}$$

$$\text{Sampling period} = (\text{CMPB}[15:0] + 1) \times \text{Count clock cycle} \times (\text{BCNT}[3:0] + 1)$$

16.4.4 Volume Control

T16P includes a volume control function. To use this function, set VOLBPS/T16P_VOL_CTL to 0. The volume control unit multiplies the PCM data stored in the compare A buffer by the specified volume level set using VOLSEL[6:0]/T16P_VOL_CTL register before loading to the compare A register. This makes it possible to adjust the volume level to 1/64 through 127/64 as well as muting.

Table 16.4.4.1 Volume Level Settings

VOLSEL[6:0]	Volume level
0x7f	× 127/64
0x7e	× 126/64
:	:
0x40	× 64/64
:	:
0x2	× 2/64
0x1	× 1/64
0x0	× 0 (mute)

(Default: 0x40)

When VOLBPS is set to 1 (default), the volume control unit is bypassed and compare A data is directly loaded to the compare A register. When 8-bit PCM data is used, the volume control unit should be bypassed by setting VOLBPS to 1.

Table 16.4.4.2 Output Data with or without Volume Control

Input PCM data	Volume control circuit used	Volume control circuit bypassed
Signed 16-bit data: A	$A \times \text{VOLSEL}[6:0] + 0x8000$	$A + 0x8000$
Unsigned 16-bit data: A	$A \times \text{VOLSEL}[6:0]$	A

The volume controlled output data is limited to the range from 0x0 to 0xffff for both signed and unsigned PCM data.

When signed 16-bit PCM data is input, the output data becomes 0x0 if the multiplication results are less than -32768 or 0xffff if the multiplication results exceed +32767.

When unsigned 16-bit PCM data is input, the output data becomes 0xffff if the multiplication results exceed +65535.

Note: It is possible to alter the VOLSEL[6:0] values during playing sound, note, however, that set VOLSEL[6:0] before setting PRESET/T16P_CTL register if the first audio data must be output with volume controlled.

16.4.5 Counter Value

The counter data can be read out from CNT_DATA[15:0]/T16P_CNT_DATA register at any time. Counter data can also be written to CNT_DATA[15:0]. This makes it possible to change the interrupt and/or timer output cycles temporarily.

16.4.6 Timing Charts

Normal mode

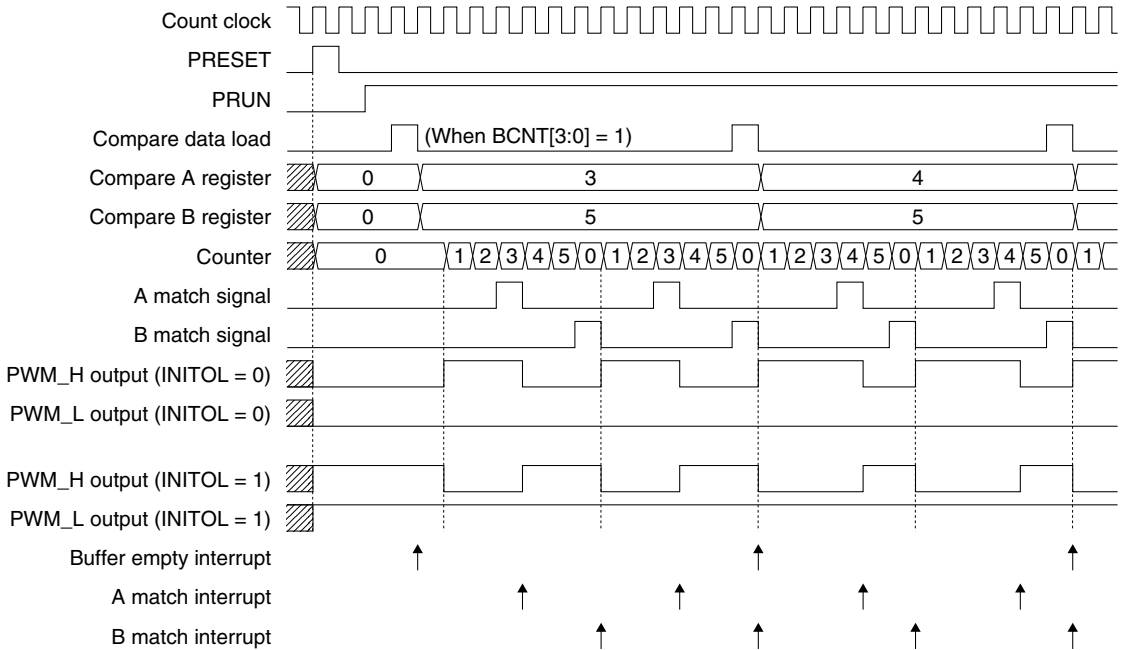


Figure 16.4.6.1 PWM Output Timing Chart 1 (normal mode)

Normal + fine mode

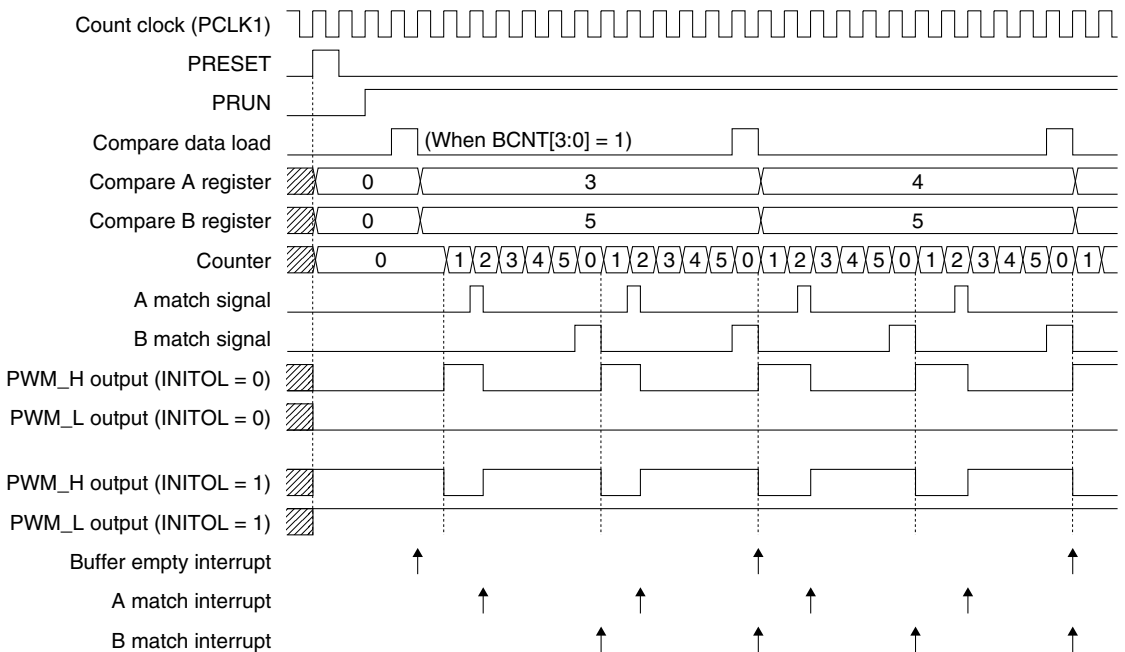


Figure 16.4.6.2 PWM Output Timing Chart 2 (normal + fine mode)

Split mode

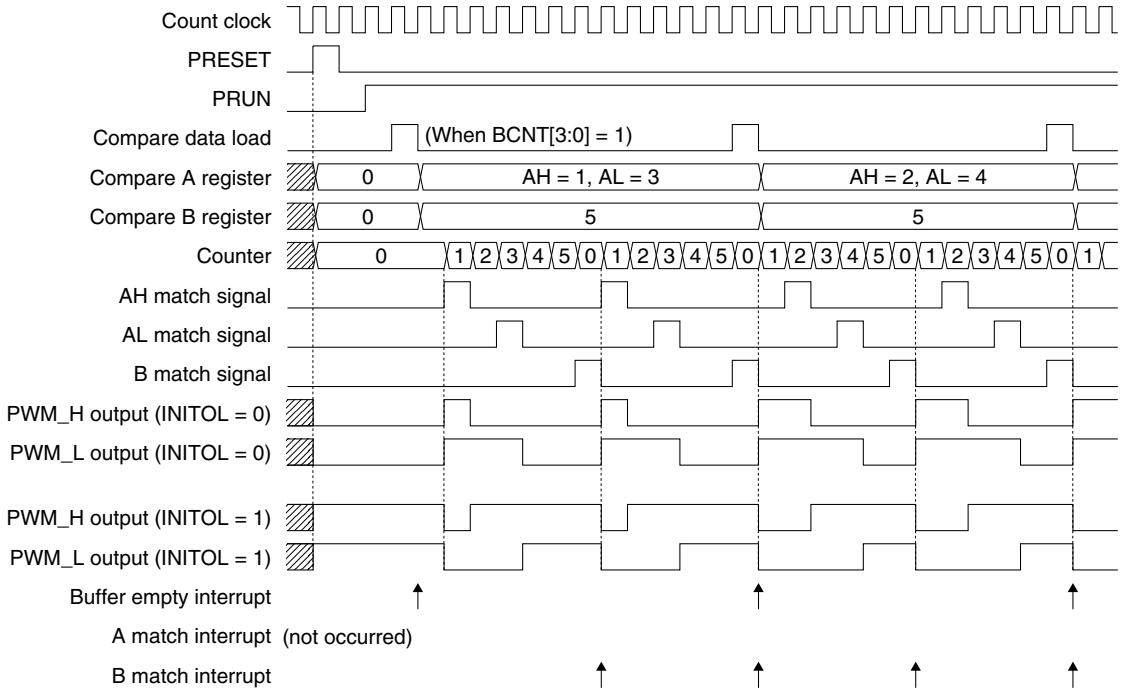


Figure 16.4.6.3 PWM Output Timing Chart 3 (split mode)

Split + fine mode

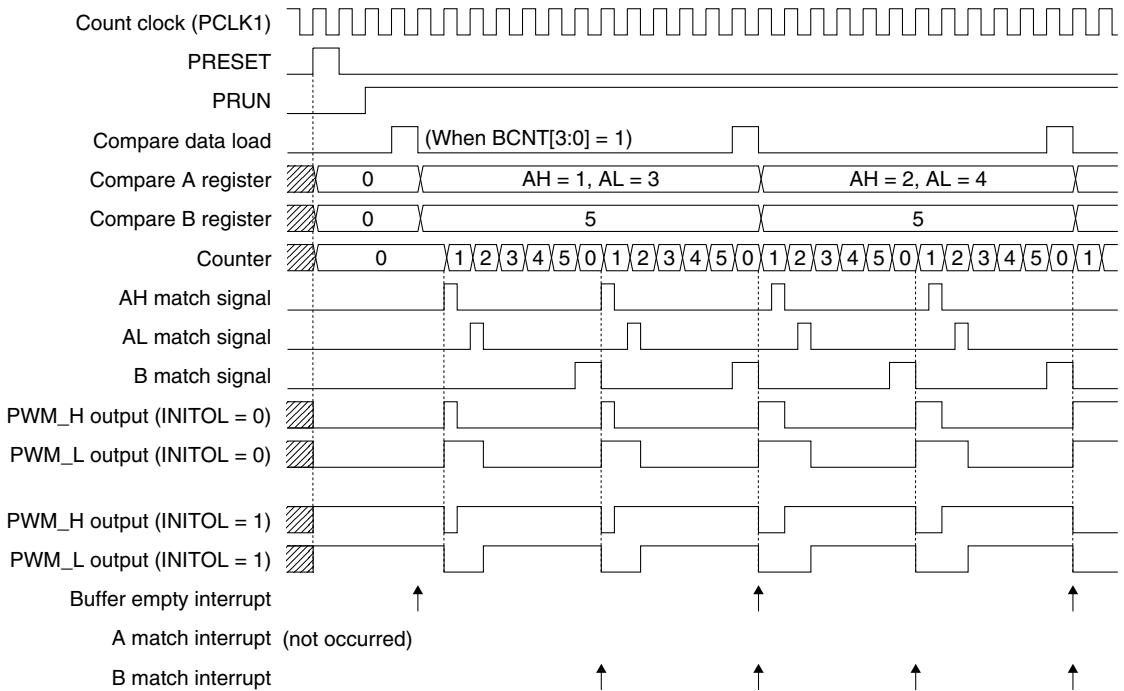


Figure 16.4.6.4 PWM Output Timing Chart 4 (split + fine mode)

16.5 T16P Interrupts and DMA

This section describes the T16P interrupts and invoking DMA.

For more information on interrupt processing and DMA transfer, see the “Interrupt Controller (ITC)” chapter and the “DMA Controller (DMAC)” chapter, respectively.

16.5.1 Interrupts

The T16P module can generate the following three kinds of interrupts:

- Buffer empty interrupt
- A match interrupt
- B match interrupt

T16P outputs a single interrupt signal shared by the above interrupt causes to the interrupt controller (ITC). Read the interrupt flags in the T16P module to identify the interrupt cause that has been occurred.

Buffer empty interrupt

This interrupt request is generated when compare A buffer data are loaded into the compare A registers. It sets the interrupt flag BUFEF/T16P_INT register in the T16P module to 1. The flag is reset by writing 1. Note, however, that the flag will be set to 1 again after resetting if the compare A buffer is still empty. Therefore, write compare data to the compare A buffer before resetting BUFEF.

To use this interrupt, set INTBEEN/T16P_INT register to 1. If INTBEEN is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Note: Resetting the T16P (writing 1 to PRESET) sets the buffer empty flag to 1, but neither an interrupt request nor a DMA request is issued at this point even if the buffer empty interrupt is enabled. Writing 1 to PRUN enables T16P to issue buffer empty interrupts and DMA requests, so that the first audio data can be sent to the buffer in the interrupt handler routine or DMA.

A match interrupt

This interrupt request is generated when the counter reaches the compare A register value during counting. It sets the interrupt flag INTAF/T16P_INT register in the T16P module to 1.

To use this interrupt, set INTAEN/T16P_INT register to 1. If INTAEN is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

When a split mode or 8-bit PCM data resolution is selected, INTAF will not be set and A match interrupts will not be occurred.

B match interrupt

This interrupt request is generated when the counter reaches the compare B register value during counting. It sets the interrupt flag INTBF/T16P_INT register in the T16P module to 1.

To use this interrupt, set INTBEN/T16P_INT register to 1. If INTBEN is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

If the interrupt flag is set to 1 when the interrupt has been enabled, the T16P module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied.

For more information on interrupt control registers and the operation when an interrupt occurs, see the “Interrupt Controller (ITC)” chapter.

- Notes:**
- Reset the interrupt flag before enabling interrupts with the interrupt enable bit to prevent occurrence of unwanted interrupt. The interrupt flag is reset by writing 1.
 - After an interrupt occurs, the interrupt flag in the T16P module must be reset in the interrupt handler routine.

16.5.2 DMA Transfer

The causes of buffer empty interrupts can invoke a DMA. This allows continuous data transfer via the DMAC between memory and the compare A buffer. The buffer empty interrupt signal is output to both the ITC and DMAC. Therefore, DMA transfer can be performed without generating a T16P interrupt.

For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

16.6 Control Register Details

Table 16.6.1 List of T16P Registers

Address	Register name		Function
0x301200	T16P_A	T16P Compare A Buffer Register	Compare A data
0x301202	T16P_B	T16P Compare B Buffer Register	Compare B data
0x301204	T16P_CNT_DATA	T16P Counter Data Register	Counter data
0x301206	T16P_VOL_CTL	T16P Volume Control Register	Enables the volume control and sets a volume level.
0x301208	T16P_CTL	T16P Control Register	Sets the timer operating conditions.
0x30120a	T16P_RUN	T16P Running Control Register	Starts/stops the timer.
0x30120c	T16P_CLK	T16P Internal Clock Control Register	Selects an internal count clock.
0x30120e	T16P_INT	T16P Interrupt Control Register	Controls T16P interrupts.

The T16P registers are described in detail below. These are 16-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

T16P Compare A Buffer Register (T16P_A)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16P Compare A Buffer Register (T16P_A)	0x301200 (16 bits)	D15-0	CMPA[15:0]	Compare A data CMPA15 = MSB CMPA0 = LSB	0x0 to 0xffff	X	R/W	

D[15:0] CMPA[15:0]: Compare A Data Bits

Sets compare A data (PCM data) to be converted to a pulse width. (Default: undefined)

The buffer data is loaded to the compare A register when the timer starts counting or when a B match occurs specified number of times, and is compared with the counter value. The output signal level is inverted at the beginning of a pulse period and when the counter reaches the compare data stored in the compare A register. This operation converts audio data set to the compare A buffer into a pulse width.

When the data written to the compare A buffer is loaded to the compare A register, the buffer empty interrupt flag (BUFEF/T16P_INT register) is set to 1 and an interrupt occurs if buffer empty interrupts are enabled. Also this cause of interrupt can invoke a DMA transfer. By using this interrupt or DMA transfer, the next output data can be set to the compare A buffer.

When the counter reaches the compare data A, the A match interrupt flag (INTAF/T16P_INT register) is set to 1 and an interrupt occurs if A match interrupts are enabled. This type of interrupts does not occur in split mode or when 8-bit PCM data resolution is selected.

The pulse width set by compare A data is as follows:

In normal comparison mode (SELFM/T16P_CTL register = 0)

Output pulse width = $CMPA \times \text{Count clock cycle}$

(CMPA: CMPA[15:0] in normal mode, CMPA[15:n] or CMPA[(n-1):0] in split mode)

In fine mode (SELFM = 1)

Output pulse width = $CMPA \times PCLK1 \text{ cycle} \times 1/2$

(CMPA: CMPA[15:0] in normal mode, CMPA[15:n] or CMPA[(n-1):0] in split mode)

8-bit audio data should be written to CMPA[15:8] in 8-bit size.

T16P Compare B Buffer Register (T16P_B)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16P Compare B Buffer Register (T16P_B)	0x301202 (16 bits)	D15-0	CMPB[15:0]	Compare B data CMPB15 = MSB CMPB0 = LSB	0x0 to 0xffff	X	R/W	

D[15:0] CMPB[15:0]: Compare B Data Bits

Sets compare B data to be converted to a pulse period. (Default: undefined)

The buffer data is loaded to the compare B register and is compared with the counter value. The output signal level is inverted when the counter reaches the compare data stored in the compare B register (B match). When a B match occurs the counter is reset to 0x0 to start the next pulse period. This operation generates a pulse period according to the compare B data specified.

When the counter reaches the compare data B, the B match interrupt flag (INTBF/T16P_INT register) is set to 1 and an interrupt occurs if B match interrupts are enabled.

When a B match occurs the number of times (BCNT[3:0] + 1), the compare A and B buffer data are loaded into the compare A and B registers to start new sampling period.

The pulse period set by compare B data is as follows:

$$\text{Output pulse period} = (\text{CMPB}[15:0] + 1) \times \text{Count clock cycle}$$

$$\text{Sampling period} = (\text{CMPB}[15:0] + 1) \times \text{Count clock cycle} \times (\text{BCNT}[3:0] + 1)$$

T16P Counter Data Register (T16P_CNT_DATA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16P Counter Data Register (T16P_CNT_DATA)	0x301204 (16 bits)	D15-0	CNT_DATA [15:0]	Counter data CNT_DATA15 = MSB CNT_DATA0 = LSB	0x0 to 0xffff	X	R/W	

D[15:0] CNT_DATA[15:0]: Counter Data Bits

The counter data can be read from this register. (Default: undefined)

Furthermore, data can be set to the counter by writing it to this register.

The counter is reset to 0x0 when a B match occurs or when T16P is reset by setting PRESET/T16P_CTL register to 1.

T16P Volume Control Register (T16P_VOL_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16P Volume Control Register (T16P_VOL_CTL)	0x301206 (16 bits)	D15-8	–	reserved	–	–	–	0 when being read.
		D7	VOLBPS	Volume control enable	1 Disable 0 Enable	1	R/W	Effective only for 16-bit data
		D6-0	VOLSEL [6:0]	Volume level select	VOLSEL[6:0] Volume level	0x40	R/W	
					0x7f × 127/64			
					0x7e × 126/64			
					⋮			
					0x40 × 64/64			
					⋮			
					0x2 × 2/64			
					0x1 × 1/64			
					0x0 × 0 (mute)			

D[15:8] Reserved

D7 VOLBPS: Volume Control Enable Bit

Enables or disables the volume control function.

1 (R/W): Disabled (bypassed) (default)

0 (R/W): Enabled

When VOLBPS is set to 0, the volume control unit multiplies the PCM data stored in the compare A buffer by the specified volume level set using VOLSEL[6:0] before loading to the compare A register.

If no volume control is required, set VOLBPS to 1.

Note: When 8-bit PCM data is used, the volume control unit should be bypassed by setting VOLBPS to 1.

D[6:0] VOLSEL[6:0]: Volume Level Select Bits

Selects a volume level when the volume control function is enabled.

Table 16.6.2 Volume Level Settings

VOLSEL[6:0]	Volume level
0x7f	× 127/64
0x7e	× 126/64
:	:
0x40	× 64/64
:	:
0x2	× 2/64
0x1	× 1/64
0x0	× 0 (mute)

(Default: 0x40)

T16P Control Register (T16P_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
T16P Control Register (T16P_CTL)	0x301208 (16 bits)	D15–12	BCNT[3:0]	B match count	0x0 to 0xf		0x0	R/W	Effective only for 16-bit data	
		D11	RESSEL	PCM data resolution select	1 16 bits	0 8 bits	1	R/W		
		D10	SGNSEL	PCM data format select	1 Signed	0 Unsigned	1	R/W		
		D9–8	SPLTMD [1:0]	Split mode select	SPLTMD[1:0]		0x0	R/W		
					0x3	10 bits + 6 bits				
					0x2	9 bits + 7 bits				
					0x1	8 bits + 8 bits				
					0x0	Normal (16 bits)				
		D7	–	reserved	–		–	–		0 when being read.
		D6	SELFM	Fine mode select	1 Fine mode	0 Normal	0	R/W		
		D5	–	reserved	–		–	–		0 when being read.
		D4	INITOL	Initial output level select	1 High	0 Low	0	R/W		
D3	CLKSEL	Input clock select	1 External	0 Internal	0	R/W				
D2	–	reserved	–		–	–	0 when being read.			
D1	PRESET	T16P reset	1 Reset	0 Ignored	0	W				
D0	–	reserved	–		–	–				

D[15:12] BCNT[3:0]: B Match Count Bits

Sets the B match counter. (Default: 0x0)

When a B match occurs (BCNT[3:0] + 1) times, the compare A and B buffer data are loaded into the compare A and B registers.

D11 RESSEL: PCM Data Resolution Select Bit

Selects the PCM data resolution.

1 (R/W): 16 bits (default)

0 (R/W): 8 bits

Notes: • When 8-bit PCM data resolution is selected, no A match interrupt will be generated.

• 16-bit audio data must be written to address 0x301200 (CMPA[15:0]/T16P_A register) in 16-bit size.

• 8-bit audio data must be written to address 0x301201 (CMPA[15:8]/T16P_A register) in 8-bit size. Furthermore, select 8 bits + 8 bits split mode to use 8-bit audio data (RESSEL = 0). The PWM pulse will be output from the PWM_H pin and the PWM_L pin is fixed at the level set by INITOL. In this case, the PWM_L pin can be used for a GPIO or other function.

D10 SGNSEL: PCM Data Format Select Bit

Selects the PCM data format.

1 (R/W): Signed data (default)

0 (R/W): Unsigned data

Note: When signed audio data is selected, CMPA15/T16P_A register is treated as the sign bit for both 16-bit and 8-bit audio data.

D[9:8] SPLTMD[1:0]: Split Mode Select Bits

Selects the split mode for manipulating 16-bit PCM data.

Table 16.6.3 Split Mode Selection

SPLTMD[1:0]	Split mode
0x3	10 bits + 6 bits split mode
0x2	9 bits + 7 bits split mode
0x1	8 bits + 8 bits split mode
0x0	16 bits normal mode

(Default: 0x0)

When a split mode is selected, the split high-order data bits (10, 9, or 8 high-order bits) and low-order data bits (6, 7, or 8 low-order bits) are compared with the counter data and the two comparison results generate two PWM output signals. The PWM signal generated from the high-order data bits is output from the PWM_H pin and another generated from the low-order data bits is output from the PWM_L pin. When a split mode or 8-bit PCM data resolution is selected, compare A interrupts cannot be generated.

Note: SPLTMD[1:0] does not affect 8-bit PCM data.

D7 **Reserved**

D6 **SELFM: Fine Mode Select Bit**

Sets T16P to fine mode.

1 (R/W): Fine mode

0 (R/W): Normal comparison mode (default)

In normal comparison mode, compare A data is compared with the counter data at the rising edge of the count clock. When T16P is set to fine mode, the comparisons are performed at both rising and falling edges of the count clock. At this time the compare A data is halved when compared.

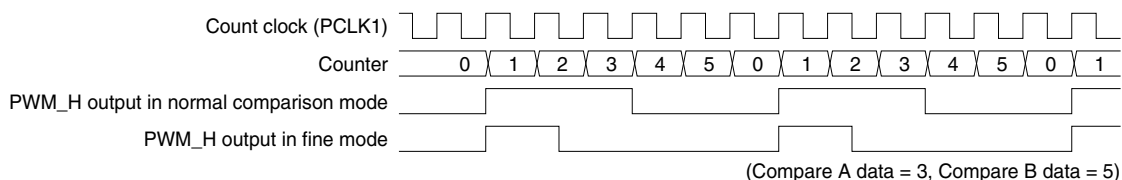


Figure 16.6.1 Fine Mode

The fine mode improves the precision of the pulse width. Note, however, that the PCLK1/1 clock can only be used as the count clock in this mode. CLKSEL and CLKDIV[3:0] settings are ineffective.

The fine mode does not affect the pulse period that is determined with compare B data.

Note: When using A match interrupts while T16P is placed into fine mode, the maximum value of CMPB[15:0] is limited to $2^{15} - 1$ (= 32,767) and the CMPA[15:0] programmable range is limited to 0 to $(2 \times \text{CMPB}[15:0] - 1)$.

However, there is no such limitation when T16P is used only for generating PWM pulses with A match interrupt disabled.

D5 **Reserved**

D4 **INITOL: Initial Output Level Select Bit**

Selects the initial output level for the PWM_H and PWM_L outputs.

1 (R/W): High

0 (R/W): Low (default)

The PWM_H and PWM_L output pins go to the initial output level when the pin function is switched for T16P before starting T16P or when T16P is stopped or reset. When INITOL is set to 0, the initial output level is set to low. When INITOL is set to 1, the initial output level is set to high.

Note: Before the pin function is switched for T16P, be sure to set INITOL and then reset the T16P (set PRESET to 1).

D3 CLKSEL: Input Clock Select Bit

Selects the input clock for the T16P.

1 (R/W): External clock

0 (R/W): Internal clock (default)

When CLKSEL is set to 0, the internal clock (prescaler output) is selected for the count clock. When CLKSEL is set to 1, an external clock (one that is fed from the PWM_EXCL pin) is selected. When using an external clock, the external clock cycle must be at least two CPU operating clock cycles.

D2 Reserved**D1 PRESET: T16P Reset Bit**

Resets T16P.

1 (W): Reset

0 (W): Has no effect

0 (R): Always 0 when read (default)

The following operations are performed when PRESET is set to 1.

- The counter (CNT_DATA[15:0]/T16P_CNT_DATA register) is reset to 0x0.
- The B match counter is reset to 0x0.
- The initial volume level (VOLSEL[6:0]/T16P_VOL_CTL register) is loaded into the volume control circuit.
- The compare A and B buffers/registers (CMPA[15:0]/T16P_A register, CMPB[15:0]/T16P_B register) are reset to 0x0.
- The buffer empty flag (BUFEF/T16P_INT register) is set to 1. (No interrupt occurs.)
- All other interrupt flags are reset to 0 and interrupt requests are canceled.
- DMA request is canceled if it has been issued.
- The PWM outputs go to the initial output level set by INITOL.

Note: Be sure to reset T16P before the GPIO pins are switched to the PWM_H and PWM_L pins, and before setting PRUN/T16P_RUN register to 1 to start T16P.

D0 Reserved**T16P Running Control Register (T16P_RUN)**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16P Running Control Register (T16P_RUN)	0x30120a (16 bits)	D15-1	–	reserved	–	–	–	0 when being read.
		D0	PRUN	T16P run/stop control	1 Run 0 Stop	0	R/W	

D[15:1] Reserved**D0 PRUN: T16P Run/Stop Control Bit**

Starts and stops T16P.

1 (R/W): Run

0 (R/W): Stop (default)

To start T16P, write 1 to PRUN.

T16P must be reset (write 1 to PRESET/T16P_CTL register) before writing 1 to PRUN. Resetting the T16P sets the buffer empty flag to 1, but neither an interrupt request nor a DMA request is issued at this point even if the buffer empty interrupt is enabled. Writing 1 to PRUN enables T16P to issue buffer empty interrupts and DMA requests, so that the first audio data can be sent to the buffer in the interrupt handler routine or DMA.

To stop T16P being run, write 0 to PRUN. The compare data buffers/registers and counter retain the value at stop. The PWM output is fixed at the level set by INITOL. Note that T16P may not stop counting until B match conditions occur (BCNT[3:0] + 1) times.

T16P Internal Clock Control Register (T16P_CLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T16P Internal Clock Control Register (T16P_CLK)	0x30120c (16 bits)	D15-4	–	reserved	–	–	–	0 when being read.	
		D3-0	CLKDIV [3:0]	Counter clock division ratio select (Prescaler output clock)	CLKDIV[3:0] Division ratio	0x0	R/W	Source clock = PCLK1	
					0xf-0xd	reserved			
					0xc	1/4096			
					0xb	1/2048			
					0xa	1/1024			
					0x9	1/512			
					0x8	1/256			
					0x7	1/128			
					0x6	1/64			
					0x5	1/32			
					0x4	1/16			
					0x3	1/8			
					0x2	1/4			
			0x1	1/2					
			0x0	1/1					

D[15:4] Reserved

D[3:0] CLKDIV[3:0]: Counter Clock Division Ratio Select Bits

Selects the counter clock (PCLK1 division ratio) from the 13 different prescaler output clocks when an internal clock is used.

Table 16.6.4 Internal Clock (PCLK1 Division Ratio) Selection

CLKDIV[3:0]	Division ratio	CLKDIV[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	Reserved	0x6	1/64
0xd	Reserved	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

- Notes:**
- Make sure the counter is halted before setting the count clock.
 - When T16P is set to fine mode, CLKDIV[3:0] is ineffective and PCLK1 is directly used as the count clock.

T16P Interrupt Control Register (T16P_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T16P Interrupt Control Register (T16P_INT)	0x30120e (16 bits)	D15-11	–	reserved	–	–	–	0 when being read.	
		D10	BUFEF	Buffer empty interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	X	R/W	Reset by writing 1.	
		D9	INTBF	B match interrupt flag		0	R/W		
		D8	INTAF	A match interrupt flag		0	R/W		
		D7-3	–	reserved	–	–	–	–	0 when being read.
		D2	INTBEEN	Buffer empty interrupt enable	1 Enable 0 Disable	0	R/W		
		D1	INTBEN	B match interrupt enable	1 Enable 0 Disable	0	R/W		
		D0	INTAEN	A match interrupt enable	1 Enable 0 Disable	0	R/W		

D[15:11] Reserved

D10 BUFEF: Buffer Empty Interrupt Flag Bit

Indicates whether the cause of buffer empty interrupt has occurred or not. (Default: undefined)

- 1 (R): Cause of interrupt has occurred
 0 (R): No cause of interrupt has occurred
 1 (W): Flag is reset
 0 (W): Ignored

BUFEF is a T16P interrupt flag that is set to 1 when the compare A buffer data is loaded into the compare A register. BUFEF is reset by writing 1. Note, however, that the flag will be set to 1 again after resetting if the compare A buffer is still empty. Therefore, write compare data to the compare A buffer before resetting BUFEF.

The BUFEF value is undefined at initial reset. However, no buffer empty interrupt request is issued until T16P starts running by setting PRUN/T16P_RUN register is set to 1 even if the interrupt is enabled when BUFEF has been set to 1 at initial reset .

D9 INTBF: B match Interrupt Flag Bit

Indicates whether the cause of B match interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

INTBF is a T16P interrupt flag that is set to 1 when the counter reaches the value set in the compare B register. INTBF is reset by writing 1.

D8 INTAF: A match Interrupt Flag Bit

Indicates whether the cause of A match interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

INTAF is a T16P interrupt flag that is set to 1 when the counter reaches the value set in the compare A register. INTAF is reset by writing 1.

D[7:3] Reserved**D2 INTBEEN: Buffer Empty Interrupt Enable Bit**

Enables or disables buffer empty interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

Setting INTBEEN to 1 enables buffer empty interrupt requests to the ITC. Setting it to 0 disables interrupts.

D1 INTBEN: B Match Interrupt Enable Bit

Enables or disables B match interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

Setting INTBEN to 1 enables B match interrupt requests to the ITC. Setting it to 0 disables interrupts.

D0 INTAEN: A Match Interrupt Enable Bit

Enables or disables A match interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

Setting INTAEN to 1 enables A match interrupt requests to the ITC. Setting it to 0 disables interrupts.

17 Watchdog Timer (WDT)

17.1 WDT Module Overview

The S1C33L26 incorporates a watchdog timer to detect the CPU running uncontrollably.

The features of WDT are listed below.

- 30-bit up counter with a comparator
- Reset or NMI can be generated when the counter reaches the specified value if WDT has not been reset.
- The count clock is selectable from the system clock (PCLK2) and an external clock (T16A_EXCL_0).
- Can output the generated NMI signal (#WDT_NMI) and the comparator output (WDT_CLK).

Figure 17.1.1 shows the WDT configuration.

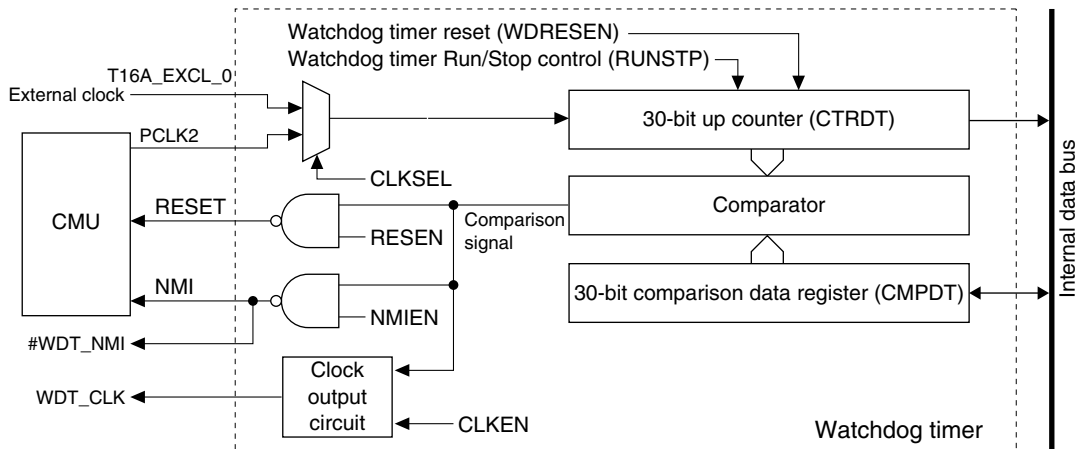


Figure 17.1.1 WDT Configuration

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 PCLK2 clock (= system clock) or external clock input for the 16-bit PWM timer (T16A5) (T16A_EXCL_0) 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.

17.2 WDT Input/Output Pins

Table 17.2.1 lists the input/output pins for the WDT module.

Table 17.2.1 List of WDT Pins

Pin name	I/O	Qty	Function
T16A_EXCL_0	I	1	T16A5 Ch.0/WDT external clock input pin Inputs an external clock as the counter clock.
WDT_CLK	O	1	Watchdog timer clock output pin Outputs the reset/NMI cycle clock generated in the watchdog timer to external devices.
#WDT_NMI	O	1	Watchdog timer NMI output pin Outputs the NMI signal generated in the watchdog timer to external devices.

The WDT input/output pins (T16A_EXCL_0, WDT_CLK, #WDT_NMI) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as WDT input/output pins.

For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

17.3 WDT Operating Clock

The watchdog timer module is clocked by the PCLK2 clock (= system clock) supplied from the CMU. At initial reset, this clock is also selected as the count clock for the watchdog timer.

For more information on clock generation and control, see the “Clock Management Unit (CMU)” chapter.

Note: Even when using an external clock as the count clock for the watchdog timer, PCLK2 is required for watchdog timer operation and access to its control register.

17.4 Control of the Watchdog Timer

17.4.1 Setting Up the Watchdog Timer

Selecting the count clock

The internal clock (PCLK2) or an external clock (T16A_EXCL_0) can be selected as the count clock for the 30-bit up-counter by using CLKSEL/WD_EN register.

Setting CLKSEL to 0 (default) selects the internal clock (PCLK2); setting it to 1 selects an external clock (T16A_EXCL_0).

Setting the NMI/reset generation cycle

The watchdog timer has a 30-bit comparison data register (CMPDT[29:0]/WD_CMP_L/H registers) that can be used to set a cycle in which to generate an NMI or reset signal.

The data set to CMPDT[29:0] 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.

$$\text{NMI generating cycle} = \frac{\text{CMPDT} + 1}{f_{\text{WDTIN}}} \text{ [s]}$$

where

CMPDT = value set to CMPDT[29:0]

f_{WDTIN} = Input clock (PCLK2 or T16A_EXCL_0) frequency [Hz]

Note: Do not set a value equal to or less than 0x1f 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/WD_EN register to 1. To output a reset signal instead, set RESEN/WD_EN register to 1.

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 handling is executed since it has priority over the NMI handling.

The NMI and reset signals are both output as pulses of 32 system 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 or RESEN.

Write protection of watchdog timer registers

The WD_EN, WD_CMP_L, and WD_CMP_H 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 WDPTC[15:0]/WD_PROTECT register in 16-bit access only. Once the registers are rewritten, be sure to write other than 0x96 to WDPTC[15:0] to reapply write protection.

17.4.2 Starting/Stopping the Watchdog Timer

Writing 1 to RUNSTP/WD_EN register starts counting by the watchdog timer; writing 0 stops the watchdog timer. Since RUNSTP exists in the write-protected WD_EN register, write protection must be removed by writing 0x96 to WDPTC[15:0]/WD_PROTECT register before the content of RUNSTP can be altered.

17.4.3 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/WD_CTL register resets the watchdog timer. The up-counter is reset to 0 at this time, then starts counting NMI/reset generation cycles all over again.

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 count value of the up-counter can be read out from CTRDRT[29:0]/WD_CNT_L/H registers at any time.

17.4.4 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/WD_EN register or RESEN/WD_EN register to 0. Otherwise, write 0 to RUNSTP/WD_EN register to stop the watchdog timer before executing the halt instruction. When NMIEN or RESEN disables NMI or reset generation, the watchdog timer continues counting even in HALT mode. To reenale 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 restarting it.

In SLEEP mode

The supply of PCLK2 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/WD_EN register or RESEN/WD_EN register as required.

17.4.5 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/WD_EN register to 1 after setting up the WDT_CLK pin.

Since CLKEN also exists in the write-protected WDT_EN register, write protection must be removed by writing 0x96 to WDPTC[15:0]/WD_PROTECT register 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.

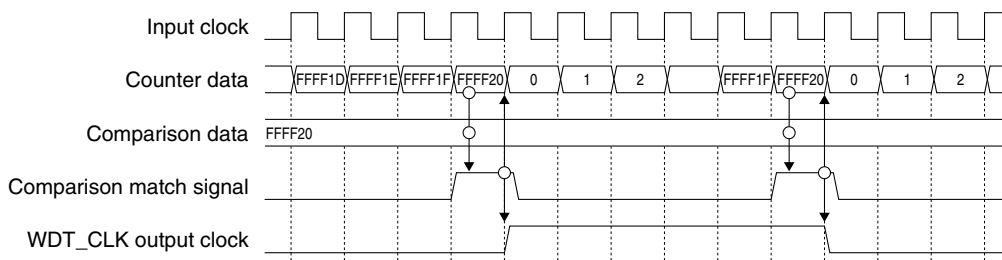


Figure 17.4.5.1 Clock Output of Watchdog Timer

17.4.6 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. Setting NMIEN/WD_EN register 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 input clock cycles.

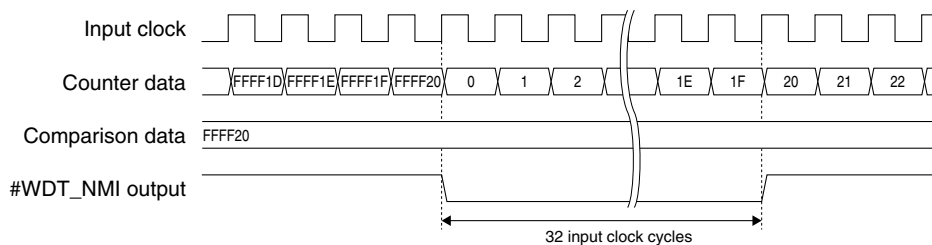


Figure 17.4.6.1 External NMI Output

17.5 Control Register Details

Table 17.5.1 List of WDT Registers

Address	Register name		Function
0x301000	WD_PROTECT	WDT Write Protect Register	Enable WDT register write protection
0x301002	WD_EN	WDT Enable and Setup Register	Configure and start watchdog timer
0x301004	WD_CMP_L	WDT Comparison Data L Register	Comparison data
0x301006	WD_CMP_H	WDT Comparison Data H Register	
0x301008	WD_CNT_L	WDT Count Data L Register	Watchdog timer counter data
0x30100a	WD_CNT_H	WDT Count Data H Register	
0x30100c	WD_CTL	WDT Control Register	Reset watchdog timer

The following describes each WDT register. These are all 16-bit registers.

- Notes:**
- When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.
 - The WD_PROTECT register (0x301000) allows 16-bit access only. Other registers (0x301002 to 0x30100c) allow 8-bit access as well as 16-bit access.

WDT Write Protect Register (WD_PROTECT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Write Protect Register (WD_PROTECT)	0x301000 (16 bits)	D15–0	WDPTC [15:0]	WDT register write protect flag	Writing 0x96 removes the write protection of the WD_EN, WD_CMP_L, and WD_CMP_H registers (0x301002–0x301006). Writing another value set the write protection.	X	W	0 when being read.

D[15:0] WDPTC[15:0]: WDT Register Write Protect Flag Bits

These bits set or clear write protection at addresses 0x301002 to 0x301006.

0x96 (W): Clears write protection

Other than 0x96 (W): Applies write protection (default, indeterminate value)

0x0 (R): Always 0x0 when read

Before altering the WDT_EN, WDT_CMP_L, or WDT_CMP_H register, write 0x96 to WDPTC[15:0] to remove write protection. Setting WDPTC[15:0] 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 WDPTC[15:0], said registers can be rewritten any number of times until WDPTC[15:0] is set to other than 0x96. When the WDT_EN, WDT_CMP_L, or WDT_CMP_H have been rewritten, be sure to write other than 0x96 to WDPTC[15:0] to prevent erroneous writing to the registers.

WDT Enable and Setup Register (WD_EN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Enable and Setup Register (WD_EN)	0x301002 (16 bits)	D15-7	–	reserved	–	–	–	0 when being read.
		D6	CLKSEL	WDT input clock select	1 External clk 0 Internal clk	0	R/W	Write-protected
		D5	CLKEN	WDT clock output control	1 On 0 Off	0	R/W	
		D4	RUNSTP	WDT Run/Stop control	1 Run 0 Stop	0	R/W	
		D3-2	–	reserved	–	–	–	0 when being read.
		D1	NMIEN	WDT NMI enable	1 Enable 0 Disable	0	R/W	Write-protected
		D0	RESEN	WDT RESET enable	1 Enable 0 Disable	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 WDPTC[15:0]/WD_PROTECT register. Once the register has been rewritten, be sure to write other than 0x96 to WDPTC[15:0] to reapply write protection.

D[15:7] Reserved

D6 **CLKSEL: WDT Input Clock Select Bit**

This bit selects the count clock for the watchdog timer.

1 (R/W): External clock (T16A_EXCL_0)

0 (R/W): Internal clock (PCLK2) (default)

Setting this bit to 0 (default) selects the internal clock (PCLK2); setting it to 1 selects the external clock (T16A_EXCL_0).

D5 **CLKEN: WDT 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.

D4 **RUNSTP: WDT 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: WDT 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 system 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: WDT 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 system 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.

WDT Comparison Data L/H Registers (WD_CMP_L, WD_CMP_H)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Comparison Data L Register (WD_CMP_L)	0x301004 (16 bits)	D15-0	CMPDT [15:0]	WDT comparison data CMPDT0 = LSB	0x0 to 0x3ffffff (low-order 16 bits)	0x0	R/W	Write-protected
WDT Comparison Data H Register (WD_CMP_H)	0x301006 (16 bits)	D15-14 D13-0	– CMPDT [29:16]	reserved WDT comparison data CMPDT29 = MSB	– 0x0 to 0x3ffffff (high-order 14 bits)	– 0x0	– R/W	0 when being read. Write-protected

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 WDPTC[15:0]/WD_PROTECT register. Once the registers have been rewritten, be sure to write other than 0x96 to WDPTC[15:0] to reapply write protection.

D[13:0]/0x301006, D[15:0]/0x301004

CMPDT[29:0]: WDT Comparison Data Bits

Sets comparison data. (Default: 0x0)

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.

Note: Do not set a value equal to or less than 0x1f as comparison data.

WDT Count Data L/H Registers (WD_CNT_L, WD_CNT_H)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Count Data L Register (WD_CNT_L)	0x301008 (16 bits)	D15-0	CTRDT [15:0]	WDT counter data CTRDT0 = LSB	0x0 to 0x3ffffff (low-order 16 bits)	X	R	
WDT Count Data H Register (WD_CNT_H)	0x30100a (16 bits)	D15-14 D13-0	– CTRDT [29:16]	reserved WDT counter data CTRDT29 = MSB	– 0x0 to 0x3ffffff (high-order 14 bits)	– X	– R	0 when being read.

D[13:0]/0x30100a, D[15:0]/0x301008

CTRDT[29:0]: WDT Counter Data Bits

The current count value of the 30-bit up-counter can be read out from these registers.

(Default: indeterminate)

WDT Control Register (WD_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Control Register (WD_CTL)	0x30100c (16 bits)	D15-1 D0	– WDRESEN	reserved WDT reset	– 1 Reset 0 ignored	– 0	– W	0 when being read.

D[15:1] Reserved

D0 WDRESEN: WDT 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.

18 Universal Serial Interface (USI)

18.1 USI Module Overview

The S1C33L26 incorporates a universal serial interface (USI) module that can be configured as a UART, SPI, or I²C interface unit by the software switch.

The following shows the main features of USI:

- Supports five interface modes: UART, SPI master, SPI slave, I²C master, and I²C slave modes.
- Contains one-byte receive data buffer and one-byte transmit buffer.
- Supports both MSB first and LSB first modes.
- UART mode
 - Character length: 7 or 8 bits
 - Parity mode: even, odd, or no parity
 - Stop bit: 1 or 2 bits
 - Start bit: 1 bit fixed
 - Parity error, framing error, and overrun error detectable
 - Can generate receive buffer full, transmit buffer empty, and receive error interrupts.
 - Supports DMA transfer.
- SPI master/slave mode
 - Data length: 8 or 9 bits (master mode), or 8 bits fixed (slave mode)
 - Supports both fast and normal modes (master mode), or normal mode only (slave mode).
 - Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
 - Can generate receive buffer full, transmit buffer empty, and receive error interrupts.
 - Supports DMA transfer.
- I²C master/slave mode
 - 7-bit addressing mode (10-bit addressing is possible by software control.)
 - Supports single master configuration only (master mode).
 - Supports clock stretch/wait functions.
 - Can generate operation (start/stop, data transfer, ACK/NAK transfer) completion interrupts and receive error interrupts.

Figure 18.1.1 shows the USI configuration.

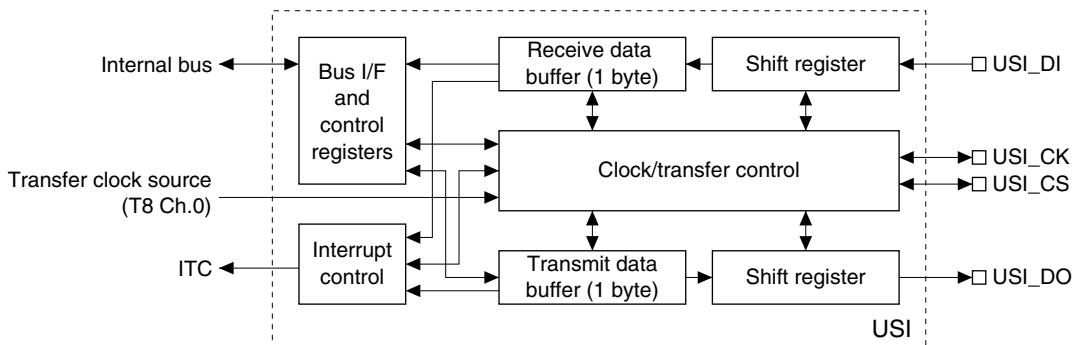


Figure 18.1.1 USI Configuration

18.2 USI Pins

Table 18.2.1 lists the USI input/output pins.

Table 18.2.1 List of USI Pins

Pin name	USI mode	Signal name	I/O	Function
USI_DI	Software reset	–	I	Data input pin
	UART	uart_rx	I	Data input pin
	SPI master	spi_di	I	Inputs serial data sent from an external serial device.
	SPI slave	spi_di	I	
	I ² C master	i2c_sda	I/O	Data input/output pin
I ² C slave	i2c_sda	I/O	Inputs/outputs serial data from/to the I ² C bus. (*1)	
USI_DO	Software reset	–	I	Data input pin
	UART	uart_tx	O	Data output pin
	SPI master	spi_do	O	Outputs serial data sent to an external serial device.
	SPI slave	spi_do	O	
	I ² C master	–	–	Not used
	I ² C slave	–	–	Not used
USI_CK	Software reset	–	I	Data input pin
	UART	–	–	Not used
	SPI master	spi_ck	O	Clock output pin Outputs the SPI clock.
	SPI slave	spi_ck	I	Clock input pin Inputs an external clock.
	I ² C master	i2c_sck	I/O	SCL input/output pin Inputs SCL line status from the I ² C bus. Also outputs the I ² C clock.
	I ² C slave	i2c_sck	I/O	SCL input/output pin Inputs SCL line status from the I ² C bus. Also outputs a clock stretch condition.
USI_CS	Software reset	–	I	Data input pin
	UART	–	–	Not used
	SPI master	–	–	Not used
	SPI slave	#spi_ss	I	SPI slave select signal input pin Low level input to this pin selects USI (in SPI slave mode) as an SPI slave device.
	I ² C master	i2c_sda	I/O	Data input/output pin
	I ² C slave	i2c_sda	I/O	Inputs/outputs serial data from/to the I ² C bus. (*1)

*1: When USI is configured to I²C master or slave mode, either the USI_DI pin or the USI_CS pin can be used as the data input/output pin. Note, however, that both the USI_DI and USI_CS pins cannot be used as the data input/output pin simultaneously.

*2: After a software reset, all USI pins are set for input if USI has not been configured to any mode.

Note: Use a GPIO port to output the slave select signal when USI is configured to SPI master mode.

The USI input/output pins (USI_DI, USI_DO, USI_CK, USI_CS) are shared with I/O ports and are initially set as general-purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as USI input/output pins.

For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

18.3 USI Clock Sources

Operating clock

The USI uses PCLK1 as the operating clock. Therefore, PCLK1 must be supplied from the CMU before starting the USI including setting the control registers. For more information on the PCLK1 supply, refer to the “Clock Management Unit (CMU).”

Transfer clock

When the USI is configured to a UART, SPI master (normal mode), or I²C master device, the source clock for transfer is supplied by the 8-bit timer (T8 Ch.0). Program T8 Ch.0 according to the transfer rate and enable supplying the source clock to the USI module. The USI module divides the source clock to generate the transfer clock (or sampling clock). Be aware that the division ratio in the USI depends on the interface mode.

When the USI is configured to an SPI master (fast mode) device, PCLK1 is used as the source clock.

When the USI is configured to an SPI slave or I²C slave device, the transfer clock is supplied from the external master device. However, SPI slave mode uses PCLK1 and I²C slave mode uses the T8 Ch.0 output clock to generate the sampling signal.

Table 18.3.1 USI Clocks

Clock	Interface mode	Clock source
Operating clock	UART	PCLK1
	SPI master	PCLK1
	SPI slave	PCLK1
	I ² C master	PCLK1
	I ² C slave	PCLK1
Transfer/sampling clock source (division ratio in USI)	UART	T8 Ch.0 (f _{SOURCE} /8)
	SPI master	Normal mode: T8 Ch.0 (f _{SOURCE} /2) Fast mode: PCLK1 (f _{PCLK1})
	SPI slave	PCLK1 (f _{PCLK1} /4) for sampling
	I ² C master	T8 Ch.0 (f _{SOURCE} /8)
	I ² C slave	T8 Ch.0 (f _{SOURCE}) for sampling

UART mode, I²C master mode

$$\text{bps} = \text{fsys_clk} \times \text{DF} / \{(\text{TR} + 1) \times 8 + \text{TFMD}\}$$

$$\text{TR} = (\text{fsys_clk} \times \text{DF} / \text{bps} - \text{TFMD} - 8) / 8$$

SPI master mode

$$\text{bps} = \text{fsys_clk} \times \text{DF} / \{(\text{TR} + 1) \times 2 + \text{TFMD}\}$$

$$\text{TR} = (\text{fsys_clk} \times \text{DF} / \text{bps} - \text{TFMD} - 2) / 2$$

f_{SOURCE}: T8 Ch.0 output clock frequency [Hz]

f_{sys_clk}: System clock frequency [Hz]

bps: Transfer rate [bps]

DF: Division ratio set by DF[3:0]/T8_CLK0 register (T8 Ch.0)

TR: Reload data to be set to the T8_TR0 register (T8 Ch.0)

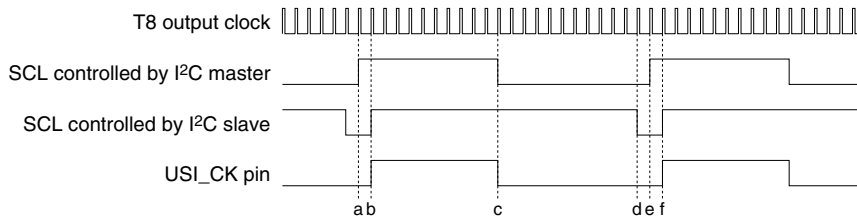
TFMD: Fine mode set value at TFMD[3:0]/T8_CTL0 register (T8 Ch.0)

Example: UART mode, transfer rate = 115,200 bps, system clock = 33 MHz, DF[3:0]/T8_CLK0 register setting (T8 Ch.0) = 1/1, TFMD[3:0]/T8_CTL0 register setting (T8 Ch.0) = 14

$$\text{TR} = (33,000,000 \times 1 / 115,200 - 14 - 8) / 8 = 33.05 (= 0x21)$$

For more information on controlling the T8 module, refer to the “8-bit Timers (T8)” chapter.

Note: When the USI is set to I²C slave mode, i2c_sck (I²C clock) is supplied from the external I²C master. The T8 output clock frequency (f_{SOURCE}) should be determined according to the i2c_sck frequency.

Figure 18.3.1 I²C Clock in I²C Slave Mode

Tbf = Ti2c_baud_rate

Tbc = Ti2c_baud_rate_high

Tcf = Ti2c_baud_rate_low

Tce: The I²C master occupies the SCL line by driving it to low.

Tac: The I²C master releases the SCL line.

Tdf: In order to finish the internal operations, the I²C slave occupies the SCL line for two source clock (T8 output clock) cycles by driving it to low after detecting that the I²C master drives the SCL line to low.

18 UNIVERSAL SERIAL INTERFACE (USI)

The T8 output clock frequency (f_{SOURCE}) must be set so that the conditions shown below are satisfied.

$$f_{SOURCE} > 3/T_{bc}$$

$$f_{SOURCE} > 4/T_{ce}$$

Be aware that the actual SCL signal will be delayed, as the I²C slave forcibly drives the SCL line to low. The figure below shows an example in which the timing becomes worse.

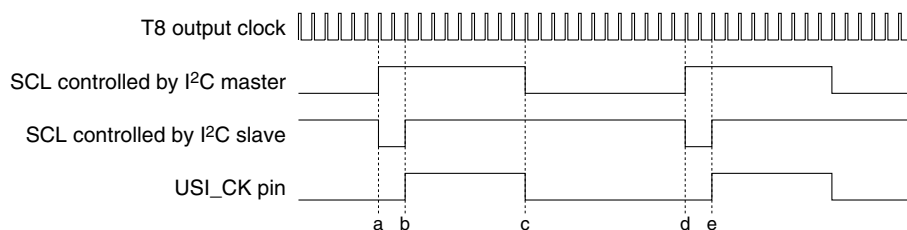


Figure 18.3.2 Example of Delayed I²C Clock

18.4 USI Module Settings

Make the following settings before starting data transfers using the USI module.

- (1) Program the clock source module to supply the clock required to the USI module. (See Section 18.3.)
- (2) Reset the USI module.
- (3) Set the USI interface mode and a general condition (MSB first/LSB first) to be applied to all interface modes.
- (4) Set the data format and operating conditions for the interface mode selected.
- (5) Set interrupt and DMA transfer conditions if necessary. (See Section 18.7.)
- (6) Select the port functions to be used for USI according to the interface mode. (See Section 18.2 and the “I/O Ports (GPIO)” chapter.)

18.4.1 USI Module Software Reset

Writing 0x0 to USIMOD[2:0]/USI_GCFG register resets the USI module circuits. Be sure to perform software reset before setting the interface mode.

18.4.2 Interface Mode

The USI module provides five serial interface functions shown in Section 18.1. Each channel can be configured to one of them using the USIMOD[2:0]/USI_GCFG register.

Table 18.4.2.1 Interface Mode Selection

USIMOD[2:0]	Interface mode
0x5	I ² C slave
0x4	I ² C master
0x3	SPI slave
0x2	SPI master
0x1	UART
0x0	Software reset

(Default: 0x0)

Note: Be sure to perform software reset and set the interface mode before changing other USI configurations.

18.4.3 General Settings for All Interface Modes

MSB first/LSB first selection

Use LSBFST/USI_GCFG register to select whether the data MSB or LSB is input/output first.

LSB first is selected when LSBFST is set to 0 (default). MSB first is selected when LSBFST is set to 1.

18.4.4 Settings for UART Mode

When the USI is used in UART mode, configure the data length, stop bit, and parity bit. The start bit length is fixed at 1 bit.

Data length

Use UCHLN/USI_UCFG register to select the data length. Setting UCHLN to 0 (default) configures the data length to 7 bits. Setting UCHLN to 1 configures it to 8 bits.

Stop bit

Use USTPB/USI_UCFG register to select the stop bit length. Setting USTPB to 0 (default) configures the stop bit length to 1 bit. Setting USTPB to 1 configures it to 2 bits.

Parity bit

Use UPREN/USI_UCFG register to select whether the parity function is enabled or not. Setting UPREN to 0 (default) disables the parity function. In this case, no parity bit will be added to transfer data and receive data will not be checked for parity. Setting UPREN to 1 enables the parity function. In this case, a parity bit will be added to transfer data and receive data will be checked for parity.

When the parity function is enabled, the parity mode should be selected using UPMD/USI_UCFG register. Setting UPMD to 0 (default) adds a parity bit and checks for odd parity. Setting UPMD to 1 adds a parity bit and checks for even parity.

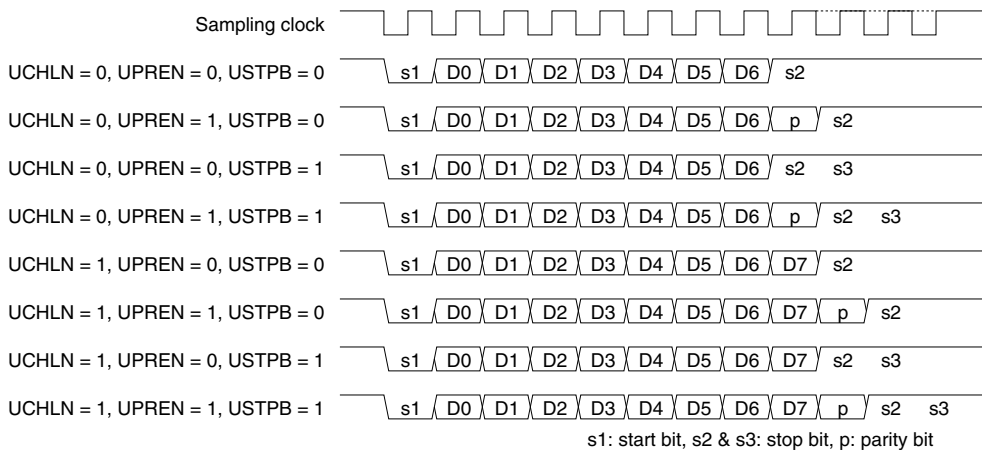


Figure 18.4.4.1 Transfer Data Format in UART Mode (LSB first)

18.4.5 Settings for SPI Mode

When the USI is used in SPI mode (master or slave), configure the SPI clock polarity/phase. When used in SPI master mode, select the clock mode and data length.

Note that the data length in SPI slave mode is fixed at 8 bits.

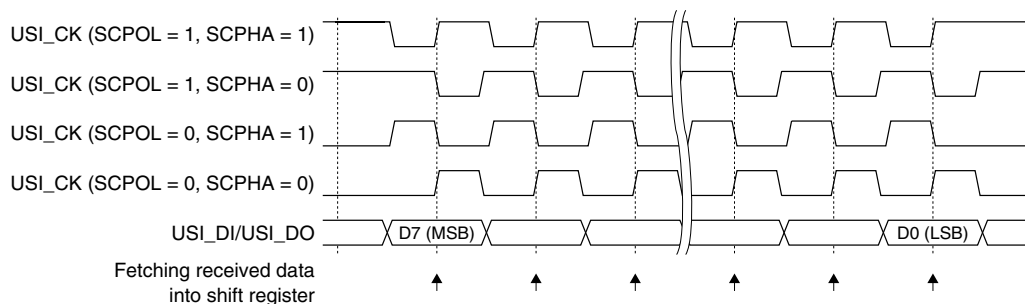
SPI clock polarity and phase settings (master mode and slave mode)

Use SCPOL/USI_SCFG register to select the SPI clock polarity. Setting SCPOL to 1 treats the SPI clock as active low. Setting it to 0 (default) treats it as active high.

The SPI clock phase can be selected using SCPHA/USI_SCFG register.

These control bits set transfer timing as shown in Figure 18.4.5.1.

Master mode



Slave mode

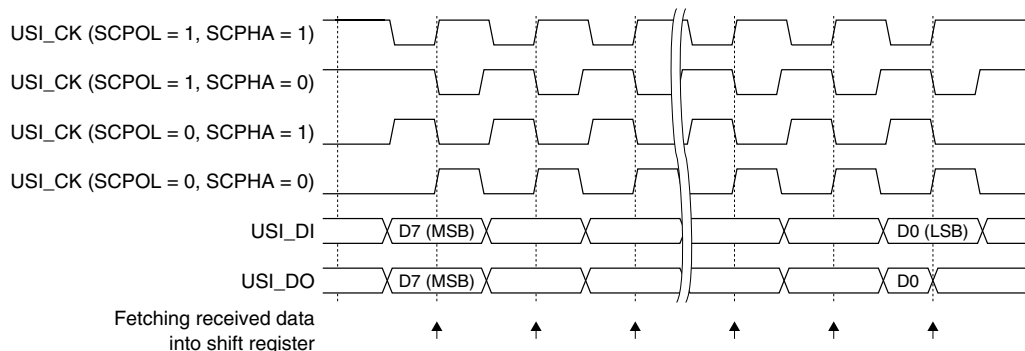


Figure 18.4.5.1 Clock and Data Transfer Timing (MSB first)

Clock mode (master mode only)

In SPI master mode, either normal or fast clock mode can be selected using SFSTMOD/USI_SCFG register. Setting SFSTMOD to 0 (default) places the USI into normal mode and the USI generates the transfer clock by dividing the T8 output by 2. Setting SFSTMOD to 1 places the USI into fast mode and the USI uses PCLK1 supplied from the CMU directly as the transfer clock. The fast mode does not use the T8.

The SPI slave mode uses the T8 output clock for generating the sampling clock.

Data length (master mode only)

In SPI master mode, the data length can be selected using SCHLN/USI_SCFG register. Setting SCHLN to 0 (default) configures the data length to 8 bits.

Setting SCHLN to 1 configures the data length to 9 bits. In 9-bit mode, 8-bit data is prefixed with a command bit (1 bit). The command bit is used for controlling the SPI LCD controller connected to the USI. The command bit value to be transmitted can be specified using SCMD/USI_SCFG register. Setting SCMD to 1 configures the command bit to high. Setting SCMD to 0 configures the command bit to low.

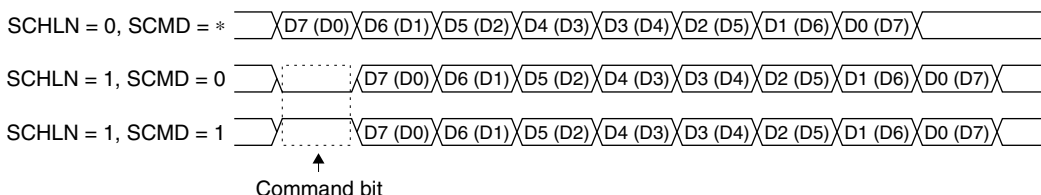


Figure 18.4.5.2 9-bit Transfer Data Format in SPI Master Mode

The data length in SPI slave mode is fixed at 8 bits.

18.4.6 Settings for I²C Mode

The I²C mode does not need to set data format and other I conditions. The data length in I²C mode is fixed at 8 bits.

18.5 Data Transfer Control

This section describes how to control data transfers. The following explanations assume that the configurations described above and interrupt/DMA settings have already been finished.

18.5.1 Data Transfer in UART Mode

Data transmission

To start data transmission in UART mode, write the transmit data to the transmit data buffer (TD[7:0]/USI_TD register).

The buffer data is sent to the transmit shift register, and the start bit is output from the USI_DO pin. The data in the shift register is then output in sequence. Following output of the eighth data bit, the parity bit (if parity is enabled) and the stop bit are output.

The transmitter circuit includes two status flags: UTDIF/USI_UIF register and UTBSY/USI_UIF register.

The UTDIF flag indicates the transmit data buffer status. This flag is set to 1 indicating that the transmit data buffer becomes empty when data written to the transmit data buffer is sent to the transmit shift register. UTDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 18.7). Write subsequent data to the transmit data buffer to start the following transmission using this interrupt or DMA. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. If an interrupt or DMA is not used for transmission, be sure to confirm that the transmit data buffer is empty before writing transmit data. Writing data before UTDIF has been set will overwrite earlier transmit data inside the transmit data buffer. After UTDIF is set to 1, it can be reset to 0 by writing 1.

The UTBSY flag indicates the USI status in UART mode. This flag switches to 1 when transmit data is written to the transmit buffer and reverts to 0 after both the shift register and transmit buffer become empty.

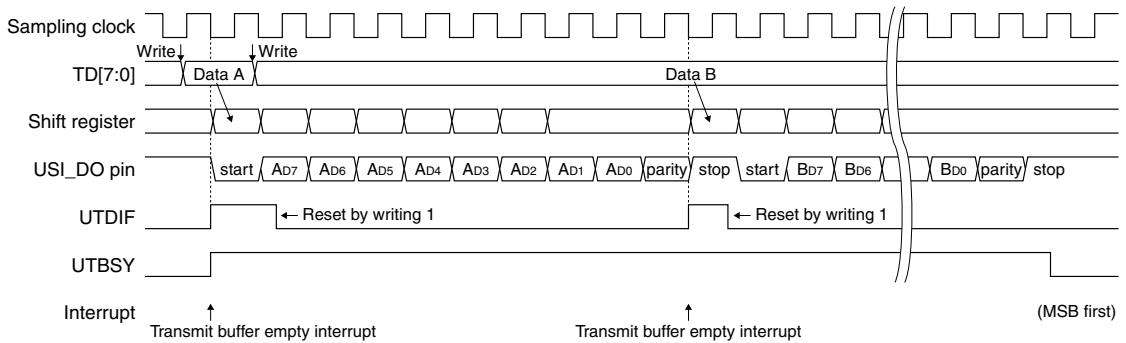


Figure 18.5.1.1 Data Transmission Timing Chart (UART mode)

Data reception

When the external serial device sends a start bit, the receiver circuit detects its low level and starts sampling the following data bits. Once the 8-bit data has been received into the shift register, the received data is loaded into the receive data buffer (RD[7:0]/USI_RD register). If parity checking is enabled, the receiver circuit checks the received data at the same time by checking the parity bit received immediately after the eighth data bit.

The receiver circuit includes two status flags: URDIF/USI_UIF register and URBSY/USI_UIF register.

The URDIF flag indicates the receive data buffer status. This flag is set to 1 indicating that the received data can be read out when data received in the shift register is loaded to the receive data buffer. URDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 18.7). Read the received data from the receive data buffer using this interrupt or DMA. The receive data buffer size is 1 byte, therefore the received data must be read before the subsequent data reception has completed. Furthermore, URDIF must be reset by writing 1. If the next reception is completed when URDIF is 1 and the receive data buffer (USI_RD register) is not read, an overrun error occurs (at the time stop bit has been received).

The URBSY flag indicates the shift register status. This flag is set to 1 while data is being received in the shift register and reverts to 0 once the received data is loaded to the receive data buffer. Read this flag to check whether the receiver circuit is operating or at standby.

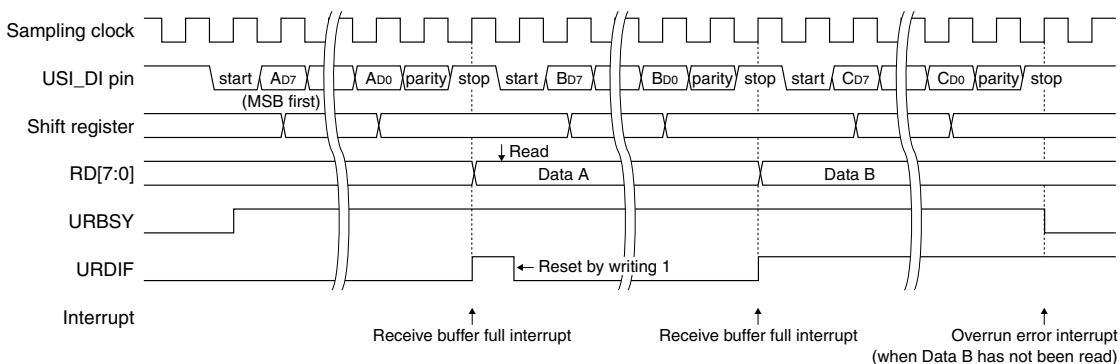


Figure 18.5.1.2 Data Receiving Timing Chart (UART mode)

18.5.2 Data Transfer in SPI Mode

Data transmission

To start data transmission in SPI mode, write the transmit data to the transmit data buffer (TD[7:0]/USI_TD register).

The buffer data is sent to the transmit shift register. In SPI master mode, the module starts clock output from the USI_CK pin. In SPI slave mode, the module awaits clock input from the USI_CK pin. The data in the shift register is shifted in sequence at the clock rising or falling edge (see Figure 18.4.5.1) and sent from the USI_DO pin.

The SPI controller includes two status flags for transfer control: STDIF/USI_SIF register and SSIF/USI_SIF register.

The STDIF flag indicates the transmit data buffer status. STDIF is set to 1 indicating that the transmit data buffer becomes empty when data written to the transmit data buffer is sent to the transmit shift register. STDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 18.7). Write subsequent data to the transmit data buffer to start the following transmission using this interrupt or DMA. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. If an interrupt or DMA is not used for transmission, be sure to confirm that the transmit data buffer is empty before writing transmit data. Writing data before STDIF has been set will overwrite earlier transmit data inside the transmit data buffer.

In SPI master mode, the SSIF flag indicates the USI status. This flag switches to 1 when transmit data is written to the transmit buffer and reverts to 0 after both the shift register and transmit buffer become empty. Read this flag to check whether the SPI controller is operating or at standby.

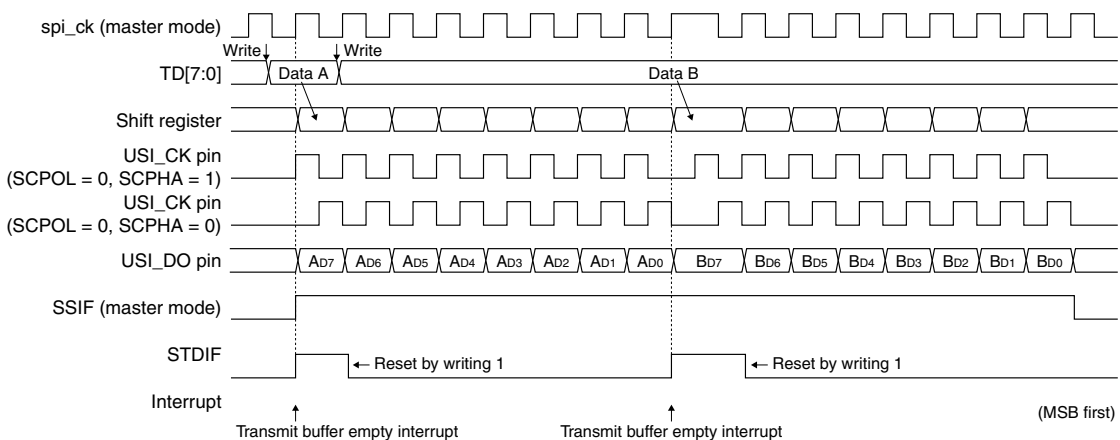


Figure 18.5.2.1 Data Transmission Timing Chart (SPI mode)

Data reception

In SPI master mode, write dummy data to the transmit data buffer. Writing to the transmit data buffer creates the trigger for reception as well as transmission start. Writing actual transmit data enables simultaneous transmission and reception. This starts the SPI clock output from the USI_CK pin.

In SPI slave mode, the module waits until the clock is input from the USI_CK pin. There is no need to write to the transmit data buffer if no transmission is required. The receiving operation is started by the clock input from the master device. If data is transmitted simultaneously, write transmit data to the transmit data buffer before the clock is input.

The data is received in sequence in the shift register at the SPI clock edge (see Figure 18.4.5.1). The received data is loaded into the receive data buffer once the 8 bits of data are received in the shift register.

The received data in the buffer can be read from RD[7:0]/USI_RD register.

The SPI controller includes two status flags for transfer control: SRDIF/USI_SIF register and SSIF/USI_SIF register.

The SRDIF flag indicates the receive data buffer status. This flag is set to 1 when the data received in the shift register is loaded into the receive data buffer, indicating that the received data can be read out. SRDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 18.7). Read the received data from the receive data buffer using this interrupt or DMA. The receive data buffer size is 1 byte, therefore the received data must be read before the subsequent data reception has completed. Furthermore, SRDIF must be reset by writing 1. While SRDIF is set to 1, the next received data will not be transferred from the shift register to the receive data buffer (the first byte data exists in the receive data buffer and the second byte data exists in the shift register). An overrun error occurs if the third byte data is received in this condition, as the second byte data in the shift register is corrupted (an overrun error occurs at the time the first bit of the third byte is fetched).

In SPI master mode, the SSIF flag indicates the shift register status. This flag switches to 1 at the beginning of data reception and reverts to 0 once the data is received. Read this flag to check whether the SPI controller is operating or at standby.

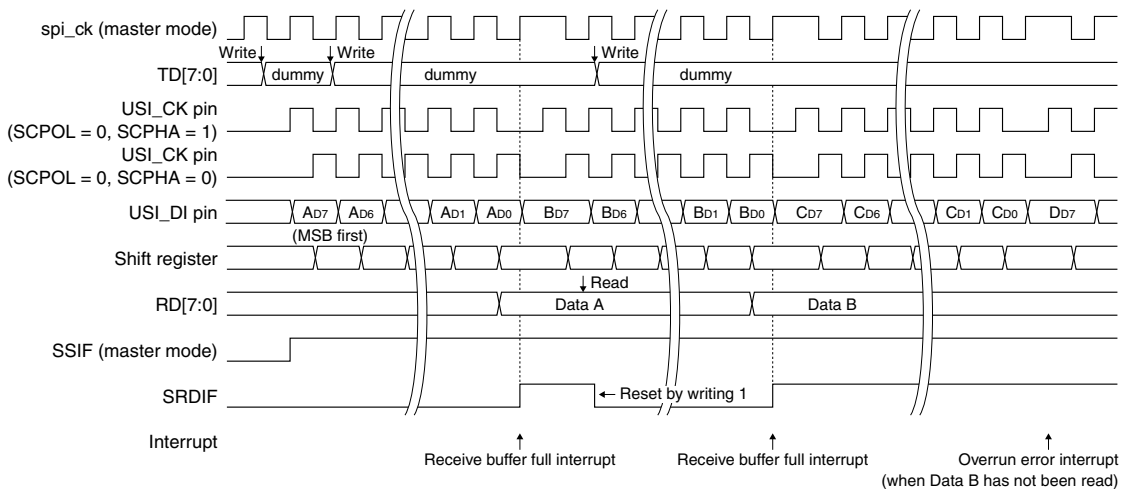


Figure 18.5.2.2 Data Receiving Timing Chart (SPI mode)

Slave select signal

In SPI slave mode, data transmission/receiving operations are enabled when the master device's slave select signal input to the USI_CS pin is low. When the slave select signal is high, the SPI controller does not start data transfer even if the clock is input to the USI_CK pin from the master device. The slave select signal status can be checked using SSIF/USI_SIF register (it functions as the shift register status flag in SPI master mode). SSIF goes 1 when the slave select signal is inactive (high); it goes 0 when the slave select signal is active (low).

If a slave select output is required in SPI master mode, use a general-purpose I/O port and control its output with software.

18.5.3 Data Transfer in I²C Mode

Control method in I²C master mode

Data transfer in I²C master mode is controlled using IMTGMOD[2:0]/USI_IMTG register and IMTG/USI_IMTG register. Select an I²C master operation using IMTGMOD[2:0] and write 1 to IMTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

Table 18.5.3.1 Trigger List in I²C Master Mode

IMTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception
0x2	Data transmission
0x1	Stop condition
0x0	Start condition

(Default: 0x0)

Writing 1 to IMTG sets IMBSY/USI_IMIF register to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, IMBSY is reset to 0. At the same time, the interrupt flag (IMIF/USI_IMIF register) is also set to 1. After an interrupt occurs, read the status bits (IMSTA[2:0]/USI_IMIF register) to check the operation finished. Then clear IMIF by writing 1. IMSTA[2:0] will be automatically cleared to 0x0.

Table 18.5.3.2 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

Data transmission in I²C master mode

The following describes the data transmission procedure in I²C master mode.

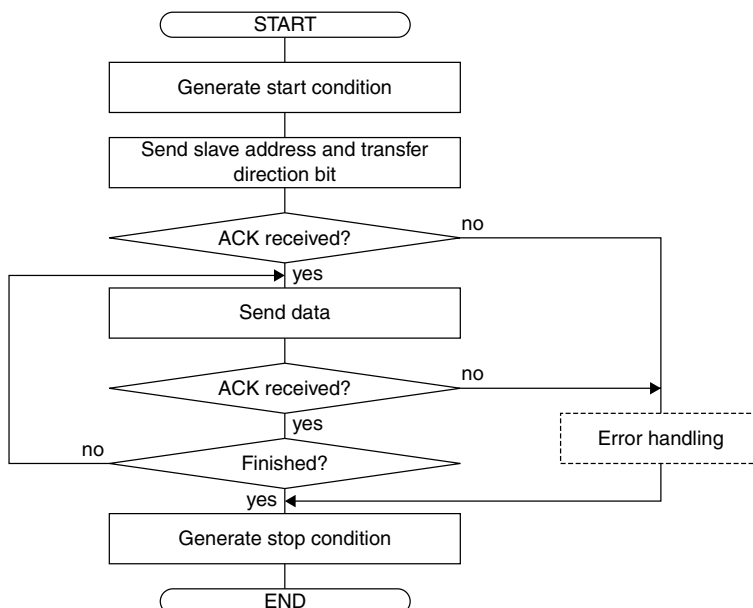
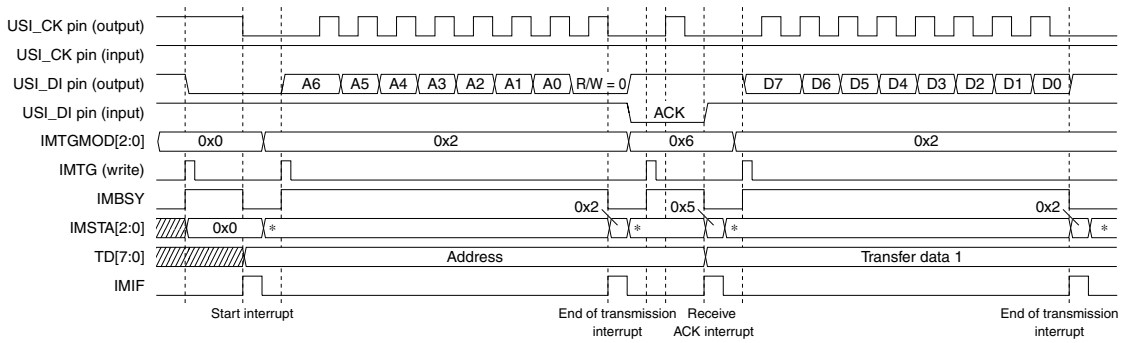
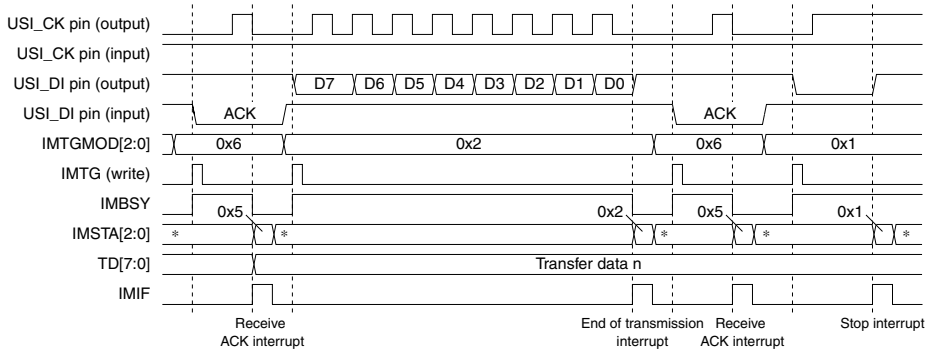


Figure 18.5.3.1 I²C Master Data Transmission Flow Chart



(1) Start condition → Data transmission



(2) Data transmission → Stop condition

* When IMIF is cleared via software, IMSTA[2:0] is also cleared to 0x0.

Figure 18.5.3.2 I²C Master Data Transmission Timing Chart

(1) Generating start condition

I²C data transfer starts when the I²C master device generates a start condition. The start condition applies when the SCL line is maintained at high and the SDA line is pulled down to low.

To generate a start condition in this I²C master, set IMTGMOD[2:0] to 0x0 (default) and write 1 to IMTG.

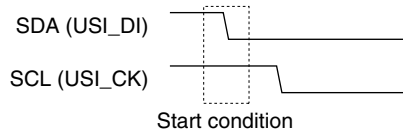


Figure 18.5.3.3 Start Condition

IMBSY is set to 1 while a start condition is being generated. When the start condition is generated, IMBSY is reset to 0 and IMSTA[2:0] is set to 0x0. The I²C bus is busy from this point on.

Note: Other operations cannot be started before a start condition is generated.

(2) Sending slave address and transfer direction bit

After a start condition has been generated, send the address of the slave device to be communicated and a transfer direction bit. I²C slave addresses are either 7-bit or 10-bit. This module uses an 8-bit transfer data buffer to send the slave address and transfer direction bit, enabling single transfers in 7-bit address mode. In 10-bit mode, data is sent twice or three times under software control. Figure 18.5.3.4 shows the configuration of the address data.

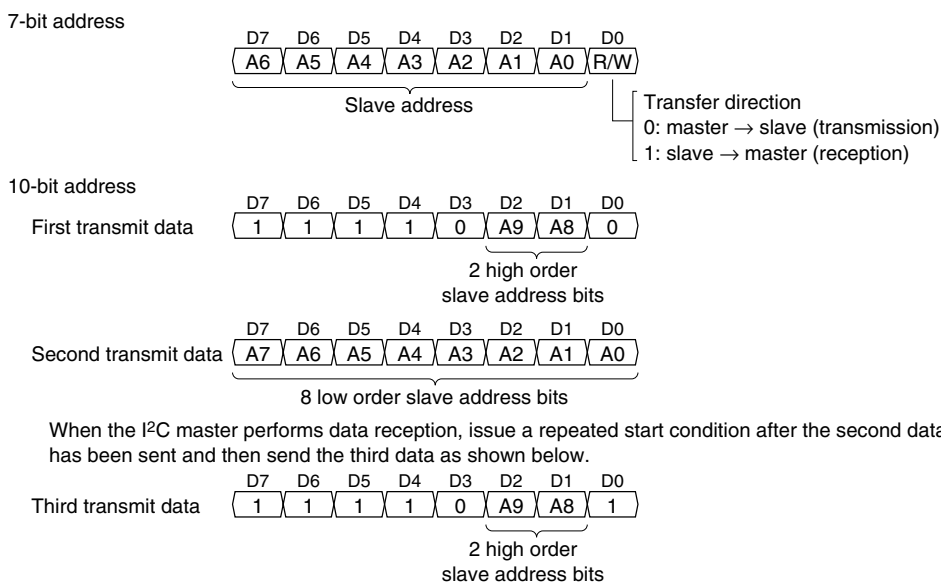


Figure 18.5.3.4 Transmit Data Specifying Slave Address and Transfer Direction

The transfer direction bit indicates the data transfer direction after the slave address has been sent. Set this bit to 0 when sending data from the master to the slave.

To send a slave address, set the address with the transfer direction bit to the transmit data buffer (TD[7:0]/USI_TD register). Then set IMTGMOD[2:0] to 0x2 and write 1 to IMTG.

To send a 10-bit address, execute this procedure twice or three times as shown in Figure 18.5.3.4.

Writing 1 to IMTG sets IMBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x2. Confirm that the slave address (each byte) has been sent by reading IMBSY or using an interrupt.

After a slave address has been sent, the selected slave device sends back an ACK by pulling down the SCL line to low. If the SCL line maintains high, it is regarded as a NAK. In this case, the I²C controller cannot communicate with the slave device specified.

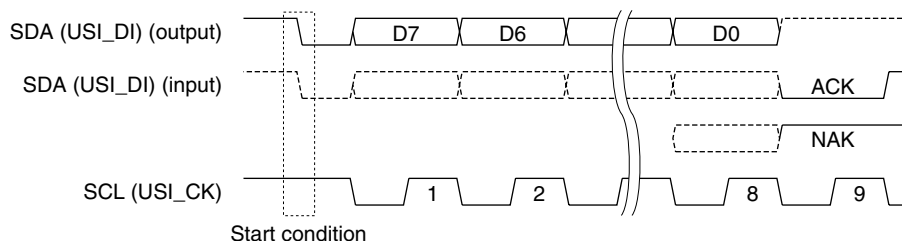


Figure 18.5.3.5 ACK and NAK

It is necessary to check that an ACK has been received before sending data. To do this, set IMTGMOD[2:0] to 0x6 and write 1 to IMTG after the slave address has been sent.

IMBSY is set to 1 while an ACK/NAK is being detected and it reverts to 0 when the detection has completed. Receiving an ACK sets IMSTA[2:0] to 0x5; receiving a NAK sets it to 0x6. Check IMSTA[2:0] after confirming IMBSY or using an interrupt. When an ACK has been received, perform data transmission. When a NAK has been received, perform an error handling.

(3) Data transmission

The data transmission procedure is the same as that of the slave address transmission.

1. Write an 8-bit transmit data to the transmit data buffer (TD[7:0]).
2. Set IMTGMOD[2:0] to 0x2 and IMTG to 1.

This trigger transfers the buffer data to the transmit shift register to start transmission. The module starts clock output from the USI_CK pin. The data in the shift register is shifted in sequence with the clock and sent from the USI_DO pin.

Writing 1 to IMTG sets IMBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x2 (end of transmit data). An interrupt request can be generated at this point. Write subsequent data to the transmit data buffer to start the following transmission using this interrupt.

However, as in the case of the slave address transmission, check that the slave device has sent back an ACK (by setting IMTGMOD[2:0] to 0x6 and IMTG to 1) before starting the following 8-bit data transmission. Repeat an 8-bit data transmission and ACK receiving check for the required number of times.

(4) Generating stop condition

To end I²C communication after all data has been sent, the I²C master must generate a stop condition. The stop condition applies when the SCL line is maintained at high and the SDA line is pulled up from low to high. To generate a stop condition in this I²C master, set IMTGMOD[2:0] to 0x1 and write 1 to IMTG.

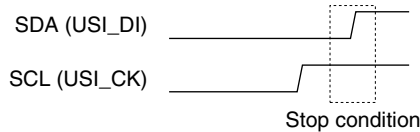


Figure 18.5.3.6 Stop Condition

IMBSY is set to 1 while a stop condition is being generated. When the stop condition is generated, IMBSY is reset to 0 and IMSTA[2:0] is set to 0x1. Read IMBSY or use an interrupt to check that a stop condition has been generated. The I²C bus subsequently switches to free state.

(5) Generating repeated start condition

To make it possible to continue with a different data transfer after a data transmission has completed, the I²C master can omit stop condition generation and generate a repeated start condition. To generate a repeated start condition, perform a start condition generation procedure described in Step (1). Slave address transmission is subsequently possible with the I²C bus remaining in the busy state.

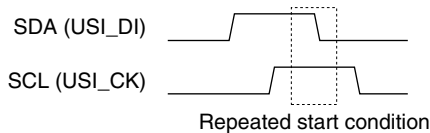


Figure 18.5.3.7 Repeated Start Condition

Data reception in I²C master mode

The following describes the data receiving procedure in I²C master mode.

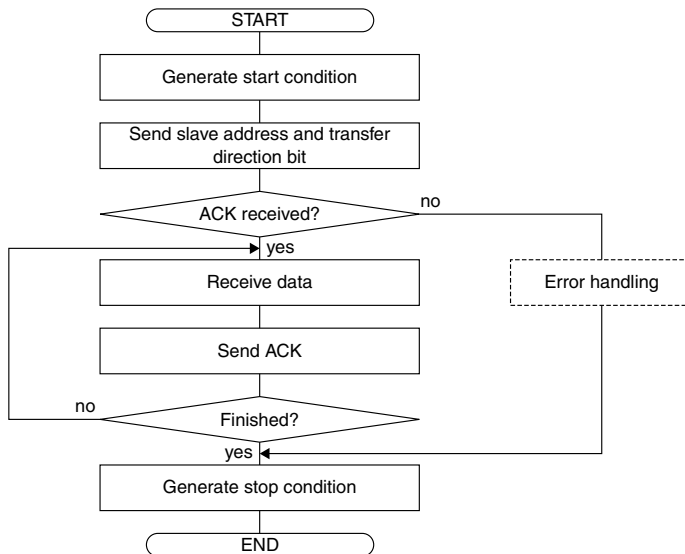
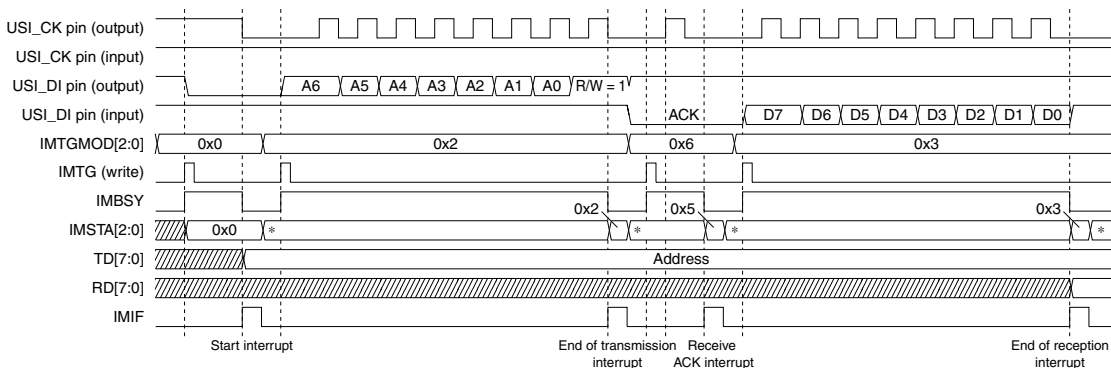
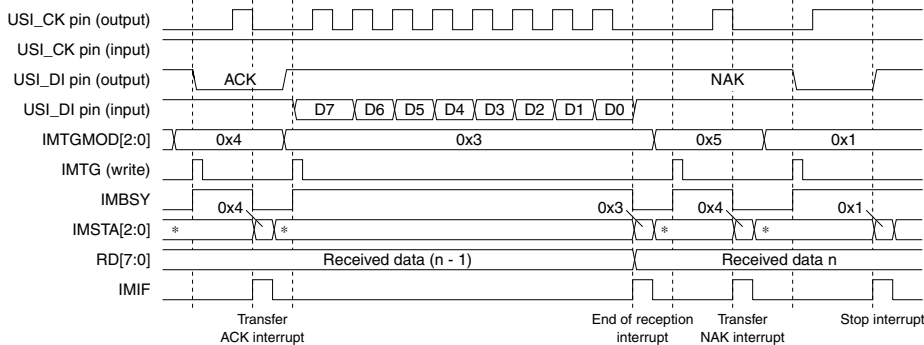


Figure 18.5.3.8 I²C Master Data Receiving Flow Chart

18 UNIVERSAL SERIAL INTERFACE (USI)



(1) Start condition → Data reception



(2) Data reception → Stop condition

* When IMIF is cleared via software, IMSTA[2:0] is also cleared to 0x0.

Figure 18.5.3.9 I²C Master Data Receiving Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See “Receiving control byte in I²C slave mode” in “18.9 Precautions.”

(1) Generating start condition

The procedure is the same as that of data transmission in I²C master mode.

(2) Sending slave address and transfer direction bit

The procedure is the same as that of data transmission in I²C master mode. However, send the slave address with the transfer direction bit set to 1. Then check that the slave device sends back an ACK.

(3) Data reception

To start data reception, set IMTGMOD[2:0] to 0x3 and write 1 to IMTG.

This trigger starts outputting 8 clocks from the USI_CK pin. The USI_DO pin status is sampled in sync with the clock and loaded to the shift register. The received data is loaded to the receive data buffer (RD[7:0]/USI_RD register) once the 8-bit data has been received in the shift register.

Writing 1 to IMTG sets IMBSY to 1. When the received data is loaded to the receive data buffer, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x3 (end of receive data). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

It is necessary to send back an ACK or NAK to the slave device after an 8-bit data has been received. To send back an ACK, set IMTGMOD[2:0] to 0x4 and write 1 to IMTG. To send back a NAK, set IMTGMOD[2:0] to 0x5 and write 1 to IMTG.

IMBSY is set to 1 while an ACK/NAK is being sent and it reverts to 0 when the transmission has completed. An interrupt can be generated at this point. When an ACK or NAK has been sent, IMSTA[2:0] is set to 0x4.

Repeat an 8-bit data reception and ACK (NAK) transmission for the required number of times.

(4) Generating stop condition

The procedure is the same as that of data transmission in I²C master mode.

(5) Generating repeated start condition

The procedure is the same as that of data transmission in I²C master mode.

Clock stretch function

During transmitting/receiving data, the slave device may issue a wait request to the master device by pulling down the SCL line to low until the slave device becomes ready to transmit/receive the subsequent data. The master device enters a standby state until the wait request is canceled (the SCL line goes high).

This I²C controller supports this clock stretch function. When a clock stretch condition is detected after a slave address or data has been sent/received, this module enters a waiting status and it does not start operating even if it accepts a trigger for data transfer until the clock stretch status is canceled. IMBSY is maintained at 1 until the triggered operation has completed including a waiting status.

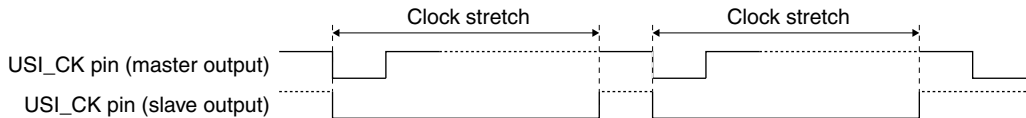


Figure 18.5.3.10 Clock Stretch

Control method in I²C slave mode

Data transfer in I²C slave mode is controlled using ISTGMOD[2:0]/USI_ISTG register and ISTG/USI_ISTG register. Select an I²C slave operation using ISTGMOD[2:0] and write 1 to ISTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

Table 18.5.3.3 Trigger List in I²C Slave Mode

ISTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception/stop condition detection
0x2	Data transmission
0x1	Reserved
0x0	Wait for start condition

(Default: 0x0)

Writing 1 to ISTG sets ISBSY/USI_ISIF register to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, ISBSY is reset to 0. At the same time, the interrupt flag (ISIF/USI_ISIF register) is also set to 1. After an interrupt occurs, read the status bits (ISSTA[2:0]/USI_ISIF register) to check the operation finished. Then, clear ISIF by writing 1. This also automatically reset ISSTA[2:0] to 0x0.

Table 18.5.3.4 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

Data transmission in I²C slave mode

The following describes the data transmission procedure in I²C slave mode.

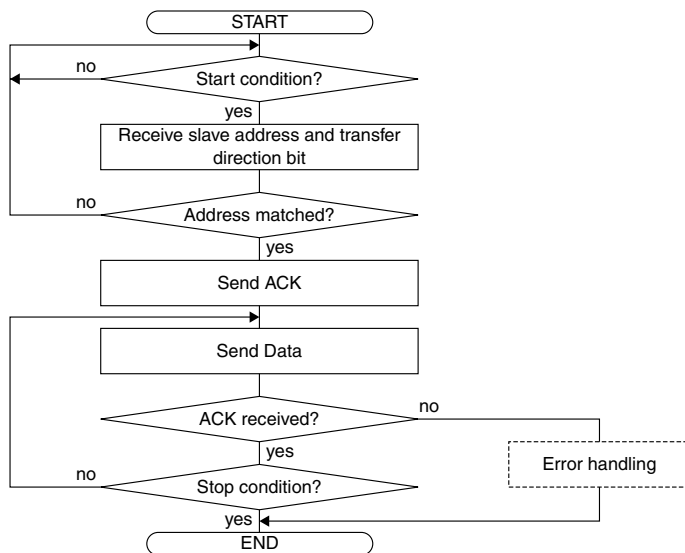
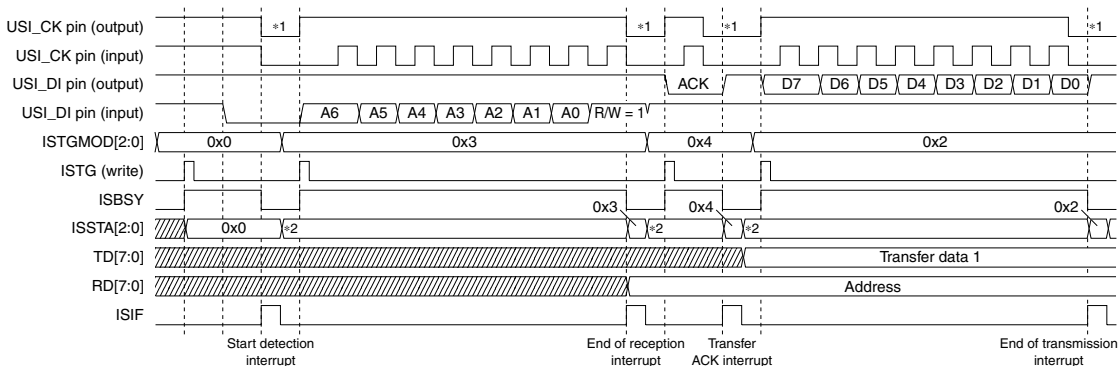
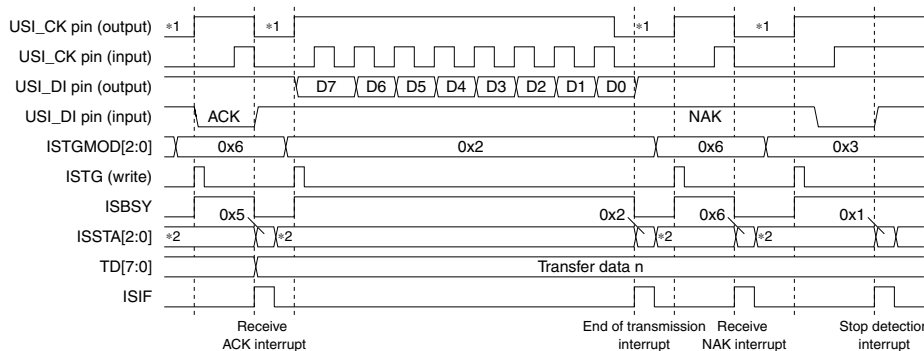


Figure 18.5.3.11 I²C Slave Data Transmission Flow Chart



(1) Start condition → Data transmission



(2) Data transmission → Stop condition

- *1 When the USI_CK input is detected as low after the operation selected by ISTGMOD[2:0] has finished, the USI I²C slave controller pulls down the USI_CK pin to low to places the external I²C master into wait state. This pull-down is canceled to release the I²C master from wait state when the subsequent operation is triggered by ISTG.
- *2 When ISIF is cleared via software, ISSTA[2:0] is also cleared to 0x0.

Figure 18.5.3.12 I²C Slave Data Transmission Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See “Receiving control byte in I²C slave mode” in “18.9 Precautions.”

(1) Waiting for start condition

I²C data transfer starts when the I²C master device generates a start condition (see Figure 18.5.3.3).

First enable this I²C slave to detect a start condition by setting ISTGMOD[2:0] to 0x0 (default) and writing 1 to ISTG. The I²C controller starts detecting a start condition and sets ISBSY to 1. ISBSY is set to 1 while a start condition is being detected. ISBSY reverts to 0 and ISSTA[2:0] is set to 0x0 when the detection has completed. Check if a start condition is generated by reading ISBSY or using an interrupt.

Note: Other operations cannot be started before a start condition is detected.

(2) Receiving slave address and transfer direction data bit

The I²C master sends the address of the slave device to be communicated and a transfer direction bit (see Figure 18.5.3.4) after it has generated a start condition. Set this I²C slave into receiving status to receive the slave address. To start reception, set ISTGMOD[2:0] to 0x3 and write 1 to ISTG.

This trigger starts sampling clocks input from the USI_{CK} pin. When clocks are input, the I²C controller loads the USI_{DO} pin status to the shift register in sync with each clock. The received data is loaded to the receive data buffer (RD[7:0]/USI_{RD} register) once the 8-bit data has been received in the shift register.

Writing 1 to ISTG sets ISBSY to 1. When the received data is loaded to the receive data buffer, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x3 (end of receive data). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

When a 7-bit address is used, the slave address and transfer direction bit can be obtained in one operation. When a 10-bit address is used, save the first data received in the receive data buffer into the memory and perform data reception again to obtain the remaining address bits.

Check whether the received address is matched to this I²C slave address or not. When they are matched, send back an ACK to the I²C master by setting ISTGMOD[2:0] to 0x4 and write 1 to ISTG. ISBSY is set to 1 while an ACK is being sent and it reverts to 0 when the transmission has completed. An interrupt request can be generated at this point. When an ACK has been sent, ISSTA[2:0] is set to 0x4.

If the received address is not for this I²C slave, abort data reception and return to Step (1) to wait the subsequent start condition.

(3) Data transmission

When the transfer direction bit received with the slave address in Step (2) is 1, start data transmission by the following procedure:

1. Write an 8-bit transmit data to the transmit data buffer (TD[7:0]).
2. Set ISTGMOD[2:0] to 0x2 and ISTG to 1.

This trigger transfers the buffer data to the transmit shift register to start transmission. When clocks are input from the USI_{CK} pin, the data in the shift register is shifted in sequence with the clock and sent from the USI_{DO} pin.

Writing 1 to ISTG sets ISBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x2 (end of transmit data). An interrupt request can be generated at this point. Write subsequent data to the transmit data buffer to start the following transmission using this interrupt.

However, check that the master device has sent back an ACK or NAK (by setting ISTGMOD[2:0] to 0x6 and ISTG to 1) before starting the following 8-bit data transmission.

ISBSY is set to 1 while an ACK/NAK is being detected and it reverts to 0 when the detection has completed. Receiving an ACK sets ISSTA[2:0] to 0x5; receiving a NAK sets it to 0x6. Check ISSTA[2:0] after confirming ISBSY or using an interrupt. When an ACK has been received, perform data transmission. When a NAK has been received, perform the appropriate handling.

(4) When a stop condition is received

If the ISSTA[2:0] value read during data transmission is 0x1, the I²C master device has generated a stop condition (see Figure 18.5.3.6). In this case, abort data transmission.

Data reception in I²C slave mode

The following describes the data receiving procedure in I²C slave mode.

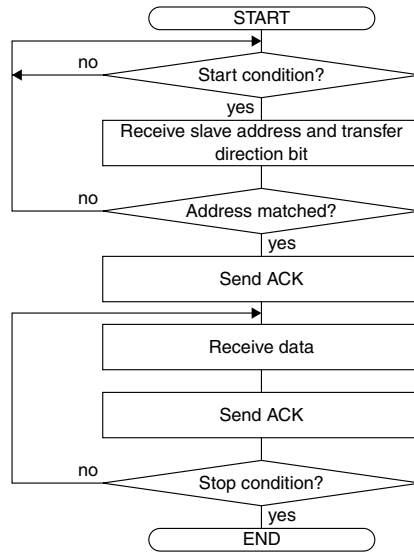
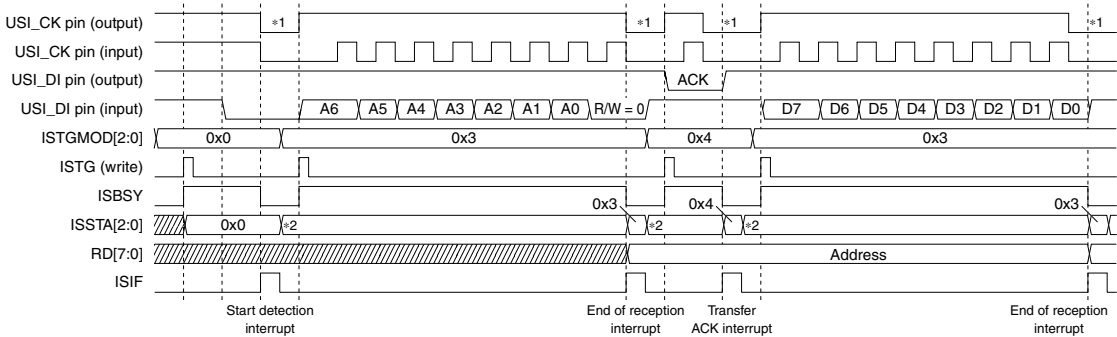
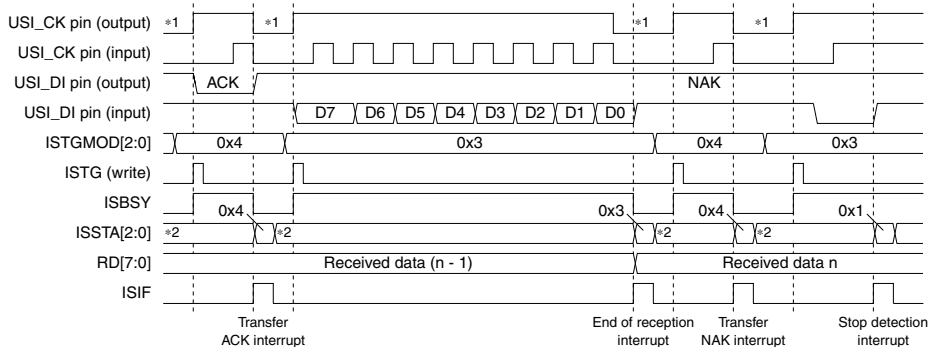


Figure 18.5.3.13 I²C Slave Data Receiving Flow Chart



(1) Start condition → Data reception



(2) Data reception → Stop condition

- *1 When the USI_CK input is detected as low after the operation selected by ISTGMOD[2:0] has finished, the USI I²C slave controller pulls down the USI_CK pin to low to places the external I²C master into wait state. This pull-down is canceled to release the I²C master from wait state when the subsequent operation is triggered by ISTG.
- *2 When ISIF is cleared via software, ISSTA[2:0] is also cleared to 0x0.

Figure 18.5.3.14 I²C Slave Data Receiving Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See “Receiving control byte in I²C slave mode” in “18.9 Precautions.”

(1) Waiting for start condition

The procedure is the same as that of data transmission in I²C slave mode.

(2) Receiving slave address and transfer direction data bit

The procedure is the same as that of data transmission in I²C slave mode.

(3) Data reception

When the transfer direction bit received with the slave address in Step (2) is 0, start data reception by setting ISTGMOD[2:0] to 0x3 and writing 1 to ISTG.

When clocks are input, the I²C controller loads the USI_DO pin status to the shift register in sync with each clock. The received data is loaded to the receive data buffer (RD[7:0]/USI_RD register) once the 8-bit data has been received in the shift register.

Writing 1 to ISTG sets ISBSY to 1. When the received data is loaded to the receive data buffer, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x3 (end of receive data). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

It is necessary to send back an ACK or NAK to the master device after an 8-bit data has been received. To send back an ACK, set ISTGMOD[2:0] to 0x4 and write 1 to ISTG. To send back a NAK, set ISTGMOD[2:0] to 0x5 and write 1 to ISTG.

ISBSY is set to 1 while an ACK/NAK is being sent and it reverts to 0 when the transmission has completed. An interrupt request can be generated at this point. When an ACK or NAK has been sent, ISSTA[2:0] is set to 0x4.

Repeat an 8-bit data reception and ACK (NAK) transmission for the required number of times.

(4) When a stop condition is received

If the ISSTA[2:0] value read during data reception is 0x1, the I²C master device has generated a stop condition (see Figure 18.5.3.6). In this case, abort data reception.

Clock stretch function

While data is being sent/received, this I²C slave generates a clock stretch status by pulling down the SCL line to low to make a wait request to the master device after an ACK is sent/received until the following data transfer is started.

18.6 Receive Errors

In UART mode, three different receive errors (overrun error, framing error, and parity error) may be detected while receiving data. In SPI and I²C modes, overrun errors may be detected while receiving data.

Since receive errors are interrupt causes, they can be processed by generating interrupts. For more information on interrupt control, see Section 18.7.

Overrun error (all interface modes)

UART mode

An overrun error occurs if the next reception is completed when URDIF is 1 and the receive data buffer (USI_RD register) is not read (an overrun error occurs at the time stop bit has been received).

When an overrun error occurs, the overrun error flag (UOEIF/USI_UIF register) is set to 1. The receiving operation continues even if this error occurs. To reset UOEIF, perform USI software reset (write 0x0 to USIMOD[2:0]/USI_GCFG register) to initialize USI.

SPI mode

An overrun error occurs if data are received successively when SRDIF is 1. While SRDIF is set to 1, the next received data will not be transferred from the shift register to the receive data buffer (the first byte data exists in the receive data buffer and the second byte data exists in the shift register). An overrun error occurs if the third byte data is received in this condition, as the second byte data in the shift register is corrupted (an overrun error occurs at the time the first bit of the third byte is fetched).

When an overrun error occurs, the overrun error flag (SEIF/USI_SIF register) is set to 1. The receiving operation continues even if this error occurs. SEIF is reset by writing 1. To reset an overrun error, write 1 to SEIF and then read the receive data buffer (USI_RD register) twice. The procedure that writes 1 to SEIF and reads USI_RD register twice can be reversed.

I²C master/slave mode

An overrun error occurs when a transmit or receive trigger is issued after two-byte data has been received (the first byte data exists in the receive data buffer and the second byte data exists in the shift register) without the receive data buffer being read.

When an overrun error occurs, the overrun error flag (IMEIF/USI_IMIF register for I²C master mode or ISEIF/USI_ISIF register for I²C slave mode) is set to 1. The receiving operation continues even if this error occurs. IMEIF/ISEIF is reset by writing 1. To reset an overrun error, write 1 to IMEIF/ISEIF and then read the receive data buffer (USI_RD register) twice.

Framing error (UART mode only)

If the stop bit is received as 0 in UART mode, the UART controller determines loss of sync and a framing error occurs. If the stop bit is configured to two bits, only the first bit is checked.

The framing error flag (USEIF/USI_UIF register) is set to 1 if this error occurs. The received data is still transferred to the receive data buffer if this error occurs and the receiving operation continues, but the data cannot be guaranteed, even if no framing error occurs for subsequent data receiving. The framing error flag is reset to 0 by writing 1.

Parity error (UART mode only)

If UPREN/USI_UCFG register has been set to 1 (parity enabled), data received is checked for parity in UART mode. Data received in the shift register is checked for parity when sent to the receive data buffer. The matching is checked against the UPMD/USI_UCFG register setting (odd or even parity). If the result is a non-match, a parity error is issued, and the parity error flag (UPEIF/USI_UIF register) is set to 1. Even if this error occurs, the data received is sent to the receive data buffer, and the receiving operation continues. However, the received data cannot be guaranteed if a parity error occurs. The UPEIF flag is reset to 0 by writing 1.

18.7 USI Interrupts and DMA

This section describes the USI interrupts generated in each interface mode and invoking DMA.

For more information on interrupt processing and DMA transfer, see the “Interrupt Controller (ITC)” chapter and the “DMA Controller (DMAC)” chapter, respectively.

The USI outputs one interrupt signal shared by the all interrupt causes to the interrupt controller (ITC). Inspect the interrupt flags available in each mode to determine the interrupt cause occurred.

18.7.1 Interrupts in UART Mode

The UART mode includes a function for generating the following three different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

Transmit buffer empty interrupt

To use this interrupt, set UTDIE/USI_UIE register to 1. If UTDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When transmit data written to the transmit data buffer is transferred to the shift register, the USI module sets UTDIF/USI_UIF register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (UTDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the UTDIF flag in the interrupt handler routine to determine whether the USI (UART mode) interrupt is attributable to a transmit buffer empty. If UTDIF is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set URDIE/USI_UIE register to 1. If URDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If a received data is loaded into the receive data buffer, the USI module sets URDIF/USI_UIF register to 1. If receive buffer full interrupts are enabled (URDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the URDIF flag in the interrupt handler routine to determine whether the USI (UART mode) interrupt is attributable to a receive buffer full. If URDIF is 1, the received data can be read from the receive data buffer by the interrupt handler routine. However, be sure to check whether a receive error has occurred or not.

Receive error interrupt

To use this interrupt, set UEIE/USI_UIE register to 1. If UEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

The USI module sets an error flag (UPEIF/USI_UIF register, USEIF/USI_UIF register, or UOEIF/USI_UIF register) to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are enabled (UEIE = 1), an interrupt request is sent simultaneously to the ITC.

If other interrupt conditions are satisfied, an interrupt occurs. You can inspect the UPEIF, USEIF, and UOEIF flags in the interrupt handler routine to determine whether the USI (UART mode) interrupt was caused by a receive error. If any of the error flags has the value 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, perform USI software reset (write 0x0 to USIMOD[2:0]/USI_GCFG register) to initialize USI.

18.7.2 Interrupts in SPI Mode

The SPI master/slave modes include a function for generating the following three different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

Transmit buffer empty interrupt

To use this interrupt, set STDIE/USI_SIE register to 1. If STDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When transmit data written to the transmit data buffer is transferred to the shift register, the USI module sets STDIF/USI_SIF register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (STDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the STDIF flag in the interrupt handler routine to determine whether the USI (SPI master/slave mode) interrupt is attributable to a transmit buffer empty. If STDIF is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set SRDIE/USI_SIE register to 1. If SRDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If a received data is loaded into the receive data buffer, the USI module sets SRDIF/USI_SIF register to 1. If receive buffer full interrupts are enabled (SRDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the SRDIF flag in the interrupt handler routine to determine whether the USI (SPI master/slave mode) interrupt is attributable to a receive buffer full. If SRDIF is 1, the received data can be read from the receive data buffer by the interrupt handler routine. However, be sure to check whether a receive error has occurred or not.

Receive error interrupt

To use this interrupt, set SEIE/USI_SIE register to 1. If SEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

18 UNIVERSAL SERIAL INTERFACE (USI)

The USI module sets SEIF/USI_SIF register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (SEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the SEIF flags in the interrupt handler routine to determine whether the USI (SPI master/slave mode) interrupt was caused by a receive error. If SEIF is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear SEIF by writing 1 and then read the receive data buffer (USI_RD register) twice.

18.7.3 Interrupts in I²C Master Mode

The I²C master mode includes a function for generating the following two different types of interrupts.

- Operation completion interrupt
- Receive error interrupt

Operation completion interrupt

To use this interrupt, set IMIE/USI_IMIE register to 1. If IMIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the operation that initiated by a software trigger has completed, the USI module sets IMIF/USI_IMIF register to 1. If operation completion interrupts are enabled (IMIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the IMSTA[2:0]/USI_IMIF register in the interrupt handler routine to determine the I²C operation/status that causes the interrupt.

Table 18.7.3.1 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

Receive error interrupt

To use this interrupt, set IMEIE/USI_IMIE register to 1. If IMEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

An overrun error occurs at the time a transmit or receive trigger is issued after two-byte data has been received without reading the receive data buffer.

The USI module sets IMEIF/USI_IMIF register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (IMEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the IMEIF flags in the interrupt handler routine to determine whether the USI (I²C master mode) interrupt was caused by a receive error. If IMEIF is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear IMEIF by writing 1, and then read the receive data buffer (USI_RD register) twice.

18.7.4 Interrupts in I²C Slave Mode

The I²C slave mode includes a function for generating the following two different types of interrupts.

- Operation completion interrupt
- Receive error interrupt

Operation completion interrupt

To use this interrupt, set ISIE/USI_ISIE register to 1. If ISIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the operation that initiated by a software trigger has completed, the USI module sets ISIF/USI_ISIF register to 1. If operation completion interrupts are enabled (ISIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the ISSTA[2:0]/USI_ISIF register in the interrupt handler routine to determine the I²C operation/status that causes the interrupt.

Table 18.7.4.1 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

Receive error interrupt

To use this interrupt, set ISEIE/USI_ISIE register to 1. If ISEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

An overrun error occurs at the time a transmit or receive trigger is issued after two-byte data has been received without reading the receive data buffer.

The USI module sets ISEIF/USI_ISIF register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (ISEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the ISEIF flags in the interrupt handler routine to determine whether the USI (I²C slave mode) interrupt was caused by a receive error. If ISEIF is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear ISEIF by writing 1, and then read the receive data buffer (USI_RD register) twice.

18.7.5 DMA Transfer

The causes of receive buffer full and transmit buffer empty interrupts in UART and SPI master/slave modes can invoke a DMA. This allows continuous data transmission/reception through DMA transfer between memory and transmit/receive data buffers. These interrupt signals are output to both the ITC and DMAC. Therefore, DMA transfer can be performed without generating any USI interrupt.

The following lists the DMAC channels that allow selection of a USI interrupt cause as the trigger.

USI receive buffer full: DMAC Ch.2

USI transmit buffer empty: DMAC Ch.3

For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

Note: The USI module cannot invoke a DMA in I²C master and slave mode.

18.8 Control Register Details

Table 18.8.1 List of USI Registers

Address	Register name		Function
0x300400	USI_GCFG	USI Global Configuration Register	Sets interface and MSB/LSB mode.
0x300401	USI_TD	USI Transmit Data Buffer Register	Transmit data buffer
0x300402	USI_RD	USI Receive Data Buffer Register	Receive data buffer
0x300440	USI_UCFG	USI UART Mode Configuration Register	Sets UART transfer conditions.
0x300441	USI_UIE	USI UART Mode Interrupt Enable Register	Enables interrupts.
0x300442	USI_UIF	USI UART Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x300450	USI_SCFG	USI SPI Master/Slave Mode Configuration Register	Sets SPI transfer conditions.
0x300451	USI_SIE	USI SPI Master/Slave Mode Interrupt Enable Register	Enables interrupts.
0x300452	USI_SIF	USI SPI Master/Slave Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x300460	USI_IMTG	USI I ² C Master Mode Trigger Register	Starts I ² C master operations.
0x300461	USI_IMIE	USI I ² C Master Mode Interrupt Enable Register	Enables interrupts.
0x300462	USI_IMIF	USI I ² C Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.

18 UNIVERSAL SERIAL INTERFACE (USI)

Address	Register name		Function
0x300470	USI_ISTG	USI I ² C Slave Mode Trigger Register	Starts I ² C slave operations.
0x300471	USI_ISIE	USI I ² C Slave Mode Interrupt Enable Register	Enables interrupts.
0x300472	USI_ISIF	USI I ² C Slave Mode Interrupt Flag Register	Indicates interrupt occurrence status.

The USI registers are described in detail below. These are 8-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

USI Global Configuration Register (USI_GCFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USI Global Configuration Register (USI_GCFG)	0x300400 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3	LSBFST	MSB/LSB first mode select	1 MSB first 0 LSB first	0	R/W		
		D2–0	USIMOD[2:0]	Interface mode configuration	USIMOD[2:0]	I/F mode	0x0	R/W	
					0x7–0x6	reserved			
					0x5	I ² C slave			
					0x4	I ² C master			
			0x3	SPI slave					
			0x2	SPI master					
			0x1	UART					
			0x0	Software reset					

Note: This register must be configured before setting other USI registers.

D[7:4] Reserved

D3 LSBFST: MSB/LSB First Mode Select Bit

Selects whether serial data will be transferred from the MSB or LSB.

1 (R/W): MSB first

0 (R/W): LSB first (default)

This setting affects all interface modes.

D[2:0] USIMOD[2:0]: Interface Mode Configuration Bits

Selects an interface mode.

Table 18.8.2 Interface Mode Selection

USIMOD[2:0]	Interface mode
0x5	I ² C slave
0x4	I ² C master
0x3	SPI slave
0x2	SPI master
0x1	UART
0x0	Software reset

(Default: 0x0)

Perform software reset (set USIMOD[2:0] to 0x0) and then set the interface mode before changing other USI configurations.

USI Transmit Data Buffer Register (USI_TD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Transmit Data Buffer Register (USI_TD)	0x300401 (8 bits)	D7–0	TD[7:0]	USI transmit data buffer TD7 = MSB TD0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] TD[7:0]: USI Transmit Data Buffer Bits

Sets transmit data to be written to the transmit data buffer. (Default: 0x0)

In UART and SPI master modes, transmission begins immediately after writing data to this register. In SPI slave mode, transmission will begin when the clock is input from the SPI master device.

In I²C master/slave mode, transmission begins by the software trigger for data transmission.

The data written to this register is converted into serial data through the shift register and is output from the USI_DO pin with the bit set to 1 as high level and the bit set to 0 as low level.

A transmit buffer empty interrupt can be generated when data written to this register has been transferred to the shift register. The subsequent transmit data can then be written, even while data is being sent.

USI Receive Data Buffer Register (USI_RD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Receive Data Buffer Register (USI_RD)	0x300402 (8 bits)	D7-0	RD[7:0]	USI receive data buffer RD7 = MSB RD0 = LSB	0x0 to 0xff	0x0	R	

D[7:0] RD[7:0]: USI Receive Data Buffer Bits

Contains the received data. (Default: 0x0)

Serial data input from the USI_DI pin is converted to parallel, with the high level bit set to 1 and the low level bit set to 0, and then it is loaded to this register.

A receive buffer full interrupt can be generated when the data received in the shift register has been loaded to this register. Data can then be read until subsequent data is received.

This register is read-only.

USI UART Mode Configuration Register (USI_UCFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USI UART Mode Configuration Register (USI_UCFG)	0x300440 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.	
		D3	UCHLN	Character length select	1 8 bits	0 7 bits	0	R/W	
		D2	USTPB	Stop bit select	1 2 bits	0 1 bit	0	R/W	
		D1	UPMD	Parity mode select	1 Even	0 Odd	0	R/W	
		D0	UPREN	Parity enable	1 With parity	0 No parity	0	R/W	

Note: This register is effective only in UART mode. Configure USI to UART mode before setting this register.

D[7:4] Reserved

D3 UCHLN: Character Length Select Bit

Selects the serial transfer data length.

1 (R/W): 8 bits

0 (R/W): 7 bits (default)

When 7-bit data length is selected, D7 in the transmit data buffer is ignored and D7 in the receive data buffer is always set to 0.

D2 USTPB: Stop Bit Select Bit

Selects the stop bit length.

1 (R/W): 2 bits

0 (R/W): 1 bit (default)

Writing 1 to USTPB selects 2 stop bits; writing 0 to it selects 1 bit. The start bit is fixed at 1 bit.

D1 UPMD: Parity Mode Select Bit

Selects the parity mode.

1 (R/W): Even parity

0 (R/W): Odd parity (default)

Parity checking and parity bit addition are enabled only when UPREN is set to 1. The UPMD setting is disabled if UPREN is 0.

D0 UPREN: Parity Enable Bit

Enables the parity function.

1 (R/W): With parity

0 (R/W): No parity (default)

UPREN is used to select whether received data parity checking is performed and whether a parity bit is added to transmit data. Setting UPREN to 1 parity-checks the received data. A parity bit is automatically added to the transmit data. If UPREN is set to 0, no parity bit is checked or added.

USI UART Mode Interrupt Enable Register (USI_UIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI UART Mode Interrupt Enable Register (USI_UIE)	0x300441 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	UEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	URDIE	Receive buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	UTDIE	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in UART mode. Configure USI to UART mode before this register can be used.

D[7:3] Reserved

D2 UEIE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when a receive error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process receive errors using interrupts.

D1 URDIE: Receive Buffer Full Interrupt Enable Bit

Enables interrupt requests to the ITC when received data is loaded to the receive data buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

D0 UTDIE: Transmit Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC when data written to the transmit data buffer is sent to the shift register (i.e. when data transmission begins).

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to write data to the transmit data buffer using interrupts.

USI UART Mode Interrupt Flag Register (USI_UIF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI UART Mode Interrupt Flag Register (USI_UIF)	0x300442 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	URBSY	Receive busy flag	1 Busy 0 Idle	0	R	
		D5	UTBSY	Transmit busy flag	1 Busy 0 Idle	0	R	
		D4	UPEIF	Parity error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D3	USEIF	Framing error flag	1 Error 0 Normal	0	R/W	
		D2	UOEIF	Overrun error flag	1 Error 0 Normal	0	R/W	
		D1	URDIF	Receive buffer full flag	1 Full 0 Not full	0	R/W	
		D0	UTDIF	Transmit buffer empty flag	1 Empty 0 Not empty	0	R/W	

Note: This register is effective only in UART mode. Configure USI to UART mode before this register can be used.

D7 Reserved

D6 URBSY: Receive Busy Flag Bit

Indicates the receive shift register status.

1 (R): Busy

0 (R): Idle (default)

URBSY is set to 1 when the first start bit is detected (when data reception begins) and is reset to 0 when the data received in the shift register is loaded into the receive data buffer. Inspect URBSY to determine whether the receiving circuit is operating or at standby.

D5 UTBSY: Transmit Busy Flag Bit

Indicates the USI status in UART mode.

1 (R): Busy

0 (R): Idle (default)

UTBSY switches to 1 when transmit data is written to the transmit buffer and reverts to 0 after both the shift register and transmit buffer become empty.

D4 UPEIF: Parity Error Flag Bit

Indicates whether a parity error has occurred or not.

- 1 (R): Error occurred
- 0 (R): No error (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

UPEIF is set to 1 when a parity error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USI_UIE register is 1. Parity checking is enabled only when UPREN/USI_UCFG register is set to 1 and is performed when received data is transferred from the shift register to the receive data buffer. UPEIF is reset by writing 1.

D3 USEIF: Framing Error Flag Bit

Indicates whether a framing error has occurred or not.

- 1 (R): Error occurred
- 0 (R): No error (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

USEIF is set to 1 when a framing error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USI_UIE register is 1. A framing error occurs when data is received with the stop bit set to 0. USEIF is reset by writing 1.

D2 UOEIF: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

- 1 (R): Error occurred
- 0 (R): No error (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

UOEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USI_UIE register is 1. An overrun error occurs if the next reception is completed when URDIF is 1 and the receive data buffer (USI_RD register) is not read (an overrun error occurs at the time stop bit has been received). To reset UOEIF, perform USI software reset (write 0x0 to USIMOD[2:0]/USI_GCFG register) to initialize USI.

D1 URDIF: Receive Buffer Full Flag Bit

Indicates the receive data buffer status.

- 1 (R): Data full
- 0 (R): No data (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

URDIF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is completed), indicating that the data can be read. At the same time a receive buffer full interrupt request is sent to the ITC if URDIE/USI_UIE register is 1. URDIF is reset by writing 1.

D0 UTDIF: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

- 1 (R): Empty
- 0 (R): Data exists (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

UTDIF is set to 1 when the transmit data written to the transmit data buffer is transferred to the shift register (when transmission starts), indicating that the next transmit data can be written to. At the same time a transmit buffer empty interrupt request is sent to the ITC if UTDIE/USI_UIE register is 1. UTDIF is reset by writing 1.

USI SPI Master/Slave Mode Configuration Register (USI_SCFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
USI SPI Master/ Slave Mode Configuration Register (USI_SCFG)	0x300450 (8 bits)	D7-6	-	reserved			-	-	0 when being read.	
		D5	SCMD	Command bit (for 9-bit data)	1 High	0 Low	0	R/W		
		D4	SCHLN	Character length select	1 9 bits	0 8 bits	0	R/W		
		D3	SCPHA	Clock phase select	1 Phase 1	0 Phase 0	0	R/W		
		D2	SCPOL	Clock polarity select	1 Active L	0 Active H	0	R/W		
		D1	-	reserved				-	-	Do not set to 1.
		D0	SFSTMOD	Fast mode select	1 Fast	0 Normal	0	R/W		

Note: This register is effective only in SPI master and slave modes. Configure USI to SPI master/slave mode before this register can be used.

D[7:6] Reserved

D5 SCMD: Command Bit (for 9-bit data in SPI master mode)

Sets the command bit value for 9-bit data (see SCHLN below).

1 (R/W): High

0 (R/W): Low (default)

D4 SCHLN: Character Length Select Bit (for SPI master mode)

Selects the serial transfer data length.

1 (R/W): 9 bits

0 (R/W): 8 bits (default)

In 9-bit mode, 8-bit data is prefixed with a command bit (1 bit). The command bit is used for controlling the SPI LCD controller connected to the USI. The command bit value to be transmitted can be specified using SCMD.

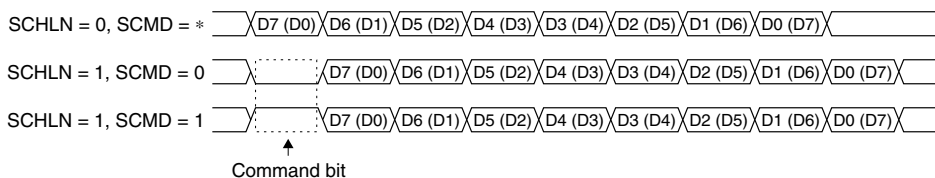


Figure 18.8.1 9-bit Transfer Data Format in SPI Master Mode

This bit is effective only in SPI master mode. The data length in SPI slave mode is fixed at 8 bits.

D3 SCPHA: Clock Phase Select Bit

Selects the SPI clock phase.

1 (R/W): Phase 1

0 (R/W): Phase 0 (default)

Set the data transfer timing together with SCPOL. (See Figure 18.8.2.)

D2 SCPOL: Clock Polarity Select Bit

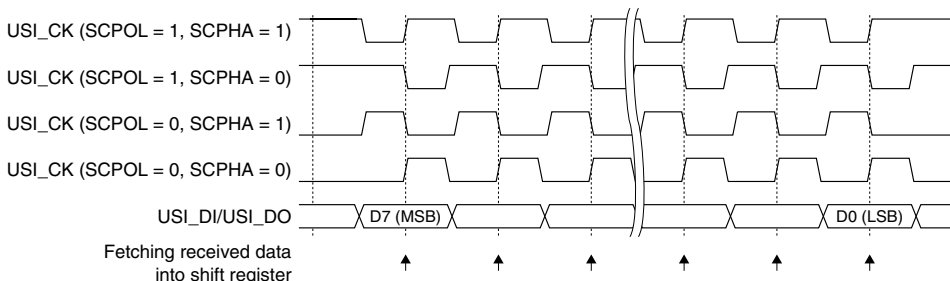
Selects the SPI clock polarity.

1 (R/W): Active low

0 (R/W): Active high (default)

Set the data transfer timing together with SCPHA. (See Figure 18.8.2.)

Master mode



Slave mode

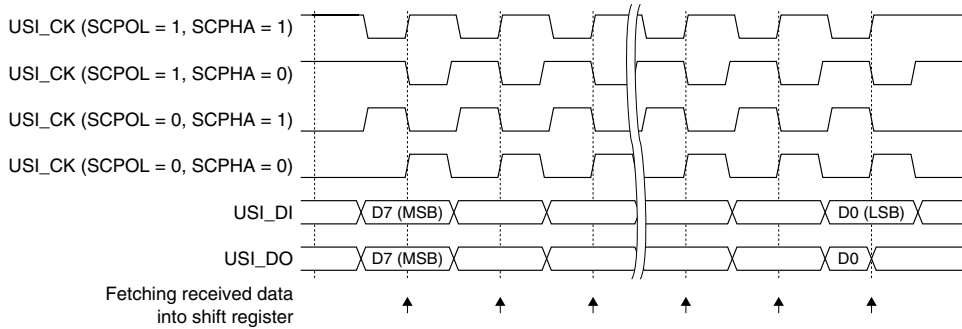


Figure 18.8.2 Clock and Data Transfer Timing (MSB first)

D1 **Reserved (Do not set to 1.)**

D0 **SFSTMOD: Fast Mode Select Bit (for SPI master mode)**

Selects Fast mode.

1 (R/W): Fast mode

0 (R/W): Normal mode (default)

In SPI master mode, either normal or fast clock mode can be selected using SFSTMOD. Setting SFSTMOD to 0 (default) places the USI into normal mode and the USI generates the transfer clock by dividing the T8 output by 2. Setting SFSTMOD to 1 places the USI into fast mode and the USI uses PCLK1 supplied from the CMU directly as the transfer clock. The fast mode does not use the T8.

The SPI slave mode uses the T8 output clock for generating the sampling clock.

USI SPI Master/Slave Mode Interrupt Enable Register (USI_SIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI SPI Master/Slave Mode Interrupt Enable Register (USI_SIE)	0x300451 (8 bits)	D7-3	—	reserved		—	—	0 when being read.
		D2	SEIE	Receive error interrupt enable	1 Enable	0 Disable	0	R/W
		D1	SRDIE	Receive buffer full interrupt enable	1 Enable	0 Disable	0	R/W
		D0	STDIE	Transmit buffer empty int. enable	1 Enable	0 Disable	0	R/W

Note: This register is effective only in SPI master and slave modes. Configure USI to SPI master/slave mode before this register can be used.

D[7:3] **Reserved**

D2 **SEIE: Receive Error Interrupt Enable Bit**

Enables interrupt requests to the ITC when an overrun error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D1 **SRDIE: Receive Buffer Full Interrupt Enable Bit**

Enables interrupt requests to the ITC when received data is loaded to the receive data buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

D0 **STDIE: Transmit Buffer Empty Interrupt Enable Bit**

Enables interrupt requests to the ITC when data written to the transmit data buffer is sent to the shift register (i.e. when data transmission begins).

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to write data to the transmit data buffer using interrupts.

USI SPI Master/Slave Mode Interrupt Flag Register (USI_SIF)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
USI SPI Master/ Slave Mode Interrupt Flag Register (USI_SIF)	0x300452 (8 bits)	D7-4	–	reserved				–	–	0 when being read.	
		D3	SSIF	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = H	0	ss = L			
		D2	SEIF	Overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 1.
		D1	SRDIF	Receive buffer full flag	1	Full	0	Not full	0	R/W	
D0	STDIF	Transmit buffer empty flag	1	Empty	0	Not empty	0	R/W			

Note: This register is effective only in SPI master and slave modes. Configure USI to SPI master/slave mode before this register can be used.

D[7:4] Reserved

D3 SSIF: Transfer Busy Flag Bit (Master Mode)/ss Signal Low Flag Bit (Slave Mode)

Master mode

Indicates the SPI transfer status.

1 (R): Operating

0 (R): Standby (default)

SSIF is set to 1 when the SPI starts data transfer in master mode and is maintained at 1 while transfer is underway. It is cleared to 0 once the transfer is completed.

Slave mode

Indicates the slave select (USI_CS) signal status.

1 (R): High level (this SPI is not selected)

0 (R): Low level (this SPI is selected) (default)

SSIF is set to 0 when the master device asserts the slave select (USI_CS) signal to select this SPI controller (slave device). It is returned to 1 when the master device clears the SPI controller selection by negating the slave select (USI_CS) signal.

D2 SEIF: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): Error occurred

0 (R): No error (default)

1 (W): Reset to 0

0 (W): Ignored

SEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if SEIE/USI_SIE register is 1. An overrun error occurs if data are received successively when SRDIF is 1. While SRDIF is set to 1, the next received data will not be transferred from the shift register to the receive data buffer (the first byte data exists in the receive data buffer and the second byte data exists in the shift register). An overrun error occurs if the third byte data is received in this condition, as the second byte data in the shift register is corrupted (an overrun error occurs at the time the first bit of the third byte is fetched).

SEIF is reset by writing 1. To reset an overrun error, write 1 to SEIF and then read the receive data buffer (USI_RD register) twice. The procedure that writes 1 to SEIF and reads USI_RD register twice can be reversed.

D1 SRDIF: Receive Buffer Full Flag Bit

Indicates the receive data buffer status.

1 (R): Data full

0 (R): No data (default)

1 (W): Reset to 0

0 (W): Ignored

SRDIF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is completed), indicating that the data can be read. At the same time a receive buffer full interrupt request is sent to the ITC if SRDIE/USI_SIE register is 1. SRDIF is reset by writing 1.

D0 STDIF: Transmit Buffer Empty Flag Bit

Indicates the transmit data buffer status.

- 1 (R): Empty
- 0 (R): Data exists (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

STDIF is set to 1 when the transmit data written to the transmit data buffer is transferred to the shift register (when transmission starts), indicating that the next transmit data can be written to. At the same time a transmit buffer empty interrupt request is sent to the ITC if STDIE/USI_SIE register is 1. STDIF is reset by writing 1.

USI I²C Master Mode Trigger Register (USI_IMTG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USI I ² C Master Mode Trigger Register (USI_IMTG)	0x300460 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4	IMTG	I ² C master operation trigger	1 Trigger 0 Ignored	0	W		
						1 Waiting 0 Finished		R	
		D3	–	reserved	–	–	–	–	0 when being read.
		D2-0	IMTGMOD [2:0]	I ² C master trigger mode select	IMTGMOD[2:0]	Trigger mode	0x0	R/W	
				0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 Stop condition 0x0 Start condition					

Note: This register is effective only in I²C master mode. Configure USI to I²C master mode before this register can be used.

D[7:5] Reserved**D4 IMTG: I²C Master Operation Trigger Bit**

Starts an I²C master operation.

- 1 (W): Trigger
- 0 (W): Ignored
- 1 (R): Waiting for starting operation
- 0 (R): Trigger has finished (default)

Select an I²C master operation using IMTGMOD[2:0] and write 1 to IMTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

D3 Reserved**D[2:0] IMTGMOD[2:0]: I²C Master Trigger Mode Select Bits**

Selects an I²C master operation.

Table 18.8.3 Trigger List in I²C Master Mode

IMTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception
0x2	Data transmission
0x1	Stop condition
0x0	Start condition

(Default: 0x0)

USI I²C Master Mode Interrupt Enable Register (USI_IMIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI I ² C Master Mode Interrupt Enable Register (USI_IMIE)	0x300461 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	IMEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	IMIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in I²C master mode. Configure USI to I²C master mode before this register can be used.

D[7:2] Reserved

D1 IMEIE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when an overrun error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D0 IMIE: Operation Completion Interrupt Enable Bit

Enables interrupt requests to the ITC when the triggered operation has completed.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to confirm whether the triggered operation has completed or not using interrupts.

USI I²C Master Mode Interrupt Flag Register (USI_IMIF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI I ² C Master Mode Interrupt Flag Register (USI_IMIF)	0x300462 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	IMBSY	I ² C master busy flag	1 Busy 0 Standby	0	R	
		D4–2	IMSTA[2:0]	I ² C master status	IMSTA[2:0] 0x7 reserved 0x6 NAK received 0x5 ACK received 0x4 ACK/NAK sent 0x3 End of Rx data 0x2 End of Tx data 0x1 Stop generated 0x0 Start generated	0x0	R	
		D1	IMEIF	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D0	IMIF	Operation completion flag	1 Completed 0 Not completed	0	R/W	

Note: This register is effective only in I²C master mode. Configure USI to I²C master mode before this register can be used.

D[7:6] Reserved

D5 IMBSY: I²C Master Busy Flag Bit

Indicates the I²C master operation status.

1 (R): Busy

0 (R): Standby (default)

Writing 1 to IMTG/USI_IMTG register (starting an I²C master operation) sets IMBSY to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, IMBSY is reset to 0.

D[4:2] IMSTA[2:0]: I²C Master Status Bits

Indicates the I²C master status.

Table 18.8.4 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

When an operation completion interrupt occurs, read IMSTA[2:0] to check the operation that has been finished. IMSTA[2:0] is automatically reset to 0x0 by writing 1 to IMIF.

D1 IMEIF: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

- 1 (R): Error occurred
- 0 (R): No error (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

IMEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if IMEIE/USI_IMIE register is 1.

An overrun error occurs when a transmit or receive trigger is issued after two-byte data has been received (the first byte data exists in the receive data buffer and the second byte data exists in the shift register) without the receive data buffer being read.

IMEIF is reset by writing 1.

To reset an overrun error, clear IMEIF by writing 1, and then read the receive data buffer (USI_RD register) twice.

D0 IMIF: Operation Completion Flag Bit

Indicates whether the triggered operation has completed or not.

- 1 (R): Completed
- 0 (R): Not completed (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

IMIF is set to 1 when the operation that is specified and triggered using the USI_IMTG register has completed. At the same time an operation completion interrupt request is sent to the ITC if IMIE/USI_IMIE register is 1. IMIF is reset by writing 1.

USI I²C Slave Mode Trigger Register (USI_ISTG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USI I ² C Slave Mode Trigger Register (USI_ISTG)	0x300470 (8 bits)	D7-5	—	reserved	—	—	—	0 when being read.	
		D4	ISTG	I ² C slave operation trigger	1 Trigger 0 Ignored	0	W		
						1 Waiting 0 Finished		R	
		D3	—	reserved	—	—	—	—	0 when being read.
		D2-0	ISTGMOD[2:0]	I ² C slave trigger mode select	ISTGMOD[2:0] Trigger mode	0x0	R/W		
						0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data/ Detect stop 0x2 Transmit data 0x1 reserved 0x0 Wait for start			

Note: This register is effective only in I²C slave mode. Configure USI to I²C slave mode before this register can be used.

D[7:5] Reserved

D4 ISTG: I²C Slave Operation Trigger Bit

Starts an I²C slave operation.

1 (W): Trigger

0 (W): Ignored

1 (R): Waiting for starting operation

0 (R): Trigger has finished (default)

Select an I²C slave operation using ISTGMOD[2:0] and write 1 to ISTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

D3 Reserved**D[2:0] ISTGMOD[2:0]: I²C Slave Trigger Mode Select Bits**

Selects an I²C slave operation.

Table 18.8.5 Trigger List in I²C Slave Mode

ISTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception/stop condition detection
0x2	Data transmission
0x1	Reserved
0x0	Wait for start condition

(Default: 0x0)

USI I²C Slave Mode Interrupt Enable Register (USI_ISIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI I ² C Slave Mode Interrupt Enable Register (USI_ISIE)	0x300471 (8 bits)	D7-2	—	reserved	—	—	—	0 when being read.
		D1	ISEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	ISIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in I²C slave mode. Configure USI to I²C slave mode before this register can be used.

D[7:2] Reserved**D1 ISEIE: Receive Error Interrupt Enable Bit**

Enables interrupt requests to the ITC when an overrun error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D0 ISIE: Operation Completion Interrupt Enable Bit

Enables interrupt requests to the ITC when the triggered operation has completed.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to confirm whether the triggered operation has completed or not using interrupts.

USI I²C Slave Mode Interrupt Flag Register (USI_ISIF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USI I ² C Slave Mode Interrupt Flag Register (USI_ISIF)	0x300472 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.	
		D5	ISBSY	I ² C slave busy flag	1 Busy	0 Standby	0	R	
		D4–2	ISSTA[2:0]	I ² C slave status	ISSTA[2:0]	Status	0x0	R	
					0x7	reserved			
					0x6	NAK received			
					0x5	ACK received			
0x4	ACK/NAK sent								
0x3	End of Rx data								
0x2	End of Tx data								
0x1	Stop detected								
0x0	Start detected								
D1	ISEIF	Overrun error flag	1 Error	0 Normal	0	R/W	Reset by writing 1.		
D0	ISIF	Operation completion flag	1 Completed	0 Not completed	0	R/W			

Note: This register is effective only in I²C slave mode. Configure USI to I²C slave mode before this register can be used.

D[7:6] Reserved

D5 ISBSY: I²C Slave Busy Flag Bit

Indicates the I²C slave operation status.

1 (R): Busy

0 (R): Standby (default)

Writing 1 to ISTG/USI_ISTG register (starting an I²C slave operation) sets ISBSY to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, ISBSY is reset to 0.

D[4:2] ISSTA[2:0]: I²C Slave Status Bits

Indicates the I²C slave status.

Table 18.8.6 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

When an operation completion interrupt occurs, read ISSTA[2:0] to check the operation that has been finished. ISSTA[2:0] is automatically reset to 0x0 by writing 1 to ISIF.

D1 ISEIF: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): Error occurred

0 (R): No error (default)

1 (W): Reset to 0

0 (W): Ignored

ISEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if ISEIE/USI_ISIE register is 1. An overrun error occurs when a transmit or receive trigger is issued after two-byte data has been received (the first byte data exists in the receive data buffer and the second byte data exists in the shift register) without the receive data buffer being read.

ISEIF is reset by writing 1.

To reset an overrun error, clear ISEIF by writing 1, and then read the receive data buffer (USI_RD register) twice.

D0 ISIF: Operation Completion Flag Bit

Indicates whether the triggered operation has completed or not.

- 1 (R): Completed
- 0 (R): Not completed (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

ISIF is set to 1 when the operation that is specified and triggered using the USI_ISTG register has completed. At the same time an operation completion interrupt request is sent to the ITC if ISIE/USI_ISIE register is 1. ISIF is reset by writing 1.

18.9 Precautions

Interface mode setting

Be sure to perform software reset (USIMOD[2:0]/USI_GCFG register = 0x0) and set the interface mode (USIMOD[2:0]/USI_GCFG register = 0x1 to 0x5) before changing other USIL configurations.

Receiving control byte in I²C slave mode

The external I²C master device sends a control byte to the I²C slave device when an ACK has been received after sending a slave address. The subsequent operations of the slave device are determined by the control byte.

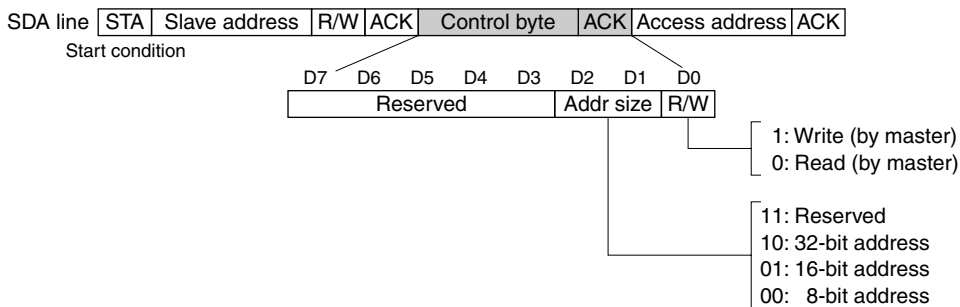


Figure 18.9.1 Control Byte Sent from I²C Master

I²C master write (data receiving from master)

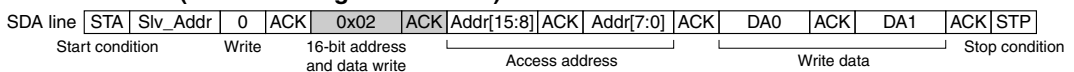


Figure 18.9.2 I²C Master Write (Data Receiving from Master)

The control byte specifies the access address size and writing operations. The received data that follow the control byte should be used as the address and the data to be written according to the access address size.

I²C master read (data transmission to master)

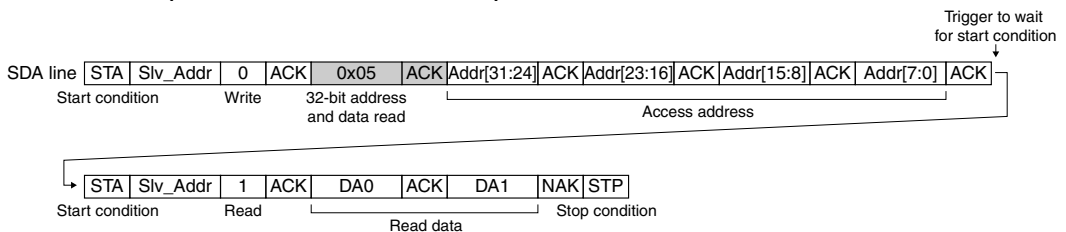


Figure 18.9.3 I²C Master Read (Data Transmission to Master)

The master sends the access address following the control byte. Perform data reception for the control byte and address data to determine the address from which transmit data is read. After sending an ACK for Addr 0, set ISTGMOD[2:0]/USIL_ISTG register to 0x0 and ISTG/USIL_ISTG register to 1 to wait for a start condition that will be sent from the master for reading data (for the slave to send the read data).

19 Universal Serial Interface with LCD Interface (USIL)

19.1 USIL Module Overview

The S1C33L26 incorporates a USIL module that can be configured as a UART, SPI, I²C, LCD SPI, and LCD parallel interface unit by the software switch.

The following shows the main features of USIL:

- Supports seven interface modes: UART, SPI master, SPI slave, I²C master, I²C slave, LCD SPI, and LCD parallel modes.
- Contains one-byte receive data buffer and one-byte transmit buffer.
- Supports both MSB first and LSB first modes.
- UART mode
 - Character length: 7 or 8 bits
 - Parity mode: even, odd, or no parity
 - Stop bit: 1 or 2 bits
 - Start bit: 1 bit fixed
 - Parity error, framing error, and overrun error detectable
 - Can generate receive buffer full, transmit buffer empty, and receive error interrupts.
 - Supports DMA transfer.
- SPI master/slave mode
 - Data length: 8 bits fixed
 - Supports both fast and normal modes (master mode), or normal mode only (slave mode).
 - Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
 - Can generate receive buffer full, transmit buffer empty, and receive error interrupts.
 - Supports DMA transfer.
- I²C master/slave mode
 - 7-bit addressing mode (10-bit addressing is possible by software control.)
 - Supports single master configuration only (master mode).
 - Supports clock stretch/wait functions.
 - Can generate operation (start/stop, data transfer, ACK/NAK transfer) completion interrupts and receive error interrupts.
- LCD SPI mode
 - Data format: 8 bits, 16 bits, 18 bits (4 data format), and 24 bits + CMD bit
 - Supports normal mode only.
 - Supports transmission only.
 - Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
 - Can generate transmit buffer empty interrupts.
 - Supports DMA transfer.
- LCD parallel mode
 - Data bus width: 8 bits
 - Control signals: A0, write, read and chip select signals can be output.
 - Supports byte read and byte write.
 - Configurable access timing parameters (setup, hold, and wait cycles)
 - Can generate write buffer empty and read buffer full interrupts.
 - Supports DMA transfer.

Figure 19.1.1 shows the USIL configuration.

19 UNIVERSAL SERIAL INTERFACE WITH LCD INTERFACE (USIL)

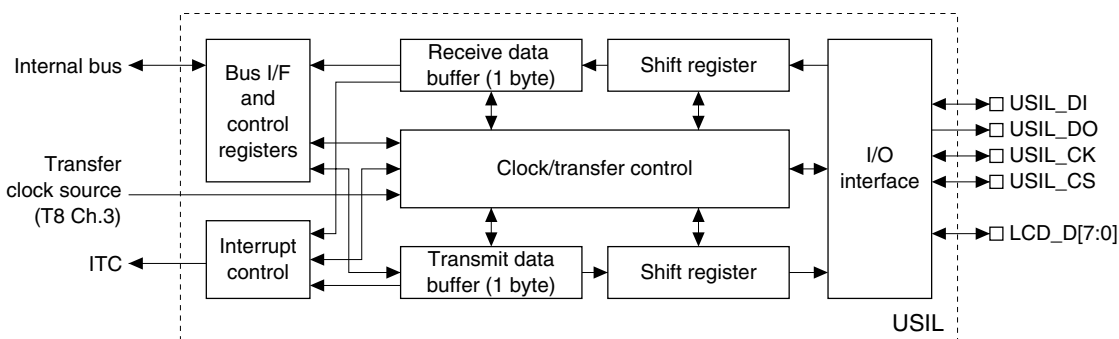


Figure 19.1.1 USIL Configuration

19.2 USIL Pins

Table 19.2.1 lists the USIL input/output pins.

Table 19.2.1 List of USIL Pins

Pin name	USIL mode	Signal name	I/O	Function
USIL_DI	UART	uart_rx	I	Data input pin
	SPI master	spi_di	I	Inputs serial data sent from an external serial device.
	SPI slave	spi_di	I	
	I ² C master	i2c_sda	I/O	Data input/output pin
	I ² C slave	i2c_sda	I/O	Inputs/outputs serial data from/to the I ² C bus. (*)
	LCD SPI	lcds_a0	O	A0 signal output pin
USIL_DO	LCD parallel	lcdp_a0	O	Outputs the A0 signal to the LCD driver/panel.
	UART	uart_tx	O	Data output pin
	SPI master	spi_do	O	Outputs serial data sent to an external serial device.
	SPI slave	spi_do	O	
	I ² C master	–	–	Not used
	I ² C slave	–	–	
USIL_CK	LCD SPI	lcds_do	O	Data output pin Outputs serial data sent to the LCD driver/panel.
	LCD parallel	lcdp_wr	O	Write signal output pin Outputs the write signal to the LCD driver/panel.
	UART	–	–	Not used
	SPI master	spi_ck	O	Clock output pin Outputs the SPI clock.
	SPI slave	spi_ck	I	Clock input pin Inputs an external clock.
	I ² C master	i2c_sck	I/O	SCL input/output pin Inputs SCL line status from the I ² C bus. Also outputs the I ² C clock.
USIL_CS	I ² C slave	i2c_sck	I/O	SCL input/output pin Inputs SCL line status from the I ² C bus. Also outputs a clock stretch condition.
	LCD SPI	lcds_ck	O	Clock output pin Outputs the SPI clock.
	LCD parallel	lcdp_rd	O	Read signal output pin Outputs the read signal to the LCD driver/panel.
	UART	–	–	Not used
	SPI master	–	–	
	SPI slave	#spi_ss	I	SPI slave select signal input pin Low level input to this pin selects USIL (in SPI slave mode) as an SPI slave device.
USIL_CS	I ² C master	i2c_sda	I/O	Data input/output pin
	I ² C slave	i2c_sda	I/O	Inputs/outputs serial data from/to the I ² C bus. (*)
	LCD parallel	lcdp_cs	O	Chip select signal output pin Outputs the chip select signal to the LCD driver/panel.

Pin name	USIL mode	Signal name	I/O	Function
LCD_D[7:0]	UART	–	–	Not used
	SPI master	–	–	
	SPI slave	–	–	
	I ² C master	–	–	
	I ² C slave	–	–	
	LCD SPI	–	–	
	LCD parallel	lcdp_da[7:0]	I/O	8-bit data input/output pins Inputs/outputs 8-bit parallel data from/to the LCD driver/panel.

* When USIL is configured to I²C master or slave mode, either the USIL_DI pin or the USIL_CS pin can be used as the data input/output pin. Note, however, that both the USIL_DI and USIL_CS pins cannot be used as the data input/output pin simultaneously.

Note: Use a GPIO port to output the slave select signal when USIL is configured to SPI master mode.

The USIL input/output pins (USIL_DI, USIL_DO, USIL_CK, USIL_CS, LCD_D[7:0]) are shared with I/O ports and are initially set as general-purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as USIL input/output pins.

For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

19.3 USIL Clock Sources

Operating clock

The USIL uses PCLK2 as the operating clock. Therefore, PCLK2 must be supplied from the CMU before starting the USIL including setting the control registers. For more information on the PCLK2 supply, refer to the “Clock Management Unit (CMU).”

Transfer clock

When the USIL is configured to a UART, SPI master (normal mode), I²C master, LCD SPI, or LCD parallel interface, the source clock for transfer is supplied by the 8-bit programmable timer (T8 Ch.3). Program T8 Ch.3 according to the transfer rate and enable supplying the source clock to the USIL module. The USIL module divides the source clock to generate the transfer clock (or sampling clock). Be aware that the division ratio in the USIL depends on the interface mode.

When the USIL is configured to an SPI master (fast mode), PCLK2 is used as the source clock.

When the USIL is configured to an SPI slave or I²C slave device, the transfer clock is supplied from the external master device. However, SPI slave mode uses PCLK2 and I²C slave mode uses the T8 Ch.3 output clock to generate the sampling signal.

Table 19.3.1 USIL Clocks

Clock	Interface mode	Clock source
Operating clock	UART	PCLK2
	SPI master	PCLK2
	SPI slave	PCLK2
	I ² C master	PCLK2
	I ² C slave	PCLK2
	LCD SPI	PCLK2
	LCD parallel	PCLK2
Transfer/sampling clock source (division ratio in USIL)	UART	T8 Ch.3 (f _{SOURCE} /8)
	SPI master	Normal mode: T8 Ch.3 (f _{SOURCE} /2)
		Fast mode: PCLK2 (f _{PCLK2})
	SPI slave	PCLK2 (f _{PCLK2} /4) for sampling
	I ² C master	T8 Ch.3 (f _{SOURCE} /8)
	I ² C slave	T8 Ch.3 (f _{SOURCE}) for sampling
	LCD SPI	T8 Ch.3 (f _{SOURCE} /2)
LCD parallel	T8 Ch.3 (f _{SOURCE})	

UART mode, I²C master mode

$$\text{bps} = \text{f}_{\text{SYS_CLK}} \times \text{DF} / \{(\text{TR} + 1) \times 8 + \text{TFMD}\}$$

$$\text{TR} = (\text{f}_{\text{SYS_CLK}} \times \text{DF} / \text{bps} - \text{TFMD} - 8) / 8$$

SPI master mode, LCD SPI mode

$$\text{bps} = \text{f}_{\text{SYS_CLK}} \times \text{DF} / \{(\text{TR} + 1) \times 2 + \text{TFMD}\}$$

$$\text{TR} = (\text{f}_{\text{SYS_CLK}} \times \text{DF} / \text{bps} - \text{TFMD} - 2) / 2$$

f_{SOURCE}: T8 Ch.3 output clock frequency [Hz]

f_{SYS_CLK}: System clock frequency [Hz]

bps: Transfer rate [bps]

DF: Division ratio set by DF[3:0]/T8_CLK3 register (T8 Ch.3)

TR: Reload data to be set to the T8_TR3 register (T8 Ch.3)

TFMD: Fine mode set value at TFMD[3:0]/T8_CTL3 register (T8 Ch.3)

Example: UART mode, transfer rate = 115,200 bps, system clock = 33 MHz, DF[3:0]/T8_CLK3 register setting (T8 Ch.3) = 1/1, TFMD[3:0]/T8_CTL3 register setting (T8 Ch.3) = 14

$$\text{TR} = (33,000,000 \times 1 / 115,200 - 14 - 8) / 8 = 33.05 (= 0x21)$$

For more information on controlling the T8 module, refer to the “8-bit Timers (T8)” chapter.

Note: When the USIL is set to I²C slave mode, i2c_sck (I²C clock) is supplied from the external I²C master. The T8 output clock frequency (f_{SOURCE}) should be determined according to the i2c_sck frequency.

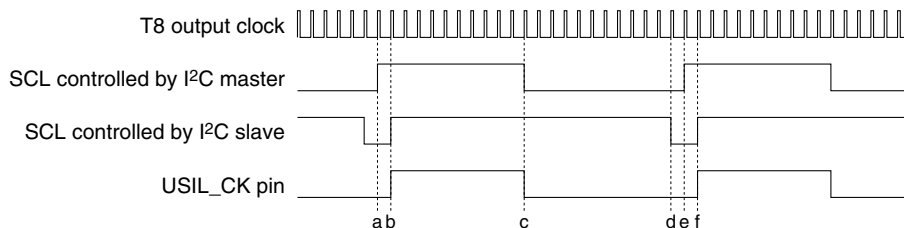


Figure 19.3.1 I²C Clock in I²C Slave Mode

$$\text{Tbf} = \text{Ti2c_baud_rate}$$

$$\text{Tbc} = \text{Ti2c_baud_rate_high}$$

$$\text{Tcf} = \text{Ti2c_baud_rate_low}$$

Tce: The I²C master occupies the SCL line by driving it to low.

Tac: The I²C master releases the SCL line.

Tdf: In order to finish the internal operations, the I²C slave occupies the SCL line for two source clock (T8 output clock) cycles by driving it to low after detecting that the I²C master drives the SCL line to low.

The T8 output clock frequency (f_{SOURCE}) must be set so that the conditions shown below are satisfied.

$$\text{f}_{\text{SOURCE}} > 3/\text{Tbc}$$

$$\text{f}_{\text{SOURCE}} > 4/\text{Tce}$$

Be aware that the actual SCL signal will be delayed, as the I²C slave forcibly drives the SCL line to low. The figure below shows an example in which the timing becomes worse.

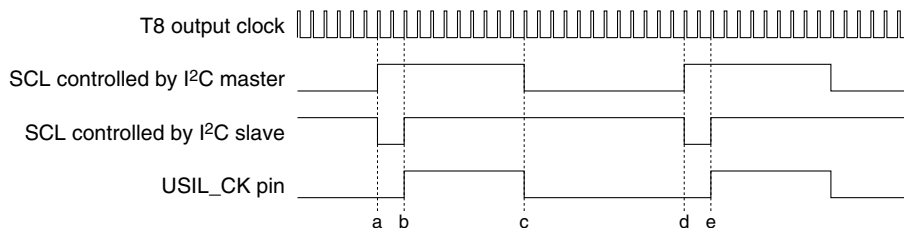


Figure 19.3.2 Example of Delayed I²C Clock

19.4 USIL Module Settings

Make the following settings before starting data transfers using the USIL module.

- (1) Program the clock source module to supply the clock required to the USIL module. (See Section 19.3.)
- (2) Reset the USIL module.
- (3) Set the USIL interface mode and a general condition (MSB first/LSB first) to be applied to all interface modes.
- (4) Set the data format and operating conditions for the interface mode selected.
- (5) Set interrupt and DMA transfer conditions if necessary. (See Section 19.7.)
- (6) Select the port functions to be used for USIL according to the interface mode. (See Section 19.2 and the “I/O Ports (GPIO)” chapter.)

19.4.1 USIL Module Software Reset

Writing 0x0 to USILMOD[2:0]/USIL_GCFG register resets the USIL module circuits. Be sure to perform software reset before setting the interface mode.

19.4.2 Interface Mode

The USIL module provides seven serial interface functions shown in Section 19.1 and the interface mode can be configured to one of them using the USILMOD[2:0]/USIL_GCFG register.

Table 19.4.2.1 Interface Mode Selection

USILMOD[2:0]	Interface mode
0x7	LCD parallel
0x6	LCD SPI
0x5	I ² C slave
0x4	I ² C master
0x3	SPI slave
0x2	SPI master
0x1	UART
0x0	Software reset

(Default: 0x0)

- Notes:**
- Be sure to perform software reset and set the interface mode before changing other USIL configurations.
 - The LCD parallel, UART, I²C master, and I²C slave must be set before configuring the I/O port for USIL.

19.4.3 General Settings for All Interface Modes

MSB first/LSB first selection

Use LSBFST/USIL_GCFG register to select whether the data MSB or LSB is input/output first.

LSB first is selected when LSBFST is set to 0 (default). MSB first is selected when LSBFST is set to 1.

19.4.4 Settings for UART Mode

When the USIL is used in UART mode, configure the data length, stop bit, and parity bit. The start bit length is fixed at 1 bit.

Data length

Use UCHLN/USIL_UCFG register to select the data length. Setting UCHLN to 0 (default) configures the data length to 7 bits. Setting UCHLN to 1 configures it to 8 bits.

Stop bit

Use USTPB/USIL_UCFG register to select the stop bit length. Setting USTPB to 0 (default) configures the stop bit length to 1 bit. Setting USTPB to 1 configures it to 2 bits.

Parity bit

Use UPREN/USIL_UCFG register to select whether the parity function is enabled or not. Setting UPREN to 0 (default) disables the parity function. In this case, no parity bit will be added to transfer data and receive data will not be checked for parity. Setting UPREN to 1 enables the parity function. In this case, a parity bit will be added to transfer data and receive data will be checked for parity.

When the parity function is enabled, the parity mode should be selected using UPMD/USIL_UCFG register. Setting UPMD to 0 (default) adds a parity bit and checks for odd parity. Setting UPMD to 1 adds a parity bit and checks for even parity.

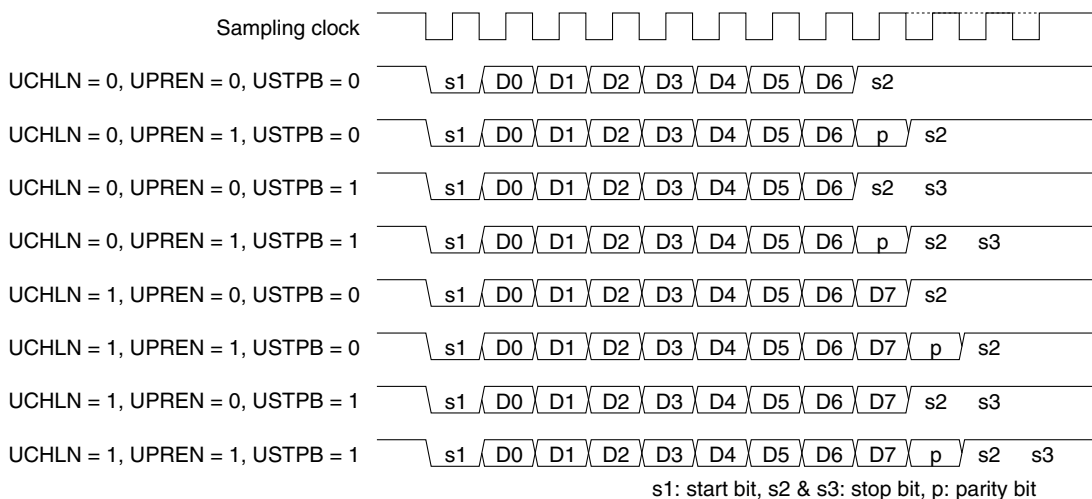


Figure 19.4.4.1 Transfer Data Format in UART Mode (LSB first)

19.4.5 Settings for SPI Mode

When the USIL is used in SPI mode (master or slave), configure the SPI clock polarity/phase. When used in SPI master mode, select the clock mode.

Note that the data length in SPI mode is fixed at 8 bits.

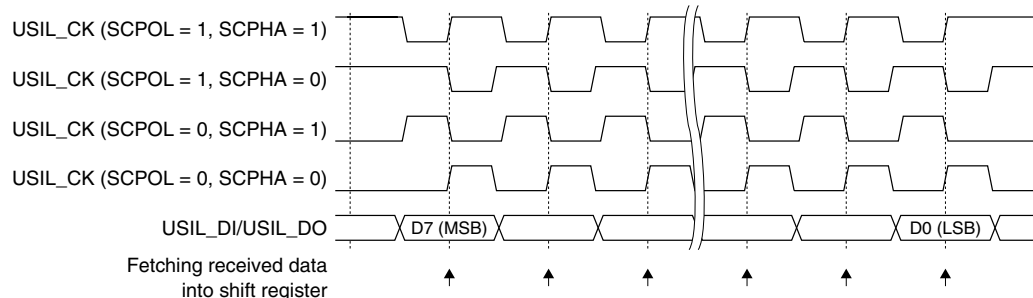
SPI clock polarity and phase settings (master mode and slave mode)

Use SCPOL/USIL_SCFG register to select the SPI clock polarity. Setting SCPOL to 1 treats the SPI clock as active low. Setting it to 0 (default) treats it as active high.

The SPI clock phase can be selected using SCPHA/USIL_SCFG register.

These control bits set transfer timing as shown in Figure 19.4.5.1.

Master mode



Slave mode

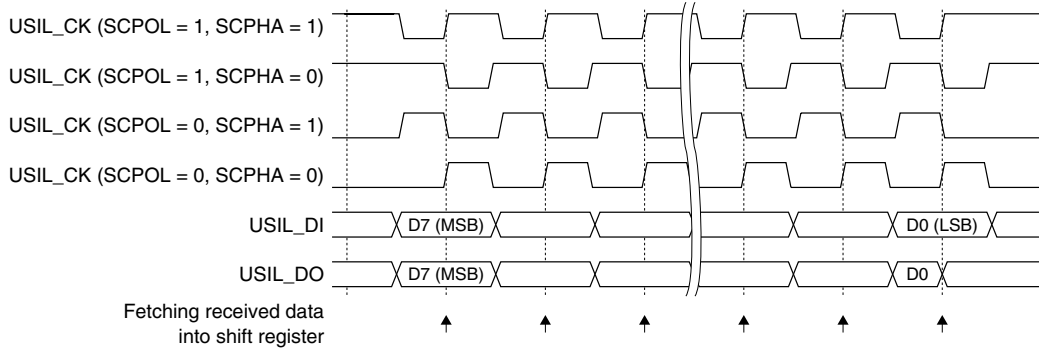


Figure 19.4.5.1 Clock and Data Transfer Timing (MSB first)

Clock mode (master mode only)

In SPI master mode, either normal or fast clock mode can be selected using SFSTMOD/USIL_SCFG register. Setting SFSTMOD to 0 (default) places the USIL into normal mode and the USIL generates the transfer clock by dividing the T8 output by 2. Setting SFSTMOD to 1 places the USIL into fast mode and the USIL uses PCLK2 supplied from the CMU directly as the transfer clock. The fast mode does not use the T8.

The SPI slave mode uses the T8 output clock for generating the sampling clock.

19.4.6 Settings for I²C Mode

The I²C mode does not need to set data format and other conditions. The data length in I²C mode is fixed at 8 bits.

19.4.7 Settings for LCD SPI Mode

When the USIL is used in LCD SPI mode, configure the SPI clock polarity/phase and the data format.

SPI clock polarity and phase settings

Use LSCPOL/USIL_LSCFG register to select the SPI clock polarity. Setting LSCPOL to 1 treats the SPI clock as active low. Setting it to 0 (default) treats it as active high.

The SPI clock phase can be selected using LSCPHA/USIL_LSCFG register.

These control bits set transfer timing as shown in Figure 19.4.7.1.

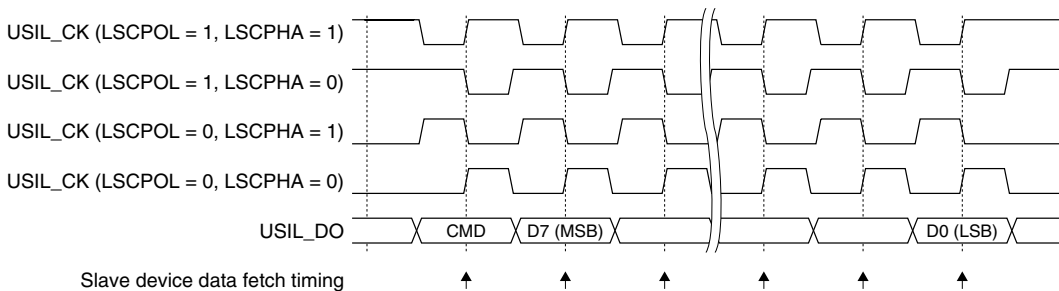


Figure 19.4.7.1 Clock and Data Transfer Timing (Command bit enabled, MSB first)

Data format

In LCD SPI mode, the display data format can be selected using LSDMOD[1:0]/USIL_LSDCFG register.

Table 19.4.7.1 LCD SPI Data Mode

LSDMOD[1:0]	Data mode
0x3	24-bit mode
0x2	18-bit mode
0x1	16-bit mode
0x0	8-bit mode

(Default: 0x0)

The 18-bit mode supports four data formats and one of them can be selected using LS18DFM[1:0]/USIL_LSD-CFG register.

Table 19.4.7.2 LCD SPI 18-bit Data Format

LS18DFM[1:0]	Data format
0x3	Format 3
0x2	Format 2
0x1	Format 1
0x0	Format 0

(Default: 0x0)

To send data prefixed with a CMD (command) bit, set LSCMDEN/USIL_LSCFG register to 1. The command bit is used for controlling the SPI LCD driver/panel connected to the USIL. The command bit value to be sent can be specified using LSCMD/USIL_LSCFG register. Setting LSCMD to 1 configures the command bit to high. Setting LSCMD to 0 configures the command bit to low.

When using an LCD driver/panel that supports the A0 signal (command/data select signal), set LSCMDEN to 0 (default) to disable sending the CMD bit on the data line. In this case, the USIL_DI pin is used as the A0 signal output pin. The USIL_DI pin goes high when LSCMD is set to 1, and it goes low when LSCMD is set to 0.

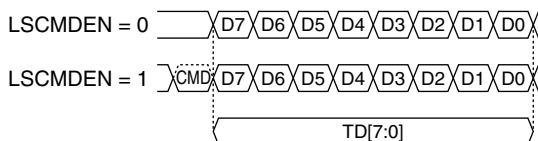
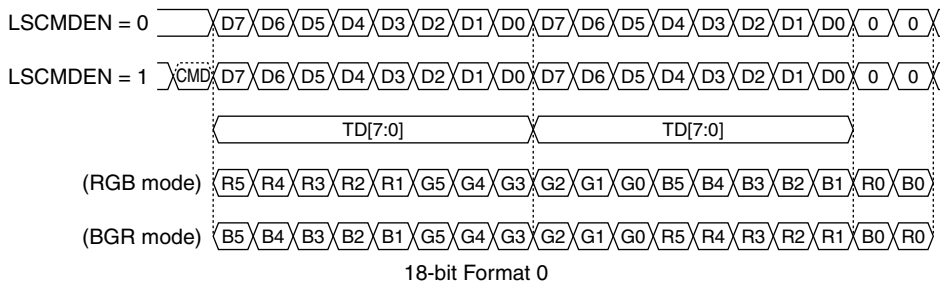


Figure 19.4.7.2 8-bit Data Format



Figure 19.4.7.3 16-bit Data Format



18-bit Format 0



18-bit Format 1

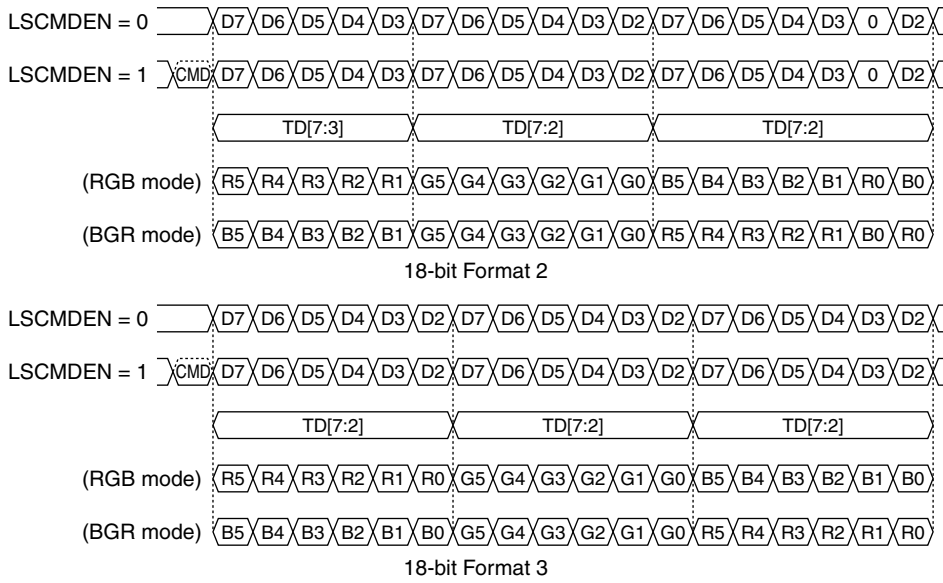


Figure 19.4.7.4 18-bit Data Format

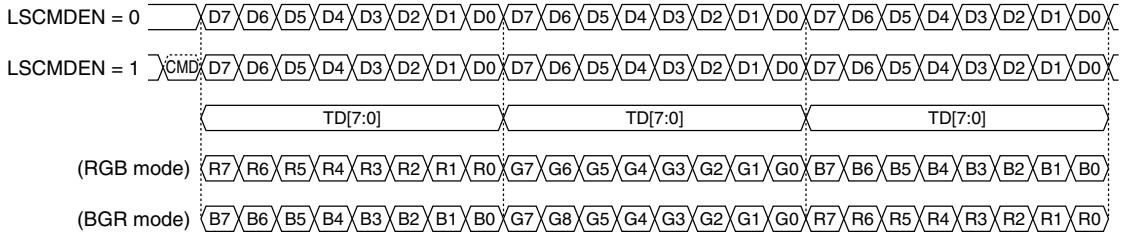


Figure 19.4.7.5 24-bit Data Format

19.4.8 Settings for LCD Parallel Mode

When the USIL is used in LCD parallel mode, configure the access timing parameters.

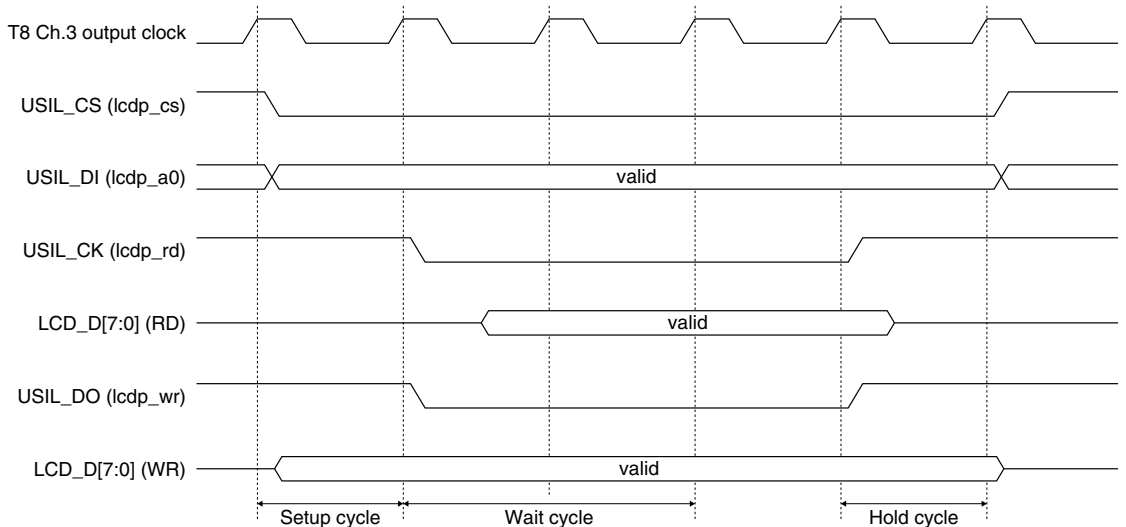


Figure 19.4.8.1 Access Timing Parameters

Setup cycle

The setup cycle can be set to 1–4 cycles using LPST[1:0]/USIL_LPAC register.

Table 19.4.8.1 Setup Cycle Settings

LPST[1:0]	Number of setup cycles
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x0)

Hold cycle

The hold cycle can be set to 1–4 cycles using LPHD[1:0]/USIL_LPAC register.

Table 19.4.8.2 Hold Cycle Settings

LPHD[1:0]	Number of hold cycles
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x0)

Wait cycle

The wait cycle can be set to 0–15 cycles using LPWT[3:0]/USIL_LPAC register.

Table 19.4.8.3 Wait Cycle Settings

LPWT[3:0]	Number of wait cycles
0xf	15 cycles
0xe	14 cycles
:	:
0x2	2 cycles
0x1	1 cycle
0x0	0 cycles

(Default: 0x0)

19.5 Data Transfer Control

This section describes how to control data transfers. The following explanations assume that the configurations described above and interrupt/DMA settings have already been finished.

19.5.1 Data Transfer in UART Mode

Data transmission

To start data transmission in UART mode, write the transmit data to the transmit data buffer (TD[7:0]/USIL_TD register).

The buffer data is sent to the transmit shift register, and the start bit is output from the USIL_DO pin. The data in the shift register is then output in sequence. Following output of the eighth data bit, the parity bit (if parity is enabled) and the stop bit are output.

The transmitter circuit includes two status flags: UTDIF/USIL_UIF register and UTBSY/USIL_UIF register.

The UTDIF flag indicates the transmit data buffer status. This flag is set to 1 indicating that the transmit data buffer becomes empty when data written to the transmit data buffer is sent to the transmit shift register. UTDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 19.7). Write subsequent data to the transmit data buffer to start the following transmission using this interrupt or DMA. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. If an interrupt or DMA is not used for transmission, be sure to confirm that the transmit data buffer is empty before writing transmit data. Writing data before UTDIF has been set will overwrite earlier transmit data inside the transmit data buffer. After UTDIF is set to 1, it can be reset to 0 by writing 1.

The UTBSY flag indicates the USIL status in UART mode. This flag switches to 1 when transmit data is written to the transmit buffer and reverts to 0 after both the shift register and transmit buffer become empty.

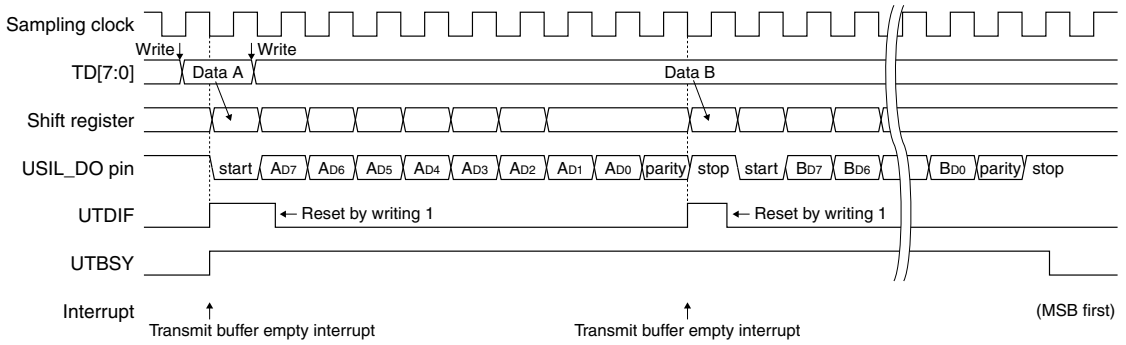


Figure 19.5.1.1 Data Transmission Timing Chart (UART mode)

Data reception

When the external serial device sends a start bit, the receiver circuit detects its low level and starts sampling the following data bits. Once the 8-bit data has been received into the shift register, the received data is loaded into the receive data buffer (RD[7:0]/USIL_RD register). If parity checking is enabled, the receiver circuit checks the received data at the same time by checking the parity bit received immediately after the eighth data bit. The receiver circuit includes two status flags: URDIF/USIL_UIF register and URBSY/USIL_UIF register.

The URDIF flag indicates the receive data buffer status. This flag is set to 1 indicating that the received data can be read out when data received in the shift register is loaded to the receive data buffer. URDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 19.7). Read the received data from the receive data buffer using this interrupt or DMA. The receive data buffer size is 1 byte, therefore the received data must be read before the subsequent data reception has completed. Furthermore, URDIF must be reset by writing 1. If the next reception is completed when URDIF is 1 and the receive data buffer (USIL_RD register) is not read, an overrun error occurs (at the time stop bit has been received).

The URBSY flag indicates the shift register status. This flag is set to 1 while data is being received in the shift register and reverts to 0 once the received data is loaded to the receive data buffer. Read this flag to check whether the receiver circuit is operating or at standby.

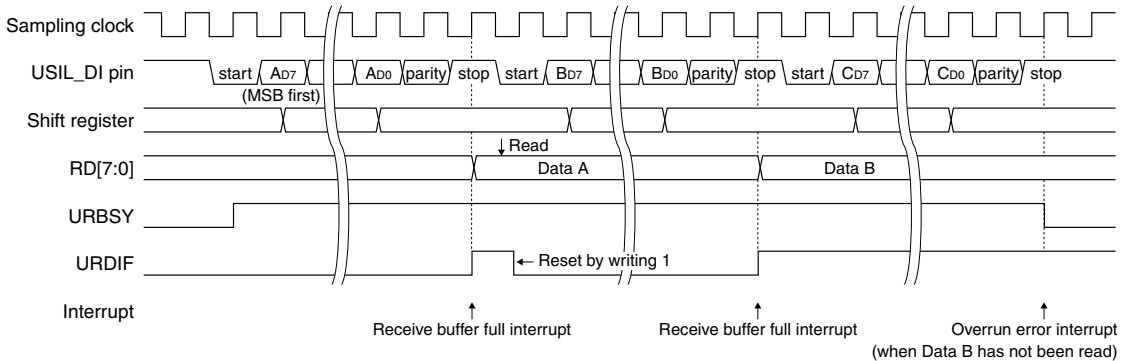


Figure 19.5.1.2 Data Receiving Timing Chart (UART mode)

19.5.2 Data Transfer in SPI Mode

Data transmission

To start data transmission in SPI mode, write the transmit data to the transmit data buffer (TD[7:0]/USIL_TD register).

The buffer data is sent to the transmit shift register. In SPI master mode, the module starts clock output from the USIL_CK pin. In SPI slave mode, the module awaits clock input from the USIL_CK pin. The data in the shift register is shifted in sequence at the clock rising or falling edge (see Figure 19.4.5.1) and sent from the USIL_DO pin.

The SPI controller includes two status flags for transfer control: STDIF/USIL_SIF register and SSIF/USIL_SIF register.

The STDIF flag indicates the transmit data buffer status. STDIF is set to 1 indicating that the transmit data buffer becomes empty when data written to the transmit data buffer is sent to the transmit shift register. STDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 19.7). Write subsequent data to the transmit data buffer to start the following transmission using this interrupt or DMA. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. If an interrupt or DMA is not used for transmission, be sure to confirm that the transmit data buffer is empty before writing transmit data. Writing data before STDIF has been set will overwrite earlier transmit data inside the transmit data buffer.

In SPI master mode, the SSIF flag indicates the USIL status. This flag switches to 1 when transmit data is written to the transmit buffer and reverts to 0 after both the shift register and transmit buffer become empty. Read this flag to check whether the SPI controller is operating or at standby.

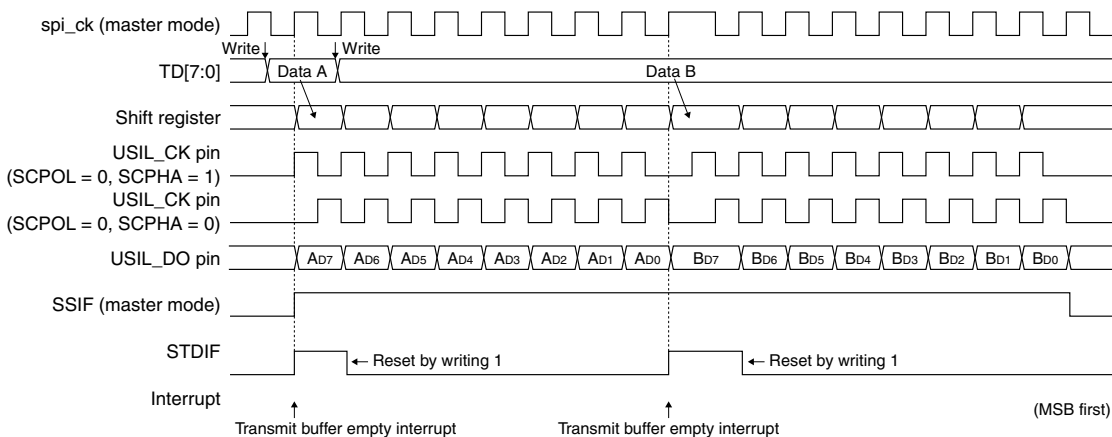


Figure 19.5.2.1 Data Transmission Timing Chart (SPI mode)

Data reception

In SPI master mode, write dummy data to the transmit data buffer. Writing to the transmit data buffer creates the trigger for reception as well as transmission start. Writing actual transmit data enables simultaneous transmission and reception. This starts the SPI clock output from the USIL_CK pin.

In SPI slave mode, the module waits until the clock is input from the USIL_CK pin. There is no need to write to the transmit data buffer if no transmission is required. The receiving operation is started by the clock input from the master device. If data is transmitted simultaneously, write transmit data to the transmit data buffer before the clock is input.

The data is received in sequence in the shift register at the SPI clock edge (see Figure 19.4.5.1). The received data is loaded into the receive data buffer once the 8 bits of data are received in the shift register.

The received data in the buffer can be read from RD[7:0]/USIL_RD register.

The SPI controller includes two status flags for transfer control: SRDIF/USIL_SIF register and SSIF/USIL_SIF register.

The SRDIF flag indicates the receive data buffer status. This flag is set to 1 when the data received in the shift register is loaded into the receive data buffer, indicating that the received data can be read out. SRDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 19.7). Read the received data from the receive data buffer using this interrupt or DMA. The receive data buffer size is 1 byte, therefore the received data must be read before the subsequent data reception has completed. Furthermore, SRDIF must be reset by writing 1. While SRDIF is set to 1, the next received data will not be transferred from the shift register to the receive data buffer (the first byte data exists in the receive data buffer and the second byte data exists in the shift register). An overrun error occurs if the third byte data is received in this condition, as the second byte data in the shift register is corrupted (an overrun error occurs at the time the first bit of the third byte is fetched).

In SPI master mode, the SSIF flag indicates the shift register status. This flag switches to 1 at the beginning of data reception and reverts to 0 once the data is received. Read this flag to check whether the SPI controller is operating or at standby.

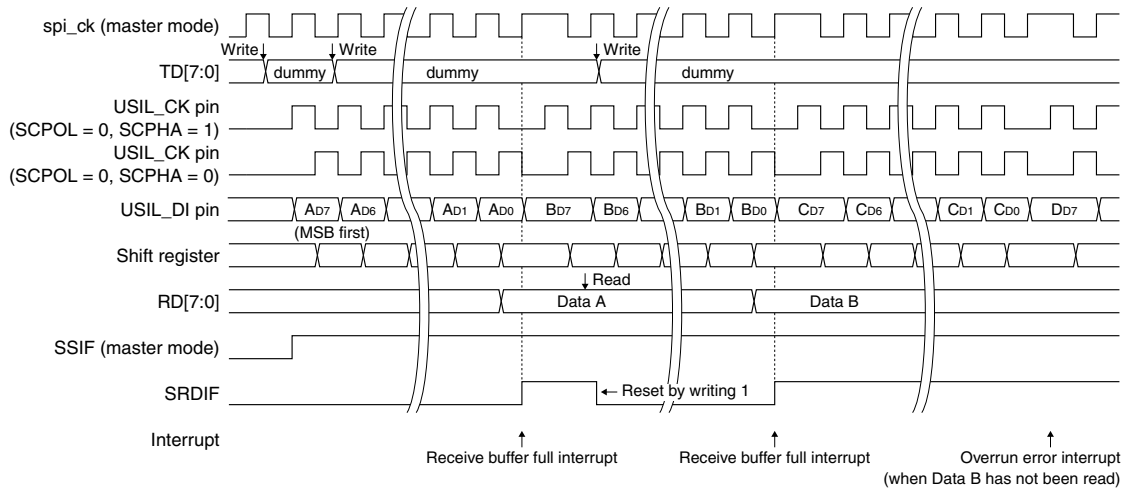


Figure 19.5.2.2 Data Receiving Timing Chart (SPI mode)

Slave select signal

In SPI slave mode, data transmission/receiving operations are enabled when the master device’s slave select signal input to the USIL_CS pin is low. When the slave select signal is high, the SPI controller does not start data transfer even if the clock is input to the USIL_CK pin from the master device. The slave select signal status can be checked using SSIF/USIL_SIF register (it functions as the shift register status flag in SPI master mode). SSIF goes 1 when the slave select signal is inactive (high); it goes 0 when the slave select signal is active (low).

If a slave select output is required in SPI master mode, use a general-purpose I/O port and control its output with software.

19.5.3 Data Transfer in I²C Mode

Control method in I²C master mode

Data transfer in I²C master mode is controlled using IMTGMOD[2:0]/USIL_IMTG register and IMTG/USIL_IMTG register. Select an I²C master operation using IMTGMOD[2:0] and write 1 to IMTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

Table 19.5.3.1 Trigger List in I²C Master Mode

IMTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception
0x2	Data transmission
0x1	Stop condition
0x0	Start condition

(Default: 0x0)

Writing 1 to IMTG sets IMBSY/USIL_IMIF register to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, IMBSY is reset to 0. At the same time, the interrupt flag (IMIF/USIL_IMIF register) is also set to 1. After an interrupt occurs, read the status bits (IMSTA[2:0]/USIL_IMIF register) to check the operation finished. Then clear IMIF by writing 1. IMSTA[2:0] will be automatically cleared to 0x0.

Table 19.5.3.2 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

Data transmission in I²C master mode

The following describes the data transmission procedure in I²C master mode.

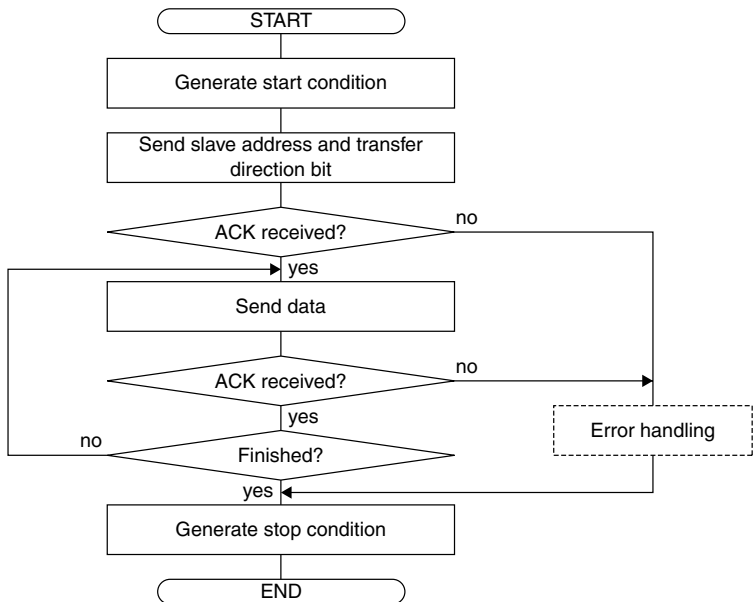
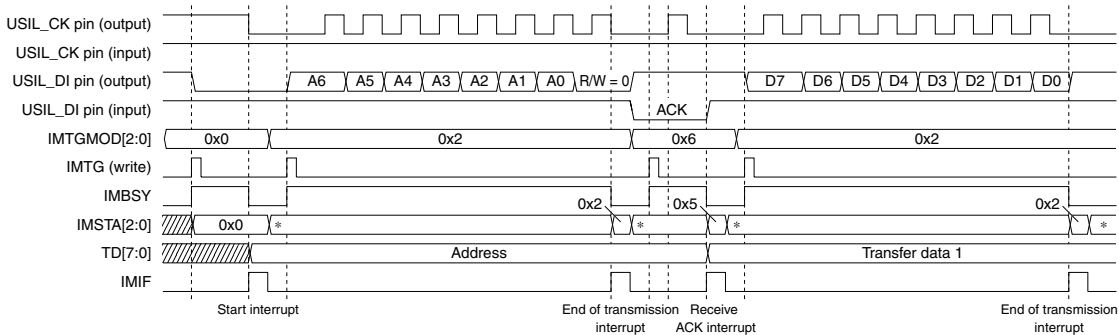
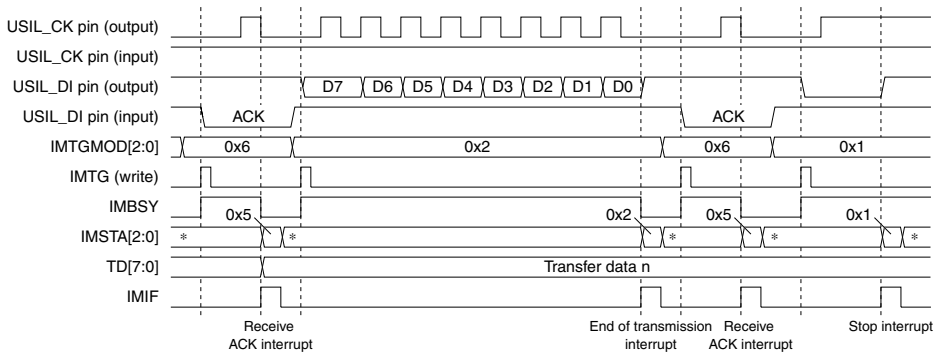


Figure 19.5.3.1 I²C Master Data Transmission Flow Chart



(1) Start condition → Data transmission



(2) Data transmission → Stop condition

* When IMIF is cleared via software, IMSTA[2:0] is also cleared to 0x0.

Figure 19.5.3.2 I²C Master Data Transmission Timing Chart

(1) Generating start condition

I²C data transfer starts when the I²C master device generates a start condition. The start condition applies when the SCL line is maintained at high and the SDA line is pulled down to low.

To generate a start condition in this I²C master, set IMTGMOD[2:0] to 0x0 (default) and write 1 to IMTG.

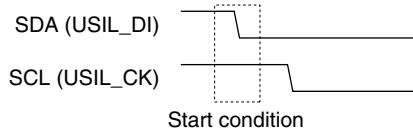


Figure 19.5.3.3 Start Condition

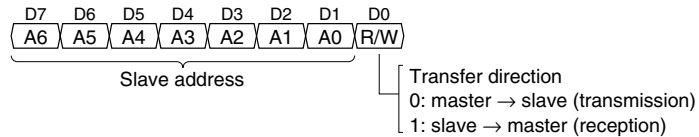
IMBSY is set to 1 while a start condition is being generated. When the start condition is generated, IMBSY is reset to 0 and IMSTA[2:0] is set to 0x0. The I²C bus is busy from this point on.

Note: Other operations cannot be started before a start condition is generated.

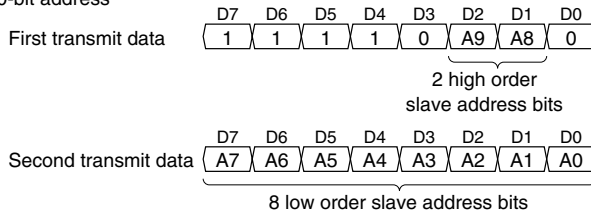
(2) Sending slave address and transfer direction bit

After a start condition has been generated, send the address of the slave device to be communicated and a transfer direction bit. I²C slave addresses are either 7-bit or 10-bit. This module uses an 8-bit transfer data buffer to send the slave address and transfer direction bit, enabling single transfers in 7-bit address mode. In 10-bit mode, data is sent twice or three times under software control. Figure 19.5.3.4 shows the configuration of the address data.

7-bit address



10-bit address



When the I²C master performs data reception, issue a repeated start condition after the second data has been sent and then send the third data as shown below.

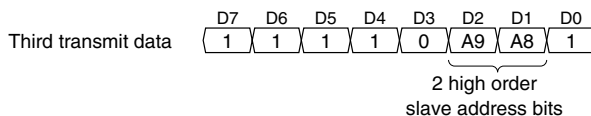


Figure 19.5.3.4 Transmit Data Specifying Slave Address and Transfer Direction

The transfer direction bit indicates the data transfer direction after the slave address has been sent. Set this bit to 0 when sending data from the master to the slave.

To send a slave address, set the address with the transfer direction bit to the transmit data buffer (TD[7:0]/USIL_TD register). Then set IMTGMOD[2:0] to 0x2 and write 1 to IMTG.

To send a 10-bit address, execute this procedure twice or three times as shown in Figure 19.5.3.4.

Writing 1 to IMTG sets IMBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x2. Confirm that the slave address (each byte) has been sent by reading IMBSY or using an interrupt.

After a slave address has been sent, the selected slave device sends back an ACK by pulling down the SCL line to low. If the SCL line maintains high, it is regarded as a NAK. In this case, the I²C controller cannot communicate with the slave device specified.

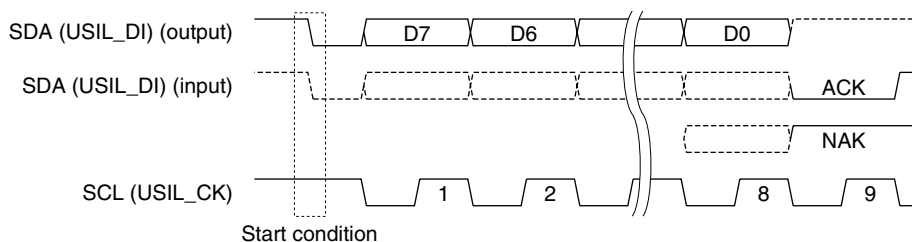


Figure 19.5.3.5 ACK and NAK

It is necessary to check that an ACK has been received before sending data. To do this, set IMTGMOD[2:0] to 0x6 and write 1 to IMTG after the slave address has been sent.

IMBSY is set to 1 while an ACK/NAK is being detected and it reverts to 0 when the detection has completed. Receiving an ACK sets IMSTA[2:0] to 0x5; receiving a NAK sets it to 0x6. Check IMSTA[2:0] after confirming IMBSY or using an interrupt. When an ACK has been received, perform data transmission. When a NAK has been received, perform an error handling.

(3) Data transmission

The data transmission procedure is the same as that of the slave address transmission.

1. Write an 8-bit transmit data to the transmit data buffer (TD[7:0]).
2. Set IMTGMOD[2:0] to 0x2 and IMTG to 1.

This trigger transfers the buffer data to the transmit shift register to start transmission. The module starts clock output from the USIL_CK pin. The data in the shift register is shifted in sequence with the clock and sent from the USIL_DO pin.

Writing 1 to IMTG sets IMBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x2 (end of transmit data). An interrupt request can be generated at this point. Write subsequent data to the transmit data buffer to start the following transmission using this interrupt.

However, as in the case of the slave address transmission, check that the slave device has sent back an ACK (by setting IMTGMOD[2:0] to 0x6 and IMTG to 1) before starting the following 8-bit data transmission.

Repeat an 8-bit data transmission and ACK receiving check for the required number of times.

(4) Generating stop condition

To end I²C communication after all data has been sent, the I²C master must generate a stop condition. The stop condition applies when the SCL line is maintained at high and the SDA line is pulled up from low to high. To generate a stop condition in this I²C master, set IMTGMOD[2:0] to 0x1 and write 1 to IMTG.

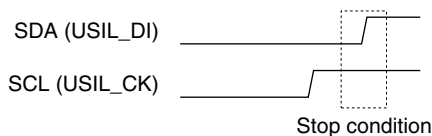


Figure 19.5.3.6 Stop Condition

IMBSY is set to 1 while a stop condition is being generated. When the stop condition is generated, IMBSY is reset to 0 and IMSTA[2:0] is set to 0x1. Read IMBSY or use an interrupt to check that a stop condition has been generated. The I²C bus subsequently switches to free state.

(5) Generating repeated start condition

To make it possible to continue with a different data transfer after a data transmission has completed, the I²C master can omit stop condition generation and generate a repeated start condition. To generate a repeated start condition, perform a start condition generation procedure described in Step (1). Slave address transmission is subsequently possible with the I²C bus remaining in the busy state.

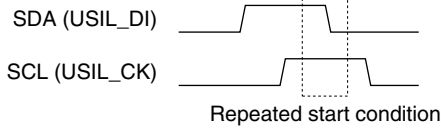


Figure 19.5.3.7 Repeated Start Condition

Data reception in I²C master mode

The following describes the data receiving procedure in I²C master mode.

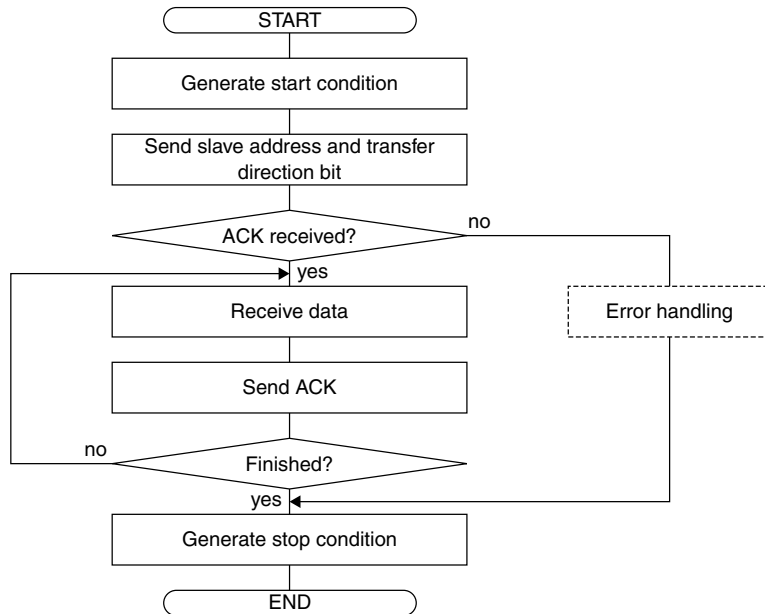
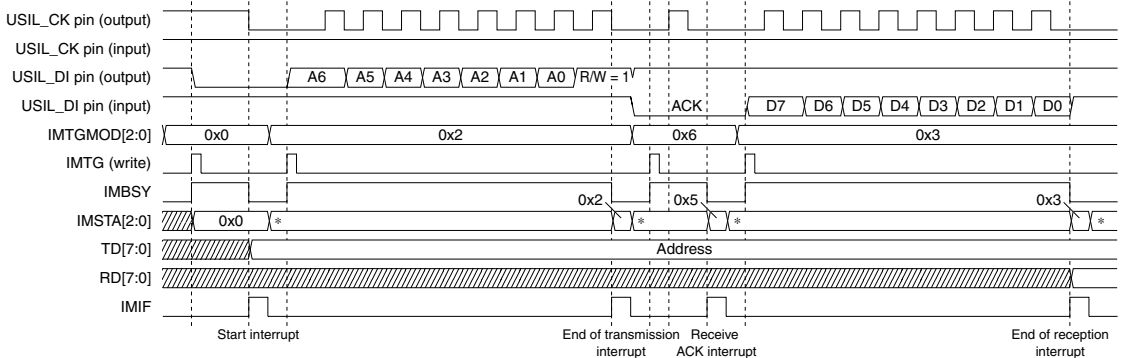
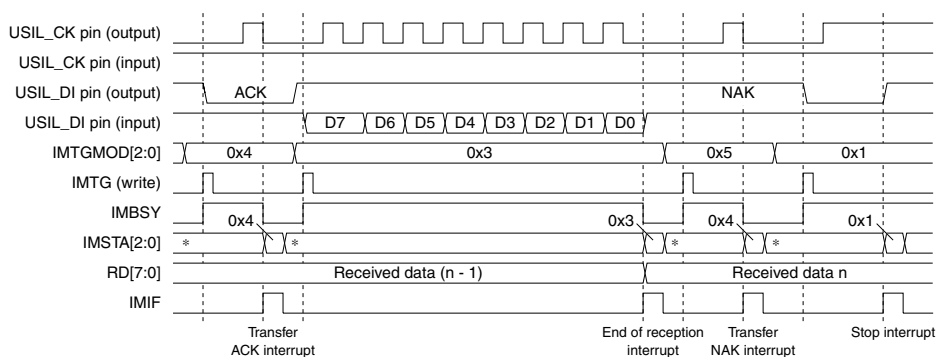


Figure 19.5.3.8 I²C Master Data Receiving Flow Chart



(1) Start condition → Data reception

19 UNIVERSAL SERIAL INTERFACE WITH LCD INTERFACE (USIL)



(2) Data reception → Stop condition

* When IMIF is cleared via software, IMSTA[2:0] is also cleared to 0x0.

Figure 19.5.3.9 I²C Master Data Receiving Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See “Receiving control byte in I²C slave mode” in “19.9 Precautions.”

(1) Generating start condition

The procedure is the same as that of data transmission in I²C master mode.

(2) Sending slave address and transfer direction bit

The procedure is the same as that of data transmission in I²C master mode. However, send the slave address with the transfer direction bit set to 1. Then check that the slave device sends back an ACK.

(3) Data reception

To start data reception, set IMTGMOD[2:0] to 0x3 and write 1 to IMTG.

This trigger starts outputting 8 clocks from the USIL_CK pin. The USIL_DO pin status is sampled in sync with the clock and loaded to the shift register. The received data is loaded to the receive data buffer (RD[7:0]/USIL_RD register) once the 8-bit data has been received in the shift register.

Writing 1 to IMTG sets IMBSY to 1. When the received data is loaded to the receive data buffer, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x3 (end of receive data). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

It is necessary to send back an ACK or NAK to the slave device after an 8-bit data has been received. To send back an ACK, set IMTGMOD[2:0] to 0x4 and write 1 to IMTG. To send back a NAK, set IMTGMOD[2:0] to 0x5 and write 1 to IMTG.

IMBSY is set to 1 while an ACK/NAK is being sent and it reverts to 0 when the transmission has completed. An interrupt request can be generated at this point. When an ACK or NAK has been sent, IMSTA[2:0] is set to 0x4.

Repeat an 8-bit data reception and ACK (NAK) transmission for the required number of times.

(4) Generating stop condition

The procedure is the same as that of data transmission in I²C master mode.

(5) Generating repeated start condition

The procedure is the same as that of data transmission in I²C master mode.

Clock stretch function

During transmitting/receiving data, the slave device may issue a wait request to the master device by pulling down the SCL line to low until the slave device becomes ready to transmit/receive the subsequent data. The master device enters a standby state until the wait request is canceled (the SCL line goes high).

This I²C controller supports this clock stretch function. When a clock stretch condition is detected after a slave address or data has been sent/received, this module enters a waiting status and it does not start operating even if it accepts a trigger for data transfer until the clock stretch status is canceled. IMBSY is maintained at 1 until the triggered operation has completed including a waiting status.

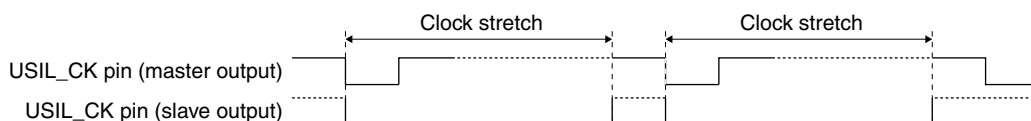


Figure 19.5.3.10 Clock Stretch

Control method in I²C slave mode

Data transfer in I²C slave mode is controlled using ISTGMOD[2:0]/USIL_ISTG register and ISTG/USIL_ISTG register. Select an I²C slave operation using ISTGMOD[2:0] and write 1 to ISTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

Table 19.5.3.3 Trigger List in I²C Slave Mode

ISTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception/stop condition detection
0x2	Data transmission
0x1	Reserved
0x0	Wait for start condition

(Default: 0x0)

Writing 1 to ISTG sets ISBSY/USIL_ISIF register to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, ISBSY is reset to 0. At the same time, the interrupt flag (ISIF/USIL_ISIF register) is also set to 1. After an interrupt occurs, read the status bits (ISSTA[2:0]/USIL_ISIF register) to check the operation finished. Then, clear ISIF by writing 1. This also automatically reset ISSTA[2:0] to 0x0.

Table 19.5.3.4 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

Data transmission in I²C slave mode

The following describes the data transmission procedure in I²C slave mode.

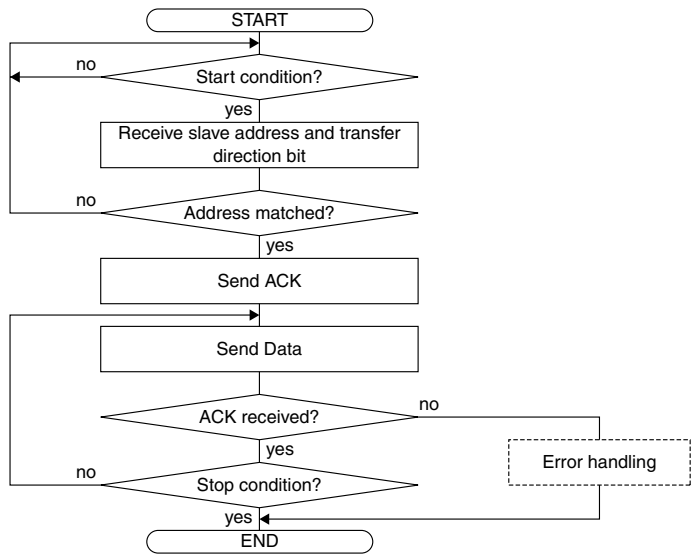
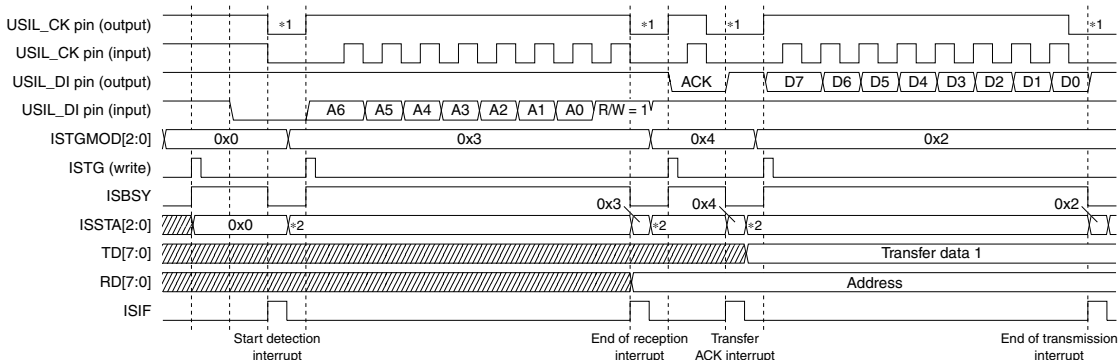
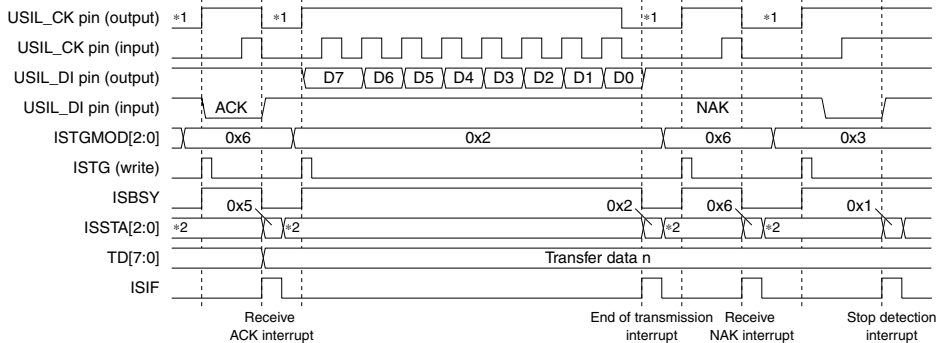


Figure 19.5.3.11 I²C Slave Data Transmission Flow Chart



(1) Start condition → Data transmission



(2) Data transmission → Stop condition

- *1 When the USIL_CK input is detected as low after the operation selected by ISTGMOD[2:0] has finished, the USIL I²C slave controller pulls down the USIL_CK pin to low to places the external I²C master into wait state. This pull-down is canceled to release the I²C master from wait state when the subsequent operation is triggered by ISTG.
- *2 When ISIF is cleared via software, ISSTA[2:0] is also cleared to 0x0.

Figure 19.5.3.12 I²C Slave Data Transmission Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See “Receiving control byte in I²C slave mode” in “19.9 Precautions.”

(1) Waiting for start condition

I²C data transfer starts when the I²C master device generates a start condition (see Figure 19.5.3.3).

First enable this I²C slave to detect a start condition by setting ISTGMOD[2:0] to 0x0 (default) and writing 1 to ISTG. The I²C controller starts detecting a start condition and sets ISBSY to 1. ISBSY is set to 1 while a start condition is being detected. ISBSY reverts to 0 and ISSTA[2:0] is set to 0x0 when the detection has completed. Check if a start condition is generated by reading ISBSY or using an interrupt.

Note: Other operations cannot be started before a start condition is detected.

(2) Receiving slave address and transfer direction data bit

The I²C master sends the address of the slave device to be communicated and a transfer direction bit (see Figure 19.5.3.4) after it has generated a start condition. Set this I²C slave into receiving status to receive the slave address. To start reception, set ISTGMOD[2:0] to 0x3 and write 1 to ISTG.

This trigger starts sampling clocks input from the USIL_CK pin. When clocks are input, the I²C controller loads the USIL_DO pin status to the shift register in sync with each clock. The received data is loaded to the receive data buffer (RD[7:0]/USIL_RD register) once the 8-bit data has been received in the shift register.

Writing 1 to ISTG sets ISBSY to 1. When the received data is loaded to the receive data buffer, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x3 (end of receive data). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

When a 7-bit address is used, the slave address and transfer direction bit can be obtained in one operation. When a 10-bit address is used, save the first data received in the receive data buffer into the memory and perform data reception again to obtain the remaining address bits.

Check whether the received address is matched to this I²C slave address or not. When they are matched, send back an ACK to the I²C master by setting ISTGMOD[2:0] to 0x4 and write 1 to ISTG. ISBSY is set to 1 while an ACK is being sent and it reverts to 0 when the transmission has completed. An interrupt request can be generated at this point. When an ACK has been sent, ISSTA[2:0] is set to 0x4.

If the received address is not for this I²C slave, abort data reception and return to Step (1) to wait the subsequent start condition.

(3) Data transmission

When the transfer direction bit received with the slave address in Step (2) is 1, start data transmission by the following procedure:

1. Write an 8-bit transmit data to the transmit data buffer (TD[7:0]).
2. Set ISTGMOD[2:0] to 0x2 and ISTG to 1.

This trigger transfers the buffer data to the transmit shift register to start transmission. When clocks are input from the USIL_CK pin, the data in the shift register is shifted in sequence with the clock and sent from the USIL_DO pin.

Writing 1 to ISTG sets ISBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x2 (end of transmit data). An interrupt request can be generated at this point. Write subsequent data to the transmit data buffer to start the following transmission using this interrupt.

However, check that the master device has sent back an ACK or NAK (by setting ISTGMOD[2:0] to 0x6 and ISTG to 1) before starting the following 8-bit data transmission.

ISBSY is set to 1 while an ACK/NAK is being detected and it reverts to 0 when the detection has completed. Receiving an ACK sets ISSTA[2:0] to 0x5; receiving a NAK sets it to 0x6. Check ISSTA[2:0] after confirming ISBSY or using an interrupt. When an ACK has been received, perform data transmission. When a NAK has been received, perform the appropriate handling.

(4) When a stop condition is received

If the ISSTA[2:0] value read during data transmission is 0x1, the I²C master device has generated a stop condition (see Figure 19.5.3.6). In this case, abort data transmission.

Data reception in I²C slave mode

The following describes the data receiving procedure in I²C slave mode.

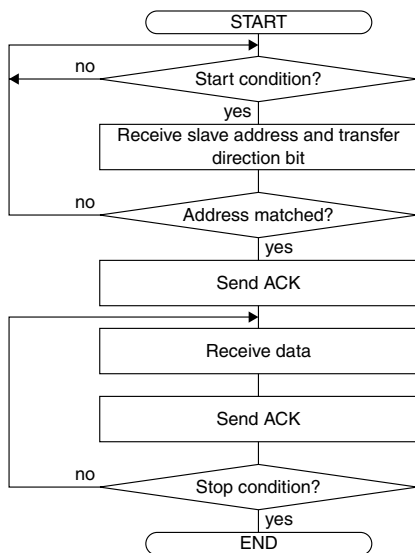
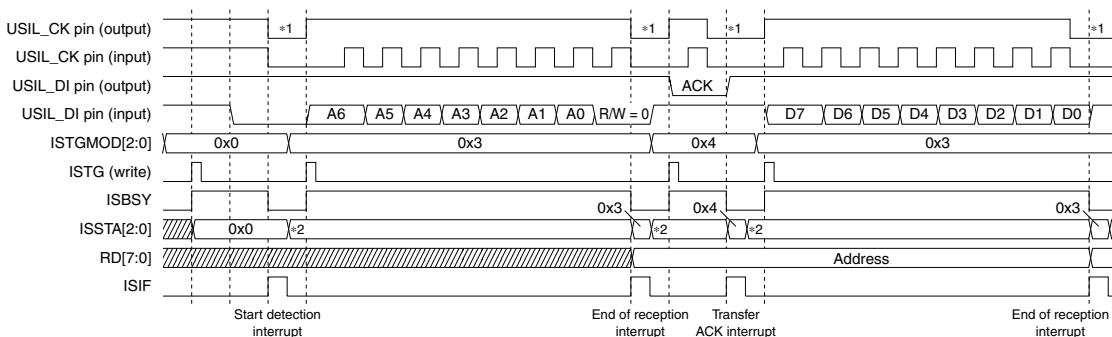
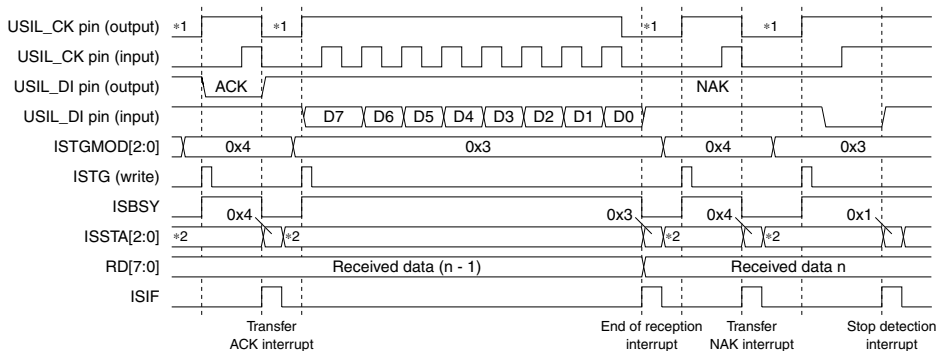


Figure 19.5.3.13 I²C Slave Data Receiving Flow Chart



(1) Start condition → Data reception



(2) Data reception → Stop condition

- *1 When the USIL_CK input is detected as low after the operation selected by ISTGMOD[2:0] has finished, the USIL I²C slave controller pulls down the USIL_CK pin to low to places the external I²C master into wait state. This pull-down is canceled to release the I²C master from wait state when the subsequent operation is triggered by ISTG.
- *2 When ISIF is cleared via software, ISSTA[2:0] is also cleared to 0x0.

Figure 19.5.3.14 I²C Slave Data Receiving Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See “Receiving control byte in I²C slave mode” in “19.9 Precautions.”

(1) Waiting for start condition

The procedure is the same as that of data transmission in I²C slave mode.

(2) Receiving slave address and transfer direction data bit

The procedure is the same as that of data transmission in I²C slave mode.

(3) Data reception

When the transfer direction bit received with the slave address in Step (2) is 0, start data reception by setting ISTGMOD[2:0] to 0x3 and writing 1 to ISTG.

When clocks are input, the I²C controller loads the USIL_DO pin status to the shift register in sync with each clock. The received data is loaded to the receive data buffer (RD[7:0]/USIL_RD register) once the 8-bit data has been received in the shift register.

Writing 1 to ISTG sets ISBSY to 1. When the received data is loaded to the receive data buffer, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x3 (end of receive data). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

It is necessary to send back an ACK or NAK to the master device after an 8-bit data has been received. To send back an ACK, set ISTGMOD[2:0] to 0x4 and write 1 to ISTG. To send back a NAK, set ISTGMOD[2:0] to 0x5 and write 1 to ISTG.

ISBSY is set to 1 while an ACK/NAK is being sent and it reverts to 0 when the transmission has completed. An interrupt request can be generated at this point. When an ACK or NAK has been sent, ISSTA[2:0] is set to 0x4.

Repeat an 8-bit data reception and ACK (NAK) transmission for the required number of times.

(4) When a stop condition is received

If the ISSTA[2:0] value read during data reception is 0x1, the I²C master device has generated a stop condition (see Figure 19.5.3.6). In this case, abort data reception.

Clock stretch function

While data is being sent/received, this I²C slave generates a clock stretch status by pulling down the SCL line to low to make a wait request to the master device after an ACK is sent/received until the following data transfer is started.

19.5.4 Data Transmission in LCD SPI Mode

The LCD SPI mode supports only data transmission.

To start data transmission in LCD SPI mode, write the transmit data to the transmit data buffer (TD[7:0]/USIL_TD register) after setting the command bit status (LSCMD/USIL_LSCFG register).

The buffer data is sent to the transmit shift register. The module starts clock output from the USIL_CK pin. The data in the shift register is shifted in sequence at the clock rising or falling edge (see Figure 19.4.7.1) and sent from the USIL_DO pin.

The LCD SPI controller includes two status flags for transfer control: LSTDIF/USIL_LSIF register and LSBSY/USIL_LSIF register.

The LSTDIF flag indicates the transmit data buffer status. LSTDIF is set to 1 indicating that the transmit data buffer becomes empty when data written to the transmit data buffer is sent to the transmit shift register. LSTDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 19.7). Write subsequent data to the transmit data buffer to start the following transmission using this interrupt or DMA. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. If an interrupt or DMA is not used for transmission, be sure to confirm that the transmit data buffer is empty before writing transmit data. Writing data before LSTDIF has been set will overwrite earlier transmit data inside the transmit data buffer.

The LSBSY flag indicates the USIL status in LCD SPI mode. This flag switches to 1 when transmit data is written to the transmit buffer and reverts to 0 after data transfer for the data size set using LSDMOD[1:0]/USIL_LSDCFG register has completed.

19 UNIVERSAL SERIAL INTERFACE WITH LCD INTERFACE (USIL)

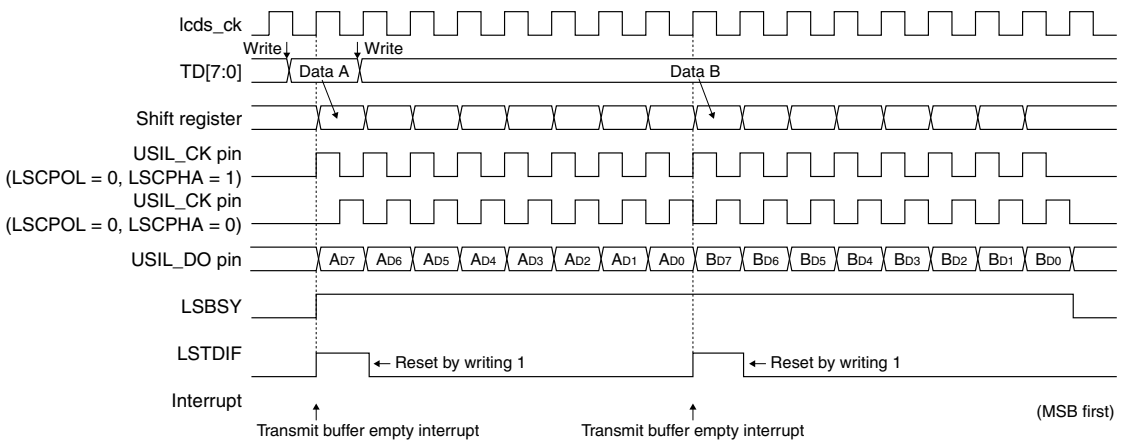


Figure 19.5.4.1 Data Transmission Timing Chart (LCD SPI mode, 16-bit data format)

19.5.5 Data Transfer in LCD Parallel Mode

Data write

To write data to the LCD driver/panel via the LCD parallel interface, write the data to the write (transmit data) buffer (TD[7:0]/USIL_TD register) after setting the command bit status (LPCMD/USIL_LPCFG register).

The command bit must be set before writing data to the write buffer. The command bit value set is output from the USIL_DI pin immediately after it is written to the register.

The LCD parallel interface asserts the chip enable signal and outputs the buffer data via the LCD_D[7:0] pins. The transmitter circuit includes two status flags: LPWRIF/USIL_LPIF register and LPBSY/USIL_LPIF register. The LPWRIF flag indicates the write buffer status. This flag is set to 1 indicating that the write buffer becomes empty when data written to the buffer is output via the LCD_D[7:0] pins. LPWRIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 19.7). Write subsequent data to the write buffer to start the following transmission using this interrupt or DMA. If an interrupt or DMA is not used for transmission, be sure to confirm that the write buffer is empty before writing transmit data. Writing data before LPWRIF has been set will overwrite earlier write data inside the write buffer. After LPWRIF is set to 1, it can be reset to 0 by writing 1.

The LPBSY flag indicates the parallel interface status. This flag switches to 1 when data is written the write buffer and reverts to 0 after the write cycle is completed. Read this flag to check whether the parallel interface circuit is operating or at standby.

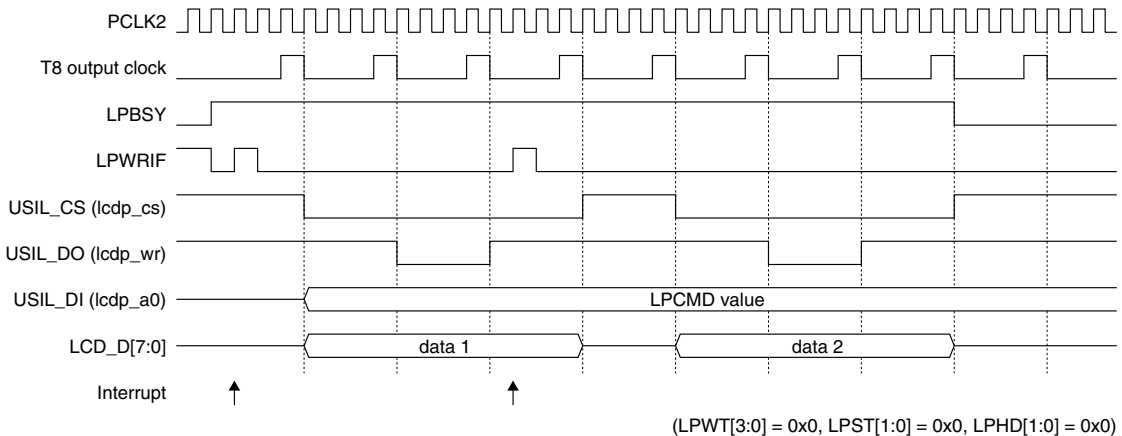


Figure 19.5.5.1 Data-Write Timing Chart (LCD parallel mode)

Data read

To read data from the LCD driver/panel via the LCD parallel interface, issue a read trigger by writing 1 to LPRD/USIL_LPCFG register.

Set the command bit (LPCMD/USIL_LPCFG register) value before writing to LPRD. The command bit value set is output from the USIL_DI pin immediately after it is written to the register. Then it loads the LCD_D[7:0] pin status to the read (receive data) buffer (RD[7:0]/USIL_RD register). LPRD retains 1 until the read data is loaded to the read buffer.

When LCD_D[7:0] is loaded to the read buffer, the read buffer full flag (LPRDIF/USIL_LPIF register) is set to 1 indicating that the buffered data can be read out. LPRDIF is an interrupt flag. An interrupt or DMA request can be generated when this flag is set to 1 (see Section 19.7). Read data from the read buffer using this interrupt or DMA. The read buffer size is 1 byte, therefore the buffered data must be read before starting the subsequent data reading. LPRDIF is reset by writing 1.

The LPBSY flag indicates the parallel interface status. This flag switches to 1 when LPRD is set to 1 and reverts to 0 after the read cycle is completed. Read this flag to check whether the parallel interface circuit is operating or at standby.

Note: Once a triggered read cycle is completed, data is stored in the read buffer and LPRDIF switches to high. After that read data from the read buffer before issuing the next read trigger. Otherwise, a subsequent read operation will not be started.

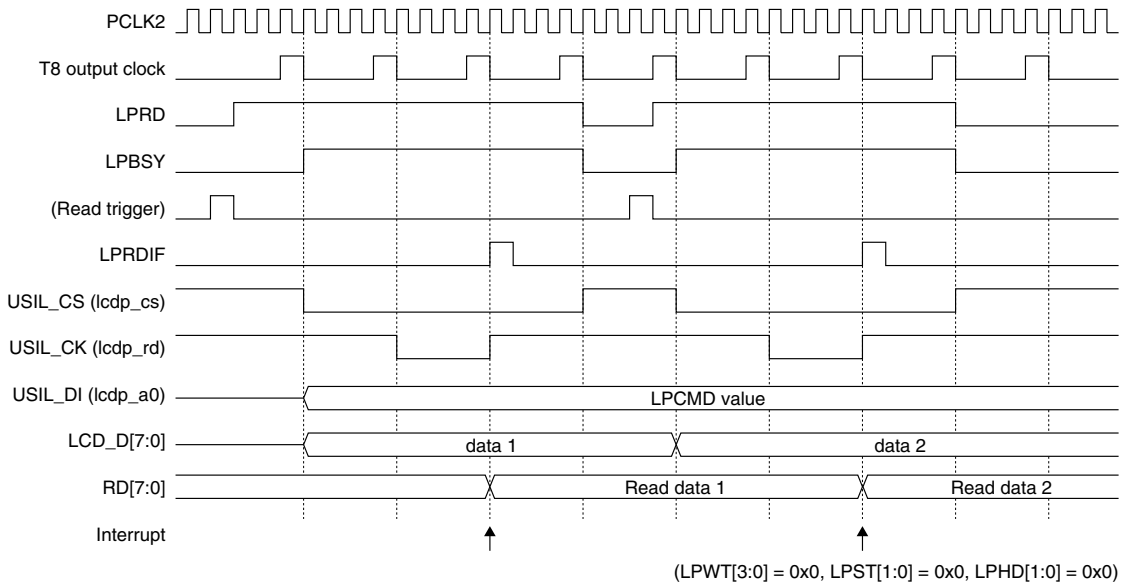


Figure 19.5.5.2 Single Read Timing Chart (LCD parallel mode)

Successive data read

The LCD parallel interface supports a successive data read function. By issuing a read trigger (writing 1 to LPRD) after setting the successive read enable bit (LPSRDEN/USIL_LPCFG register) to 1, the LCD parallel interface repeats data reading from the LCD driver/panel while LPSRDEN is 1 (reading the read buffer issues the next read trigger). When LPSRDEN is set to 0, the LCD parallel interface stops data reading after the read cycle being currently executed has finished. An interrupt or DMA can be generated in each read cycle (when data is loaded to the read buffer) similar to the single read operation, use it to read out the read buffer.

Note: After successive data reading has been terminated, there are 2 or 3 data remained in the read buffer. Be sure to read them as shown in the flow chart (Figure 19.5.5.4).

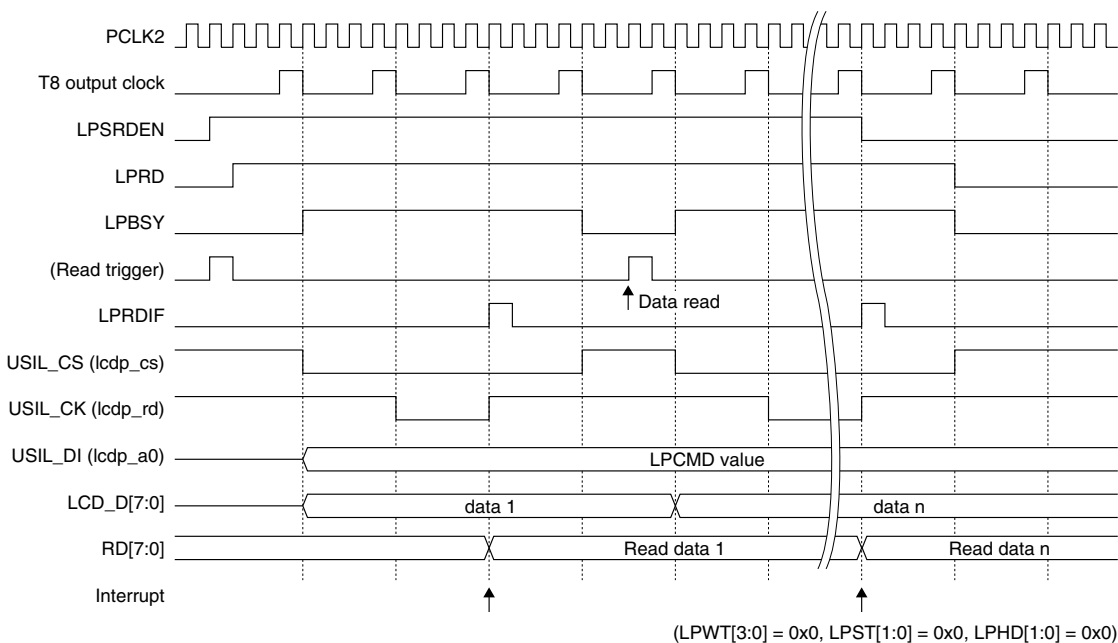


Figure 19.5.5.3 Successive Read Timing Chart (LCD parallel mode)

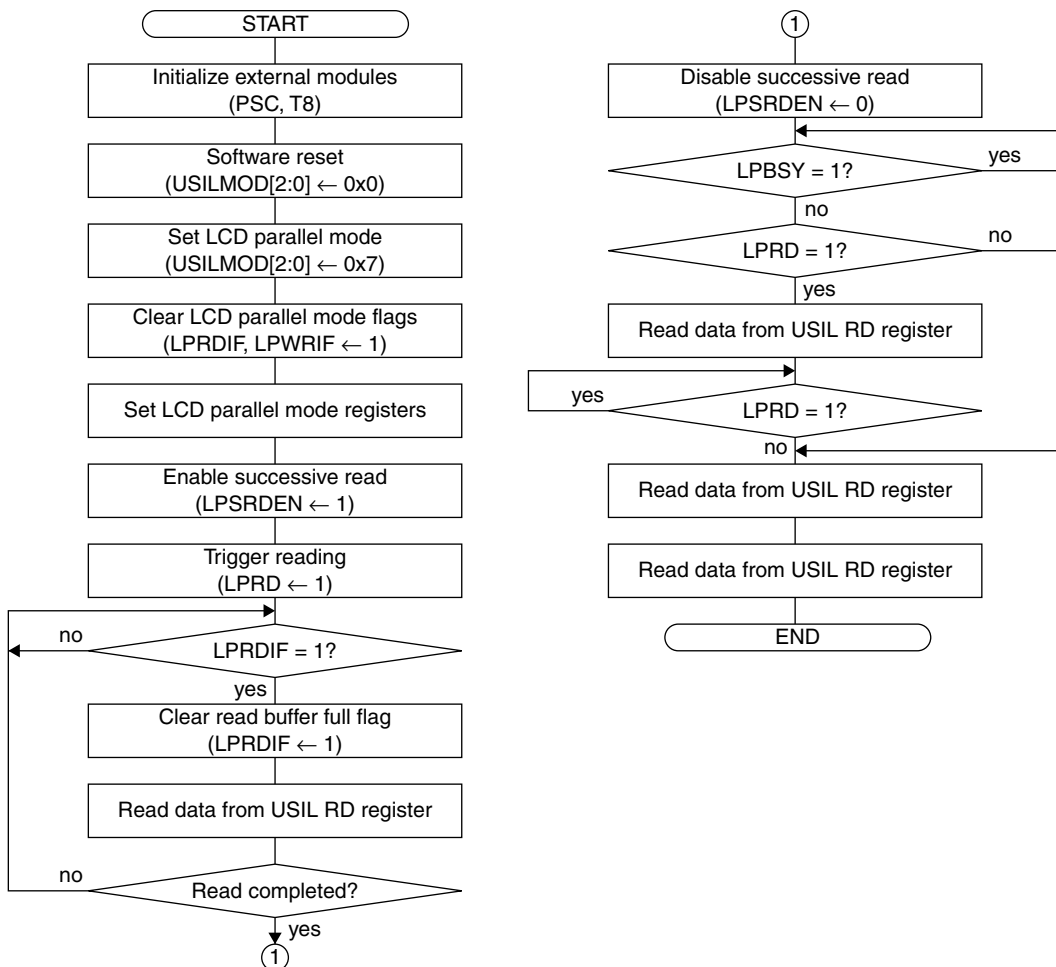


Figure 19.5.5.4 Successive Read Flow Chart (LCD parallel mode)

19.6 Receive Errors

In UART mode, three different receive errors (overrun error, framing error, and parity error) may be detected while receiving data. In SPI and I²C modes, overrun errors may be detected while receiving data.

Since receive errors are interrupt causes, they can be processed by generating interrupts. For more information on interrupt control, see Section 19.7.

Overrun error (UART, SPI, I²C master/slave modes)

UART mode

An overrun error occurs if the next reception is completed when URDIF is 1 and the receive data buffer (USIL_RD register) is not read (an overrun error occurs at the time stop bit has been received).

When an overrun error occurs, the overrun error flag (UOEIF/USIL_UIF register) is set to 1. The receiving operation continues even if this error occurs. To reset UOEIF, perform USIL software reset (write 0x0 to USILMOD[2:0]/USIL_GCFG register) to initialize USIL.

SPI mode

An overrun error occurs if data are received successively when SRDIF is 1. While SRDIF is set to 1, the next received data will not be transferred from the shift register to the receive data buffer (the first byte data exists in the receive data buffer and the second byte data exists in the shift register). An overrun error occurs if the third byte data is received in this condition, as the second byte data in the shift register is corrupted (an overrun error occurs at the time the first bit of the third byte is fetched).

When an overrun error occurs, the overrun error flag (SEIF/USIL_SIF register) is set to 1. The receiving operation continues even if this error occurs. SEIF is reset by writing 1. To reset an overrun error, write 1 to SEIF and then read the receive data buffer (USIL_RD register) twice. The procedure that writes 1 to SEIF and reads USIL_RD register twice can be reversed.

I²C master/slave mode

An overrun error occurs when a transmit or receive trigger is issued after two-byte data has been received (the first byte data exists in the receive data buffer and the second byte data exists in the shift register) without the receive data buffer being read.

When an overrun error occurs, the overrun error flag (IMEIF/USIL_IMIF register for I²C master mode or ISEIF/USIL_ISIF register for I²C slave mode) is set to 1. The receiving operation continues even if this error occurs. IMEIF/ISEIF is reset by writing 1. To reset an overrun error, write 1 to IMEIF/ISEIF and then read the receive data buffer (USIL_RD register) twice.

Framing error (UART mode only)

If the stop bit is received as 0 in UART mode, the UART controller determines loss of sync and a framing error occurs. If the stop bit is configured to two bits, only the first bit is checked.

The framing error flag (USEIF/USIL_UIF register) is set to 1 if this error occurs. The received data is still transferred to the receive data buffer if this error occurs and the receiving operation continues, but the data cannot be guaranteed, even if no framing error occurs for subsequent data receiving. The framing error flag is reset to 0 by writing 1.

Parity error (UART mode only)

If UPREN/USIL_UCFG register has been set to 1 (parity enabled), data received is checked for parity in UART mode. Data received in the shift register is checked for parity when sent to the receive data buffer. The matching is checked against the UPMD/USIL_UCFG register setting (odd or even parity). If the result is a non-match, a parity error is issued, and the parity error flag (UPEIF/USIL_UIF register) is set to 1. Even if this error occurs, the data received is sent to the receive data buffer, and the receiving operation continues. However, the received data cannot be guaranteed if a parity error occurs. The UPEIF flag is reset to 0 by writing 1.

19.7 USIL Interrupts and DMA

This section describes the USIL interrupts generated in each interface mode and invoking DMA.

For more information on interrupt processing and DMA transfer, see the “Interrupt Controller (ITC)” chapter and the “DMA Controller (DMAC)” chapter, respectively.

USIL outputs one interrupt signal shared by the all interrupt causes to the interrupt controller (ITC). Inspect the interrupt flags available in each mode to determine the interrupt cause occurred.

19.7.1 Interrupts in UART Mode

The UART mode includes a function for generating the following three different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

Transmit buffer empty interrupt

To use this interrupt, set UTDIE/USIL_UIE register to 1. If UTDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When transmit data written to the transmit data buffer is transferred to the shift register, the USIL module sets UTDIF/USIL_UIF register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (UTDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the UTDIF flag in the interrupt handler routine to determine whether the USIL (UART mode) interrupt is attributable to a transmit buffer empty. If UTDIF is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set URDIE/USIL_UIE register to 1. If URDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If a received data is loaded into the receive data buffer, the USIL module sets URDIF/USIL_UIF register to 1. If receive buffer full interrupts are enabled (URDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the URDIF flag in the interrupt handler routine to determine whether the USIL (UART mode) interrupt is attributable to a receive buffer full. If URDIF is 1, the received data can be read from the receive data buffer by the interrupt handler routine. However, be sure to check whether a receive error has occurred or not.

Receive error interrupt

To use this interrupt, set UEIE/USIL_UIE register to 1. If UEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

The USIL module sets an error flag (UPEIF/USIL_UIF register, USEIF/USIL_UIF register, or UOEIF/USIL_UIF register) to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are enabled (UEIE = 1), an interrupt request is sent simultaneously to the ITC. If other interrupt conditions are satisfied, an interrupt occurs. You can inspect the UPEIF, USEIF, and UOEIF flags in the interrupt handler routine to determine whether the USIL (UART mode) interrupt was caused by a receive error. If any of the error flags has the value 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, perform USIL software reset (write 0x0 to USILMOD[2:0]/USIL_GCFG register) to initialize USIL.

19.7.2 Interrupts in SPI Mode

The SPI master/slave modes include a function for generating the following three different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

Transmit buffer empty interrupt

To use this interrupt, set `STDIE/USIL_SIE` register to 1. If `STDIE` is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When transmit data written to the transmit data buffer is transferred to the shift register, the USIL module sets `STDIF/USIL_SIF` register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (`STDIE = 1`), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the `STDIF` flag in the interrupt handler routine to determine whether the USIL (SPI master/slave mode) interrupt is attributable to a transmit buffer empty. If `STDIF` is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set `SRDIE/USIL_SIE` register to 1. If `SRDIE` is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If a received data is loaded into the receive data buffer, the USIL module sets `SRDIF/USIL_SIF` register to 1. If receive buffer full interrupts are enabled (`SRDIE = 1`), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the `SRDIF` flag in the interrupt handler routine to determine whether the USIL (SPI master/slave mode) interrupt is attributable to a receive buffer full. If `SRDIF` is 1, the received data can be read from the receive data buffer by the interrupt handler routine. However, be sure to check whether a receive error has occurred or not.

Receive error interrupt

To use this interrupt, set `SEIE/USIL_SIE` register to 1. If `SEIE` is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

The USIL module sets `SEIF/USIL_SIF` register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (`SEIE = 1`), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the `SEIF` flags in the interrupt handler routine to determine whether the USIL (SPI master/slave mode) interrupt was caused by a receive error. If `SEIF` is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear `SEIF` by writing 1 and then read the receive data buffer (`USIL_RD` register) twice.

19.7.3 Interrupts in I²C Master Mode

The I²C master mode includes a function for generating the following two different types of interrupts.

- Operation completion interrupt
- Receive error interrupt

Operation completion interrupt

To use this interrupt, set `IMIE/USIL_IMIE` register to 1. If `IMIE` is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the operation that initiated by a software trigger has completed, the USIL module sets `IMIF/USIL_IMIF` register to 1. If operation completion interrupts are enabled (`IMIE = 1`), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the `IMSTA[2:0]/USIL_IMIF` register in the interrupt handler routine to determine the I²C operation/status that causes the interrupt.

Table 19.7.3.1 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

Receive error interrupt

To use this interrupt, set IMEIE/USIL_IMIE register to 1. If IMEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

An overrun error occurs at the time a transmit or receive trigger is issued after two-byte data has been received without reading the receive data buffer.

The USIL module sets IMEIF/USIL_IMIF register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (IMEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the IMEIF flags in the interrupt handler routine to determine whether the USIL (I²C master mode) interrupt was caused by a receive error. If IMEIF is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear IMEIF by writing 1, and then read the receive data buffer (USIL_RD register) twice.

19.7.4 Interrupts in I²C Slave Mode

The I²C slave mode includes a function for generating the following two different types of interrupts.

- Operation completion interrupt
- Receive error interrupt

Operation completion interrupt

To use this interrupt, set ISIE/USIL_ISIE register to 1. If ISIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the operation that initiated by a software trigger has completed, the USIL module sets ISIF/USIL_ISIF register to 1. If operation completion interrupts are enabled (ISIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the ISSTA[2:0]/USIL_ISIF register in the interrupt handler routine to determine the I²C operation/status that causes the interrupt.

Table 19.7.4.1 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

Receive error interrupt

To use this interrupt, set ISEIE/USIL_ISIE register to 1. If ISEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

An overrun error occurs at the time a transmit or receive trigger is issued after two-byte data has been received without reading the receive data buffer.

The USIL module sets ISEIF/USIL_ISIF register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (ISEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the ISEIF flags in the interrupt handler routine to determine whether the USIL (I²C slave mode) interrupt was caused by a receive error. If ISEIF is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear ISEIF by writing 1, and then read the receive data buffer (USIL_RD register) twice.

19.7.5 Interrupts in LCD SPI Mode

The LCD SPI mode includes a function for generating the following interrupt.

- Transmit buffer empty interrupt

Transmit buffer empty interrupt

To use this interrupt, set LSTDIE/USIL_LSIIE register to 1. If LSTDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When transmit data written to the transmit data buffer is transferred to the shift register, the USIL module sets LSTDIF/USIL_LSIF register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (LSTDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the LSTDIF flag in the interrupt handler routine to determine whether the USIL (LCD SPI mode) interrupt is attributable to a transmit buffer empty. If LSTDIF is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

19.7.6 Interrupts in LCD Parallel Mode

The LCD parallel mode includes a function for generating the following two different types of interrupts.

- Write buffer empty interrupt
- Read buffer full interrupt

Write buffer empty interrupt

To use this interrupt, set LPWRIF/USIL_LPIE register to 1. If LPWRIF is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When data written to the write (transmit data) buffer is output via the LCD_D[7:0] pins, the USIL module sets LPWRIF/USIL_LPIF register to 1, indicating that the write buffer is empty. If write buffer empty interrupts are enabled (LPWRIF = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the LPWRIF flag in the interrupt handler routine to determine whether the USIL (LCD parallel mode) interrupt is attributable to a write buffer empty. If LPWRIF is 1, the next data can be written to the write buffer by the interrupt handler routine.

Read buffer full interrupt

To use this interrupt, set LPRDIF/USIL_LPIE register to 1. If LPRDIF is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If data sent from the LCD driver/panel is loaded into the read (receive data) buffer, the USIL module sets LPRDIF/USIL_LPIF register to 1. If read buffer full interrupts are enabled (LPRDIF = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the LPRDIF flag in the interrupt handler routine to determine whether the USIL (LCD parallel mode) interrupt is attributable to a read buffer full. If LPRDIF is 1, data can be read from the read buffer by the interrupt handler routine.

19.7.7 DMA Transfer

The causes of receive (read) buffer full and transmit (write) buffer empty interrupts in UART, SPI master/slave, LCD SPI, and LCD parallel modes can invoke a DMA. This allows continuous data transmission/reception through DMA transfer between memory and transmit/receive data buffers. These interrupt signals are output to both the ITC and DMAC. Therefore, DMA transfer can be performed without generating any USIL interrupt.

The following lists the DMAC channels that allow selection of a USIL interrupt cause as the trigger.

USIL receive (read) buffer full: DMAC Ch.6

USIL transmit (write) buffer empty: DMAC Ch.7

For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

Note: The USIL module cannot invoke a DMA in I²C master and slave mode.

19.8 Control Register Details

Table 19.8.1 List of USIL Registers

Address	Register name		Function
0x300600	USIL_GCFG	USIL Global Configuration Register	Set interface and MSB/LSB modes
0x300601	USIL_TD	USIL Transmit Data Buffer Register	Transmit data buffer
0x300602	USIL_RD	USIL Receive Data Buffer Register	Receive data buffer
0x300640	USIL_UCFG	USIL UART Mode Configuration Register	Set UART transfer conditions
0x300641	USIL_UIE	USIL UART Mode Interrupt Enable Register	Enable/disable UART interrupts
0x300642	USIL_UIF	USIL UART Mode Interrupt Flag Register	Indicate UART interrupt cause status
0x300650	USIL_SCFG	USIL SPI Master/Slave Mode Configuration Register	Set SPI transfer conditions
0x300651	USIL_SIE	USIL SPI Master/Slave Mode Interrupt Enable Register	Enable/disable SPI interrupts
0x300652	USIL_SIF	USIL SPI Master/Slave Mode Interrupt Flag Register	Indicate SPI interrupt cause status
0x300660	USIL_IMTG	USIL I ² C Master Mode Trigger Register	Start I ² C master operations
0x300661	USIL_IMIE	USIL I ² C Master Mode Interrupt Enable Register	Enable/disable I ² C master interrupts
0x300662	USIL_IMIF	USIL I ² C Master Mode Interrupt Flag Register	Indicate I ² C master interrupt cause status
0x300670	USIL_ISTG	USIL I ² C Slave Mode Trigger Register	Start I ² C slave operations
0x300671	USIL_ISIE	USIL I ² C Slave Mode Interrupt Enable Register	Enable/disable I ² C slave interrupts
0x300672	USIL_ISIF	USIL I ² C Slave Mode Interrupt Flag Register	Indicate I ² C slave interrupt cause status
0x300680	USIL_LSCFG	USIL LCD SPI Mode Configuration Register	Set LCD SPI transfer conditions
0x300681	USIL_LSIE	USIL LCD SPI Mode Interrupt Enable Register	Enable/disable LCD SPI interrupts
0x300682	USIL_LSIF	USIL LCD SPI Mode Interrupt Flag Register	Indicate LCD SPI interrupt cause status
0x30068f	USIL_LSDCFG	USIL LCD SPI Mode Data Configuration Register	Select display data format
0x300690	USIL_LPCFG	USIL LCD Parallel I/F Mode Configuration Register	Set LCD parallel interface conditions
0x300691	USIL_LPIE	USIL LCD Parallel I/F Mode Interrupt Enable Register	Enable/disable LCD parallel interface interrupts
0x300692	USIL_LPIF	USIL LCD Parallel I/F Mode Interrupt Flag Register	Indicate LCD parallel interface interrupt cause status
0x30069f	USIL_LPAC	USIL LCD Parallel I/F Mode Access Timing Register	Set LCD parallel interface access timing parameters

The USIL registers are described in detail below. These are 8-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

USIL Global Configuration Register (USIL_GCFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL Global Configuration Register (USIL_GCFG)	0x300600 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3	LSBFST	MSB/LSB first mode select	1 MSB first	0 LSB first	0	R/W	
		D2–0	USILMOD	Interface mode configuration	USILMOD[2:0]		0x0	R/W	
					0x7	LCD Parallel			
					0x6	LCD SPI			
					0x5	I ² C slave			
					0x4	I ² C master			
					0x3	SPI slave			
					0x2	SPI master			
					0x1	UART			
			0x0	Software reset					

Note: This register must be configured before setting other USIL registers.

D[7:4] Reserved

D3 LSBFST: MSB/LSB First Mode Select Bit

Selects whether serial data will be transferred from the MSB or LSB.

1 (R/W): MSB first

0 (R/W): LSB first (default)

This setting affects all interface modes.

D[2:0] USILMOD[2:0]: Interface Mode Configuration Bits

Selects an interface mode.

Table 19.8.2 Interface Mode Selection

USILMOD[2:0]	Interface mode
0x7	LCD parallel
0x6	LCD SPI
0x5	I ² C slave
0x4	I ² C master
0x3	SPI slave
0x2	SPI master
0x1	UART
0x0	Software reset

(Default: 0x0)

Perform software reset (set USILMOD[2:0] to 0x0) and then set the interface mode before changing other USIL configurations.

USIL Transmit Data Buffer Register (USIL_TD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL Transmit Data Buffer Register (USIL_TD)	0x300601 (8 bits)	D7-0	TD[7:0]	USIL transmit data buffer TD7 = MSB TD0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] TD[7:0]: USIL Transmit Data Buffer Bits

Sets transmit data to be written to the transmit data buffer. (Default: 0x0)

In UART, SPI master, LCD SPI, and LCD parallel modes, transmission begins immediately after writing data to this register. In SPI slave mode, transmission will begin when the clock is input from the SPI master device.

In I²C master/slave mode, transmission begins by the software trigger for data transmission.

The data written to this register is converted into serial data through the shift register and is output from the USIL_DO pin with the bit set to 1 as high level and the bit set to 0 as low level.

In LCD parallel mode, the data written to this register is output via the LCD_D[7:0] pins.

A transmit (write) buffer empty interrupt can be generated when data written to this register has been transferred to the shift register or output from the LCD_D[7:0] pins. The subsequent transmit data can then be written, even while data is being sent.

USIL Receive Data Buffer Register (USIL_RD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL Receive Data Buffer Register (USIL_RD)	0x300602 (8 bits)	D7-0	RD[7:0]	USIL receive data buffer RD7 = MSB RD0 = LSB	0x0 to 0xff	0x0	R	

D[7:0] RD[7:0]: USIL Receive Data Buffer Bits

Contains the received data. (Default: 0x0)

Serial data input from the USIL_DI pin is converted to parallel, with the high level bit set to 1 and the low level bit set to 0, and then it is loaded to this register. In LCD parallel mode, data input from the LCD_D[7:0] pins are loaded to this register.

A receive (read) buffer full interrupt can be generated when the data received has been loaded to this register. Data can then be read until subsequent data is received.

This register is read-only.

USIL UART Mode Configuration Register (USIL_UCFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL UART Mode Configuration Register (USIL_UCFG)	0x300640 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	UCLN	Character length select	1 8 bits 0 7 bits	0	R/W	
		D2	USTPB	Stop bit select	1 2 bits 0 1 bit	0	R/W	
		D1	UPMD	Parity mode select	1 Even 0 Odd	0	R/W	
		D0	UPREN	Parity enable	1 With parity 0 No parity	0	R/W	

Note: This register is effective only in UART mode. Configure USIL to UART mode before setting this register.

D[7:4] Reserved

D3 UCHLN: Character Length Select Bit

Selects the serial transfer data length.

1 (R/W): 8 bits

0 (R/W): 7 bits (default)

When 7-bit data length is selected, D7 in the transmit data buffer is ignored and D7 in the receive data buffer is always set to 0.

D2 USTPB: Stop Bit Select Bit

Selects the stop bit length.

1 (R/W): 2 bits

0 (R/W): 1 bit (default)

Writing 1 to USTPB selects 2 stop bits; writing 0 to it selects 1 bit. The start bit is fixed at 1 bit.

D1 UPMD: Parity Mode Select Bit

Selects the parity mode.

1 (R/W): Even parity

0 (R/W): Odd parity (default)

Parity checking and parity bit addition are enabled only when UPREN is set to 1. The UPMD setting is disabled if UPREN is 0.

D0 UPREN: Parity Enable Bit

Enables the parity function.

1 (R/W): With parity

0 (R/W): No parity (default)

UPREN is used to select whether received data parity checking is performed and whether a parity bit is added to transmit data. Setting UPREN to 1 parity-checks the received data. A parity bit is automatically added to the transmit data. If UPREN is set to 0, no parity bit is checked or added.

USIL UART Mode Interrupt Enable Register (USIL_UIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL UART Mode Interrupt Enable Register (USIL_UIE)	0x300641 (8 bits)	D7-3	-	reserved	-	-	-	0 when being read.
		D2	UEIE	Receive error interrupt enable	1 Enable	0 Disable	0 R/W	
		D1	URDIE	Receive buffer full interrupt enable	1 Enable	0 Disable	0 R/W	
		D0	UTDIE	Transmit buffer empty int. enable	1 Enable	0 Disable	0 R/W	

Note: This register is effective only in UART mode. Configure USIL to UART mode before this register can be used.

D[7:3] Reserved

D2 UEIE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when a receive error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process receive errors using interrupts.

D1 URDIE: Receive Buffer Full Interrupt Enable Bit

Enables interrupt requests to the ITC when received data is loaded to the receive data buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

D0 UTDIE: Transmit Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC when data written to the transmit data buffer is sent to the shift register (i.e. when data transmission begins).

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to write data to the transmit data buffer using interrupts.

USIL UART Mode Interrupt Flag Register (USIL_UIF)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
USIL UART Mode Interrupt Flag Register (USIL_UIF)	0x300642 (8 bits)	D7	–	reserved	–		–	–	0 when being read.	
		D6	URBSY	Receive busy flag	1	Busy	0	Idle	0	R
		D5	UTBSY	Transmit busy flag	1	Busy	0	Idle	0	R
		D4	UPEIF	Parity error flag	1	Error	0	Normal	0	R/W
		D3	USEIF	Framing error flag	1	Error	0	Normal	0	R/W
		D2	UOEIF	Overrun error flag	1	Error	0	Normal	0	R/W
		D1	URDIF	Receive buffer full flag	1	Full	0	Not full	0	R/W
		D0	UTDIF	Transmit buffer empty flag	1	Empty	0	Not empty	0	R/W

Note: This register is effective only in UART mode. Configure USIL to UART mode before this register can be used.

D7 Reserved**D6 URBSY: Receive Busy Flag Bit**

Indicates the receive shift register status.

1 (R): Busy

0 (R): Idle (default)

URBSY is set to 1 when the first start bit is detected (when data reception begins) and is reset to 0 when the data received in the shift register is loaded into the receive data buffer. Inspect URBSY to determine whether the receiving circuit is operating or at standby.

D5 UTBSY: Transmit Busy Flag Bit

Indicates the USIL status in UART mode.

1 (R): Busy

0 (R): Idle (default)

UTBSY switches to 1 when transmit data is written to the transmit buffer and reverts to 0 after both the shift register and transmit buffer become empty.

D4 UPEIF: Parity Error Flag Bit

Indicates whether a parity error has occurred or not.

1 (R): Error occurred

0 (R): No error (default)

1 (W): Reset to 0

0 (W): Ignored

UPEIF is set to 1 when a parity error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USIL_UIE register is 1. Parity checking is enabled only when UPREN/USIL_UCFG register is set to 1 and is performed when received data is transferred from the shift register to the receive data buffer. UPEIF is reset by writing 1.

D3 USEIF: Framing Error Flag Bit

Indicates whether a framing error has occurred or not.

1 (R): Error occurred

0 (R): No error (default)

1 (W): Reset to 0

0 (W): Ignored

USEIF is set to 1 when a framing error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USIL_UIE register is 1. A framing error occurs when data is received with the stop bit set to 0. USEIF is reset by writing 1.

D2 UOEIF: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

- 1 (R): Error occurred
- 0 (R): No error (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

UOEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USIL_UIE register is 1. An overrun error occurs if the next reception is completed when URDIF is 1 and the receive data buffer (USIL_RD register) is not read (an overrun error occurs at the time stop bit has been received). To reset UOEIF, perform USIL software reset (write 0x0 to USILMOD[2:0]/USIL_GCFG register) to initialize USIL.

D1 URDIF: Receive Buffer Full Flag Bit

Indicates the receive data buffer status.

- 1 (R): Data full
- 0 (R): No data (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

URDIF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is completed), indicating that the data can be read. At the same time a receive buffer full interrupt request is sent to the ITC if URDIE/USIL_UIE register is 1. URDIF is reset by writing 1.

D0 UTDIF: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

- 1 (R): Empty
- 0 (R): Data exists (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

UTDIF is set to 1 when the transmit data written to the transmit data buffer is transferred to the shift register (when transmission starts), indicating that the next transmit data can be written to. At the same time a transmit buffer empty interrupt request is sent to the ITC if UTDIE/USIL_UIE register is 1. UTDIF is reset by writing 1.

USIL SPI Master/Slave Mode Configuration Register (USIL_SCFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL SPI Master/Slave Mode Configu- ration Register (USIL_SCFG)	0x300650 (8 bits)	D7-4	-	reserved		-	-	0 when being read.
		D3	SCPHA	Clock phase select	1 Phase 1 0 Phase 0	0	R/W	
		D2	SCPOL	Clock polarity select	1 Active L 0 Active H	0	R/W	
		D1	-	reserved		-	-	Do not set to 1.
		D0	SFSTMOD	Fast mode select	1 Fast 0 Normal	0	R/W	

Note: This register is effective only in SPI master and slave modes. Configure USIL to SPI master/slave mode before this register can be used.

D[7:4] Reserved**D3 SCPHA: Clock Phase Select Bit**

Selects the SPI clock phase.

- 1 (R/W): Phase 1
- 0 (R/W): Phase 0 (default)

Set the data transfer timing together with SCPOL. (See Figure 19.8.1.)

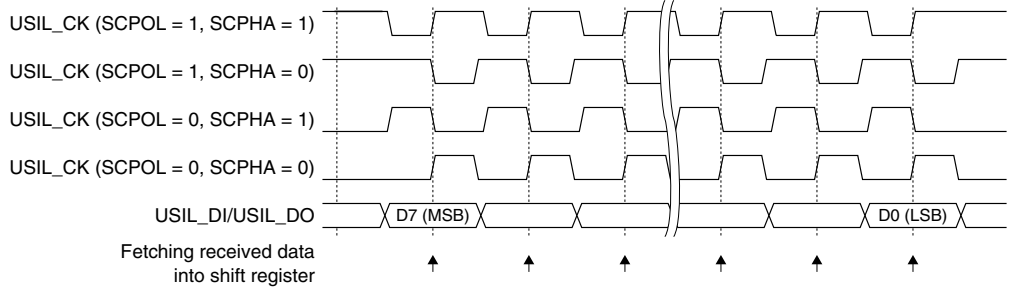
D2 SCPOL: Clock Polarity Select Bit

Selects the SPI clock polarity.

- 1 (R/W): Active low
- 0 (R/W): Active high (default)

Set the data transfer timing together with SCPHA. (See Figure 19.8.1.)

Master mode



Slave mode

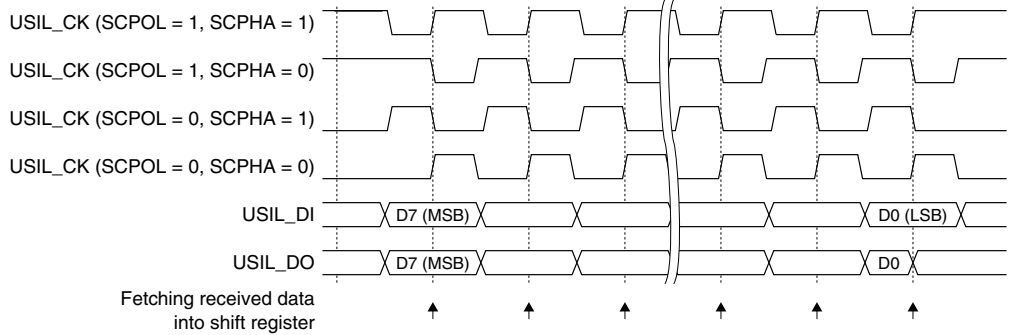


Figure 19.8.1 Clock and Data Transfer Timing (MSB first)

D1 Reserved (Do not set to 1.)

D0 **SFSTMOD: Fast Mode Select Bit (for SPI master mode)**

Selects Fast mode.

1 (R/W): Fast mode

0 (R/W): Normal mode (default)

In SPI master mode, either normal or fast clock mode can be selected using SFSTMOD. Setting SFSTMOD to 0 (default) places the USIL into normal mode and the USIL generates the transfer clock by dividing the T8 output by 2. Setting SFSTMOD to 1 places the USIL into fast mode and the USIL uses PCLK2 supplied from the CMU directly as the transfer clock. The fast mode does not use the T8.

The SPI slave mode uses the T8 output clock for generating the sampling clock.

USIL SPI Master/Slave Mode Interrupt Enable Register (USIL_SIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL SPI Master/Slave Mode Interrupt Enable Register (USIL_SIE)	0x300651 (8 bits)	D7-3	-	reserved	-	-	-	0 when being read.
		D2	SEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SRDIE	Receive buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	STDIE	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in SPI master and slave modes. Configure USIL to SPI master/slave mode before this register can be used.

D[7:3] Reserved

D2 **SEIE: Receive Error Interrupt Enable Bit**

Enables interrupt requests to the ITC when an overrun error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D1 SRDIE: Receive Buffer Full Interrupt Enable Bit

Enables interrupt requests to the ITC when received data is loaded to the receive data buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

D0 STDIE: Transmit Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC when data written to the transmit data buffer is sent to the shift register (i.e. when data transmission begins).

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to write data to the transmit data buffer using interrupts.

USIL SPI Master/Slave Mode Interrupt Flag Register (USIL_SIF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL SPI Master/Slave Mode Interrupt Flag Register (USIL_SIF)	0x300652 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.	
		D3	SSIF	Transfer busy flag (master) ss signal low flag (slave)	1 Busy 1 ss = H	0 Idle 0 ss = L	0	R	
		D2	SEIF	Overrun error flag	1 Error	0 Normal	0	R/W	Reset by writing 1.
		D1	SRDIF	Receive buffer full flag	1 Full	0 Not full	0	R/W	
		D0	STDIF	Transmit buffer empty flag	1 Empty	0 Not empty	0	R/W	

Note: This register is effective only in SPI master and slave modes. Configure USIL to SPI master/slave mode before this register can be used.

D[7:4] Reserved**D3 SSIF: Transfer Busy Flag Bit (Master Mode)/ss Signal Low Flag Bit (Slave Mode)****Master mode**

Indicates the SPI transfer status.

1 (R): Operating

0 (R): Standby (default)

SSIF is set to 1 when the SPI starts data transfer in master mode and is maintained at 1 while transfer is underway. It is cleared to 0 once the transfer is completed.

Slave mode

Indicates the slave select (USIL_CS) signal status.

1 (R): High level (this SPI is not selected)

0 (R): Low level (this SPI is selected) (default)

SSIF is set to 0 when the master device asserts the slave select (USIL_CS) signal to select this SPI controller (slave device). It is returned to 1 when the master device clears the SPI controller selection by negating the slave select (USIL_CS) signal.

D2 SEIF: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): Error occurred

0 (R): No error (default)

1 (W): Reset to 0

0 (W): Ignored

SEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if SEIE/USIL_SIE register is 1. An overrun error occurs if data are received successively when SRDIF is 1. While SRDIF is set to 1, the next received data will not be transferred from the shift register to the receive data buffer (the first byte data exists in the receive data buffer and the second byte data exists in the shift register). An overrun error occurs if the third byte data is received in this condition, as the second byte data in the shift register is corrupted (an overrun error occurs at the time the first bit of the third byte is fetched).

SEIF is reset by writing 1. To reset an overrun error, write 1 to SEIF and then read the receive data buffer (USIL_RD register) twice. The procedure that writes 1 to SEIF and reads USIL_RD register twice can be reversed.

D1 SRDIF: Receive Buffer Full Flag Bit

Indicates the receive data buffer status.

- 1 (R): Data full
- 0 (R): No data (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

SRDIF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is completed), indicating that the data can be read. At the same time a receive buffer full interrupt request is sent to the ITC if SRDIE/USIL_SIE register is 1. SRDIF is reset by writing 1.

D0 STDIF: Transmit Buffer Empty Flag Bit

Indicates the transmit data buffer status.

- 1 (R): Empty
- 0 (R): Data exists (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

STDIF is set to 1 when the transmit data written to the transmit data buffer is transferred to the shift register (when transmission starts), indicating that the next transmit data can be written to. At the same time a transmit buffer empty interrupt request is sent to the ITC if STDIE/USIL_SIE register is 1. STDIF is reset by writing 1.

USIL I²C Master Mode Trigger Register (USIL_IMTG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL I ² C Master Mode Trigger Register (USIL_IMTG)	0x300660 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4	IMTG	I ² C master operation trigger	1 Trigger 1 Waiting	0 Ignored 0 Finished	0	W R	
		D3	–	reserved	–	–	–	–	0 when being read.
		D2-0	IMTGMOD [2:0]	I ² C master trigger mode select	IMTGMOD[2:0]	Trigger mode	0x0	R/W	
						0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 Stop condition 0x0 Start condition			

Note: This register is effective only in I²C master mode. Configure USIL to I²C master mode before this register can be used.

D[7:5] Reserved

D4 IMTG: I²C Master Operation Trigger Bit

Starts an I²C master operation.

- 1 (W): Trigger
- 0 (W): Ignored
- 1 (R): Waiting for starting operation
- 0 (R): Trigger has finished (default)

Select an I²C master operation using IMTGMOD[2:0] and write 1 to IMTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

D3 Reserved

D[2:0] IMTGMOD[2:0]: I²C Master Trigger Mode Select Bits

Selects an I²C master operation.

Table 19.8.3 Trigger List in I²C Master Mode

IMTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception
0x2	Data transmission
0x1	Stop condition
0x0	Start condition

(Default: 0x0)

USIL I²C Master Mode Interrupt Enable Register (USIL_IMIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL I ² C Master Mode Interrupt Enable Register (USIL_IMIE)	0x300661 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	IMEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	IMIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in I²C master mode. Configure USIL to I²C master mode before this register can be used.

D[7:2] Reserved

D1 IMEIE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when an overrun error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D0 IMIE: Operation Completion Interrupt Enable Bit

Enables interrupt requests to the ITC when the triggered operation has completed.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to confirm whether the triggered operation has completed or not using interrupts.

USIL I²C Master Mode Interrupt Flag Register (USIL_IMIF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL I ² C Master Mode Interrupt Flag Register (USIL_IMIF)	0x300662 (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	IMBSY	I ² C master busy flag	1 Busy 0 Standby	0	R	
		D4-2	IMSTA[2:0]	I ² C master status	IMSTA[2:0] Status	0x0	R	
					0x7	reserved		
					0x6	NAK received		
					0x5	ACK received		
					0x4	ACK/NAK sent		
0x3	End of Rx data							
0x2	End of Tx data							
0x1	Stop generated							
0x0	Start generated							
D1	IMEIF	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.		
D0	IMIF	Operation completion flag	1 Completed 0 Not completed	0	R/W			

Note: This register is effective only in I²C master mode. Configure USIL to I²C master mode before this register can be used.

D[7:6] Reserved

D5 IMBSY: I²C Master Busy Flag Bit

Indicates the I²C master operation status.

1 (R): Busy

0 (R): Standby (default)

Writing 1 to IMTG/USIL_IMTG register (starting an I²C master operation) sets IMBSY to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, IMBSY is reset to 0.

D[4:2] IMSTA[2:0]: I²C Master Status Bits

Indicates the I²C master status.

Table 19.8.4 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

When an operation completion interrupt occurs, read IMSTA[2:0] to check the operation that has been finished. IMSTA[2:0] is automatically reset to 0x0 by writing 1 to IMIF.

D1 IMEIF: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): Error occurred

0 (R): No error (default)

1 (W): Reset to 0

0 (W): Ignored

IMEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if IMEIE/USIL_IMIE register is 1.

An overrun error occurs when a transmit or receive trigger is issued after two-byte data has been received (the first byte data exists in the receive data buffer and the second byte data exists in the shift register) without the receive data buffer being read.

IMEIF is reset by writing 1.

To reset an overrun error, clear IMEIF by writing 1, and then read the receive data buffer (USIL_RD register) twice.

D0 IMIF: Operation Completion Flag Bit

Indicates whether the triggered operation has completed or not.

1 (R): Completed

0 (R): Not completed (default)

1 (W): Reset to 0

0 (W): Ignored

IMIF is set to 1 when the operation that is specified and triggered using the USIL_IMTG register has completed. At the same time an operation completion interrupt request is sent to the ITC if IMIE/USIL_IMIE register is 1. IMIF is reset by writing 1.

USIL I²C Slave Mode Trigger Register (USIL_ISTG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL I ² C Slave Mode Trigger Register (USIL_ISTG)	0x300670 (8 bits)	D7-5	—	reserved	—	—	—	0 when being read.	
		D4	ISTG	I ² C slave operation trigger	1 Trigger 0 Ignored	0	W		
						1 Waiting 0 Finished		R	
		D3	—	reserved	—	—	—	—	0 when being read.
		D2-0	ISTGMOD[2:0]	I ² C slave trigger mode select	ISTGMOD[2:0]	Trigger mode	0x0	R/W	
				0x7 reserved					
				0x6 Receive ACK/NAK					
				0x5 Transmit NAK					
				0x4 Transmit ACK					
				0x3 Receive data/ Detect stop					
				0x2 Transmit data					
				0x1 reserved					
				0x0 Wait for start					

Note: This register is effective only in I²C slave mode. Configure USIL to I²C slave mode before this register can be used.

D[7:5] Reserved

D4 ISTG: I²C Slave Operation Trigger Bit

Starts an I²C slave operation.

1 (W): Trigger

0 (W): Ignored

1 (R): Waiting for starting operation

0 (R): Trigger has finished (default)

Select an I²C slave operation using ISTGMOD[2:0] and write 1 to ISTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

D3 Reserved

D[2:0] ISTGMOD[2:0]: I²C Slave Trigger Mode Select Bits

Selects an I²C slave operation.

Table 19.8.5 Trigger List in I²C Slave Mode

ISTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception/stop condition detection
0x2	Data transmission
0x1	Reserved
0x0	Wait for start condition

(Default: 0x0)

USIL I²C Slave Mode Interrupt Enable Register (USIL_ISIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL I ² C Slave Mode Interrupt Enable Register (USIL_ISIE)	0x300671 (8 bits)	D7-2	-	reserved	-	-	-	0 when being read.
		D1	ISEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	ISIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in I²C slave mode. Configure USIL to I²C slave mode before this register can be used.

D[7:2] Reserved

D1 ISEIE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when an overrun error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D0 ISIE: Operation Completion Interrupt Enable Bit

Enables interrupt requests to the ITC when the triggered operation has completed.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to confirm whether the triggered operation has completed or not using interrupts.

USIL I²C Slave Mode Interrupt Flag Register (USIL_ISIF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL I ² C Slave Mode Interrupt Flag Register (USIL_ISIF)	0x300672 (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.	
		D5	ISBSY	I ² C slave busy flag	1 Busy 0 Standby	0	R		
		D4-2	ISSTA[2:0]	I ² C slave status	ISSTA[2:0]	Status	0x0	R	
					0x7	reserved			
					0x6	NAK received			
					0x5	ACK received			
0x4	ACK/NAK sent								
0x3	End of Rx data								
0x2	End of Tx data								
0x1	Stop detected								
0x0	Start detected								
D1	ISEIF	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.			
D0	ISIF	Operation completion flag	1 Completed 0 Not completed	0	R/W				

Note: This register is effective only in I²C slave mode. Configure USIL to I²C slave mode before this register can be used.

D[7:6] Reserved**D5 ISBSY: I²C Slave Busy Flag Bit**

Indicates the I²C slave operation status.

1 (R): Busy

0 (R): Standby (default)

Writing 1 to ISTG/USIL_ISTG register (starting an I²C slave operation) sets ISBSY to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, ISBSY is reset to 0.

D[4:2] ISSTA[2:0]: I²C Slave Status Bits

Indicates the I²C slave status.

Table 19.8.6 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been sent.
0x3	End of receive data.
0x2	End of transmit data.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

When an operation completion interrupt occurs, read ISSTA[2:0] to check the operation that has been finished. ISSTA[2:0] is automatically reset to 0x0 by writing 1 to ISIF.

D1 ISEIF: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): Error occurred

0 (R): No error (default)

1 (W): Reset to 0

0 (W): Ignored

ISEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if ISEIE/USIL_ISIE register is 1. An overrun error occurs when a transmit or receive trigger is issued after two-byte data has been received (the first byte data exists in the receive data buffer and the second byte data exists in the shift register) without the receive data buffer being read.

ISEIF is reset by writing 1.

To reset an overrun error, clear ISEIF by writing 1, and then read the receive data buffer (USIL_RD register) twice.

D0 ISIF: Operation Completion Flag Bit

Indicates whether the triggered operation has completed or not.

- 1 (R): Completed
- 0 (R): Not completed (default)
- 1 (W): Reset to 0
- 0 (W): Ignored

ISIF is set to 1 when the operation that is specified and triggered using the USIL_ISTG register has completed. At the same time an operation completion interrupt request is sent to the ITC if ISIE/USIL_ISIE register is 1. ISIF is reset by writing 1.

USIL LCD SPI Mode Configuration Register (USIL_LSCFG)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
USIL LCD SPI Mode Configuration Register (USIL_LSCFG)	0x300680 (8 bits)	D7-4	-	reserved				-	-	0 when being read.	
		D3	LSCPHA	Clock phase select	1	Phase 1	0	Phase 0	0		R/W
		D2	LSCPOL	Clock polarity select	1	Active L	0	Active H	0		R/W
		D1	LSCMD	Command bit	1	High	0	Low	0		R/W
		D0	LSCMDEN	Command bit enable	1	Enable	0	Disable	0		R/W

Note: This register is effective only in LCD SPI mode. Configure USIL to LCD SPI mode before setting this register.

D[7:4] Reserved

D3 LSCPHA: Clock Phase Select Bit

Selects the LCD SPI clock phase.

- 1 (R/W): Phase 1
- 0 (R/W): Phase 0 (default)

Set the data transfer timing together with LSCPOL. (See Figure 19.8.2.)

D2 LSCPOL: Clock Polarity Select Bit

Selects the LCD SPI clock polarity.

- 1 (R/W): Active low
- 0 (R/W): Active high (default)

Set the data transfer timing together with LSCPHA. (See Figure 19.8.2.)

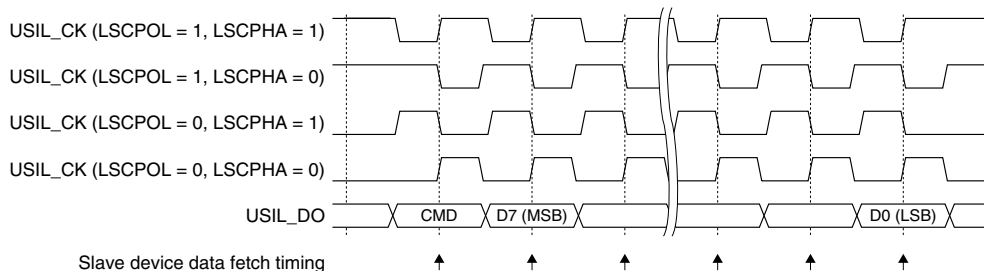


Figure 19.8.2 Clock and Data Transfer Timing

D1 LSCMD: Command Bit

Sets the command bit value (see LSCMDEN below).

- 1 (R/W): High
- 0 (R/W): Low (default)

D0 LSCMDEN: Command Bit Enable Bit

Enables sending the command bit on the data line (USIL_DO).

- 1 (R/W): Enabled
- 0 (R/W): Disabled (default)

When LSCMDEN is set to 1, data is prefixed with a command bit (1 bit). The command bit is used for controlling the SPI LCD driver/panel connected to the USIL. The command bit value to be transmitted can be specified using LSCMD.

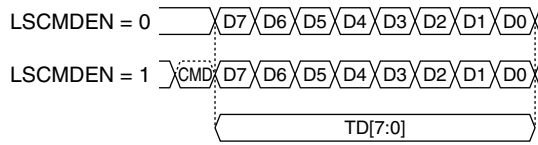


Figure 19.8.3 Data Configuration (8-bit data mode)

When LSCMDEN is set to 0, the command bit selected using LSCMD is output from the USIL_DI (lcds_a0) pin.

USIL LCD SPI Mode Interrupt Enable Register (USIL_LSIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL LCD SPI Mode Interrupt Enable Register (USIL_LSIE)	0x300681 (8 bits)	D7-1	–	reserved	–		–	–	0 when being read.
		D0	LSTDIE	Transmit buffer empty int. enable	1 Enable	0 Disable	0	R/W	

Note: This register is effective only in LCD SPI mode. Configure USIL to LCD SPI mode before setting this register.

D[7:1] Reserved

D0 LSTDIE: Transmit Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC when data written to the transmit data buffer is sent to the shift register (i.e. when data transmission begins).

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to write data to the transmit data buffer using interrupts.

USIL LCD SPI Mode Interrupt Flag Register (USIL_LSIF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL LCD SPI Mode Interrupt Flag Register (USIL_LSIF)	0x300682 (8 bits)	D7-2	–	reserved	–		–	–	X when being read.
		D1	LSBSY	Transfer busy flag	1 Busy	0 Idle	0	R	
		D0	LSTDIF	Transmit buffer empty flag	1 Empty	0 Not empty	0	R/W	Reset by writing 1.

Note: This register is effective only in LCD SPI mode. Configure USIL to LCD SPI mode before setting this register.

D[7:2] Reserved

D1 LSBSY: Transfer Busy Flag Bit

Indicates the LCD SPI transfer status.

1 (R): Busy

0 (R): Idle (default)

LSBSY is set to 1 when the LCD SPI starts data transfer and is maintained at 1 while transfer is underway. It is cleared to 0 after data transfer for the data size set using LSDMOD[1:0]/USIL_LSDCFG register has completed.

D0 LSTDIF: Transmit Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Empty

0 (R): Data exists (default)

1 (W): Reset to 0

0 (W): Ignored

LSTDIF is set to 1 when the transmit data written to the transmit data buffer is transferred to the shift register (when transmission starts), indicating that the next transmit data can be written to. At the same time a transmit buffer empty interrupt request is sent to the ITC if LSTDIE/USIL_LSIE register is 1. LSTDIF is reset by writing 1.

USIL LCD SPI Mode Data Configuration Register (USIL_LSDFCFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL LCD SPI Mode Data Configuration Register (USIL_LSDFCFG)	0x30068f (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.	
		D3-2	LS18DFM [1:0]	LCD SPI 18-bit data format select	LS18DFM[1:0]	Data format	0x0	R/W	
					0x3	Format 3			
					0x2	Format 2			
0x1	Format 1								
0x0	Format 0								
D1-0	LSDMOD [1:0]	LCD SPI data mode select	LSDMOD[1:0]	Data mode	0x0	R/W			
			0x3	24-bit mode					
			0x2	18-bit mode					
			0x1	16-bit mode					
0x0	8-bit mode								

Note: This register is effective only in LCD SPI mode. Configure USIL to LCD SPI mode before setting this register.

D[7:4] Reserved

D[3:2] LS18DFM[1:0]: LCD SPI 18-bit Data Format Select bits

Selects a data format in 18-bit mode. (See Figure 19.8.6.)

Table 19.8.7 LCD SPI 18-bit Data Format

LS18DFM[1:0]	Data format
0x3	Format 3
0x2	Format 2
0x1	Format 1
0x0	Format 0

(Default: 0x0)

D[1:0] LSDMOD[1:0]: LCD SPI Data Mode Select Bit

Selects the LCD SPI data mode.

Table 19.8.8 LCD SPI Data Mode

LSDMOD[1:0]	Data mode
0x3	24-bit mode
0x2	18-bit mode
0x1	16-bit mode
0x0	8-bit mode

(Default: 0x0)

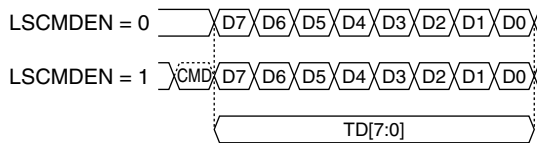


Figure 19.8.4 8-bit Data Format



Figure 19.8.5 16-bit Data Format

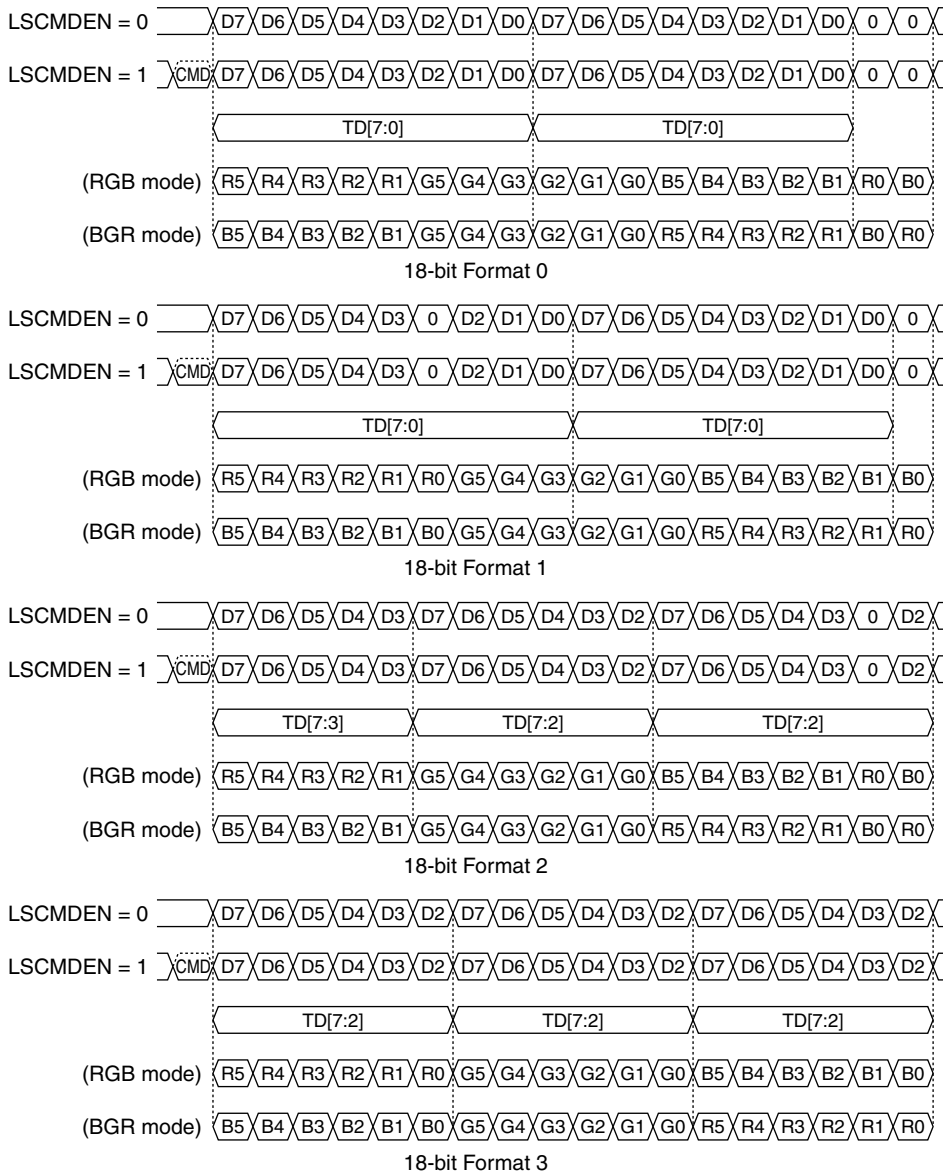


Figure 19.8.6 18-bit Data Format

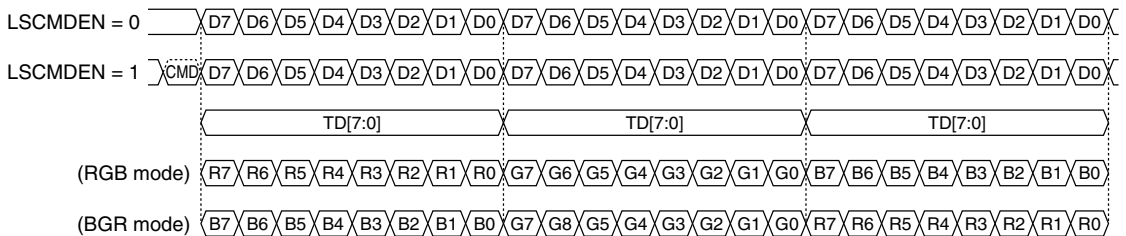


Figure 19.8.7 24-bit Data Format

USIL LCD Parallel I/F Mode Configuration Register (USIL_LPCFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL LCD Parallel I/F Mode Configuration Register (USIL_LPCFG)	0x300690 (8 bits)	D7-3	-	reserved	-	-	-	0 when being read.
		D2	LPSRDEN	Successive read enable	1 Enable 0 Disable	0	R/W	
		D1	LPCMD	Command bit	1 High 0 Low	0	R/W	
		D0	LPRD	Read trigger	1 Trigger 0 Ignored	0	W	
					1 Read cycle 0 Read finished		R	

19 UNIVERSAL SERIAL INTERFACE WITH LCD INTERFACE (USIL)

Note: This register is effective only in LCD parallel mode. Configure USIL to LCD parallel mode before setting this register.

D[7:3] Reserved

D2 LPSRDEN: Successive Read Enable Bit

Enables the successive read function.

1 (R/W): Enabled

0 (R/W): Disabled (default)

By issuing a read trigger (writing 1 to LPRD) after setting LPSRDEN to 1, the LCD parallel interface repeats data reading from the LCD driver/panel while LPSRDEN is 1. When LPSRDEN is set to 0, the LCD parallel interface stops data reading after the read cycle being currently executed has finished.

D1 LPCMD: Command bit

Sets the command bit value.

1 (R/W): High

0 (R/W): Low (default)

The command bit selected using LPCMD is output from the USIL_DI (lcdp_a0) pin.

D0 LPRD: Read Trigger Bit

Starts a read cycle of the LCD parallel interface.

1 (W): Trigger (start reading)

0 (W): Ignored

1 (R): During reading

0 (R): Read cycle has finished

To read data from the LCD driver/panel via the LCD parallel interface, issue a read trigger by writing 1 to LPRD. Set the command bit (LPCMD) value before writing to LPRD. The command bit value set is output from the USIL_DI pin immediately after it is written to the register. Then it loads the LCD_D[7:0] pin status to the read (receive data) buffer (RD[7:0]/USIL_RD register). LPRD retains 1 until the read data is loaded to the read buffer.

USIL LCD Parallel I/F Mode Interrupt Enable Register (USIL_LPIE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL LCD Parallel I/F Mode Interrupt Enable Register (USIL_LPIE)	0x300691 (8 bits)	D7-2	—	reserved	—	—	—	0 when being read.
		D1	LPRDIE	Read buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	LPWRIE	Write buffer empty interrupt enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in LCD parallel mode. Configure USIL to LCD parallel mode before setting this register.

D[7:2] Reserved

D1 LPRDIE: Read Buffer Full Interrupt Enable Bit

Enables interrupt requests to the ITC when data is loaded to the read (receive data) buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to read data using interrupts.

D0 LPWRIE: Write Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC when data written to the write (transmit data) buffer is output via the LCD_D[7:0] pins.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to write data to the write buffer using interrupts.

USIL LCD Parallel I/F Mode Interrupt Flag Register (USIL_LPIF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL LCD Parallel I/F Mode Interrupt Flag Register (USIL_LPIF)	0x300692 (8 bits)	D7-3	–	reserved	–	–	–	X when being read.
		D2	LPBSY	Transfer busy flag	1 Busy 0 Idle	0	R	
		D1	LPRDIF	Read buffer full flag	1 Full 0 Not full	0	R/W	Reset by writing 1.
		D0	LPWRIF	Write buffer empty flag	1 Empty 0 Not empty	0	R/W	

Note: This register is effective only in LCD parallel mode. Configure USIL to LCD parallel mode before setting this register.

D[7:3] Reserved

D2 LPBSY: Transfer Busy Flag Bit

Indicates the LCD parallel interface status.

1 (R): Busy

0 (R): Idle (default)

LPBSY is set to 1 when the LCD parallel interface starts data transfer and is maintained at 1 while transfer is underway. It is cleared to 0 once the transfer is completed.

D1 LPRDIF: Read Buffer Full Flag Bit

Indicates the read (receive data) buffer status.

1 (R): Data full

0 (R): No data (default)

1 (W): Reset to 0

0 (W): Ignored

LPRDIF is set to 1 when data received is loaded to the read buffer (when receiving is completed), indicating that the data can be read. At the same time a read buffer full interrupt request is sent to the ITC if LPRDIE/USIL_LPIE register is 1. LPRDIF is reset by writing 1.

D0 LPWRIF: Write Buffer Empty Flag Bit

Indicates the write (transmit data) buffer status.

1 (R): Empty

0 (R): Data exists (default)

1 (W): Reset to 0

0 (W): Ignored

LPWRIF is set to 1 when the data written to the write buffer is output via the LCD_D[7:0] pins, indicating that the next data can be written to. At the same time a write buffer empty interrupt request is sent to the ITC if LPWRIE/USIL_LPIE register is 1. LPWRIF is reset by writing 1.

USIL LCD Parallel I/F Mode Access Timing Register (USIL_LPAC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL LCD Parallel I/F Mode Access Timing Register (USIL_LPAC)	0x30069f (8 bits)	D7-6	LPHD[1:0]	Hold cycle	LPHD[1:0]	Hold cycle	0x0	R/W	
					0x3	4 cycles			
					0x2	3 cycles			
					0x1	2 cycles			
					0x0	1 cycle			
		D5-4	LPST[1:0]	Setup cycle	LPST[1:0]	Setup cycle	0x0	R/W	
					0x3	4 cycles			
					0x2	3 cycles			
					0x1	2 cycles			
					0x0	1 cycle			
		D3-0	LPWT[3:0]	Wait cycle	LPWT[3:0]	Wait cycle	0x0	R/W	
					0xf	15 cycles			
0xe	14 cycles								
:	:								
0x1	1 cycle								
0x0	0 cycles								

Note: This register is effective only in LCD parallel mode. Configure USIL to LCD parallel mode before setting this register.

D[7:6] LPHD[1:0]: Hold Cycle Bits

Configures the hold cycle of the LCD parallel interface.

Table 19.8.9 Hold Cycle Settings

LPHD[1:0]	Number of hold cycles
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x0)

D[5:4] LPST[1:0]: Setup Cycle Bits

Configures the setup cycle of the LCD parallel interface.

Table 19.8.10 Setup Cycle Settings

LPST[1:0]	Number of setup cycles
0x3	4 cycles
0x2	3 cycles
0x1	2 cycles
0x0	1 cycle

(Default: 0x0)

D[3:0] LPWT[3:0]: Wait Cycle Bits

Configures the wait cycle of the LCD parallel interface.

Table 19.8.11 Wait Cycle Settings

LPWT[3:0]	Number of wait cycles
0xf	15 cycles
0xe	14 cycles
:	:
0x1	1 cycle
0x0	0 cycles

(Default: 0x0)

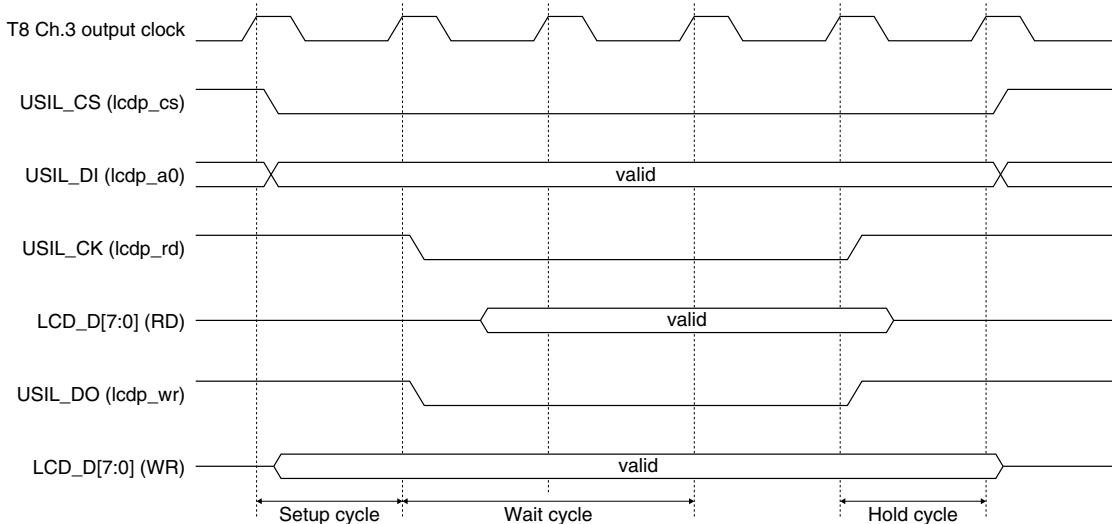


Figure 19.4.8.8 Access Timing Parameters

19.9 Precautions

Interface mode setting

Be sure to perform software reset (USILMOD[2:0]/USIL_GCFCG register = 0x0) and set the interface mode (USILMOD[2:0]/USIL_GCFCG register = 0x1 to 0x7) before changing other USIL configurations.

Receiving control byte in I²C slave mode

The external I²C master device sends a control byte to the I²C slave device when an ACK has been received after sending a slave address. The subsequent operations of the slave device are determined by the control byte.

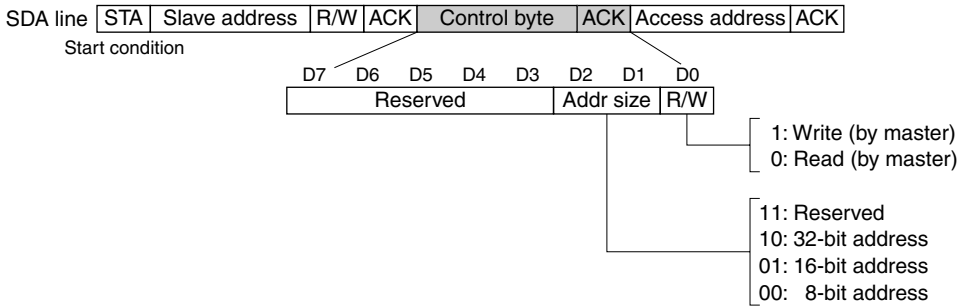


Figure 19.9.1 Control Byte Sent from I²C Master

I²C master write (data receiving from master)

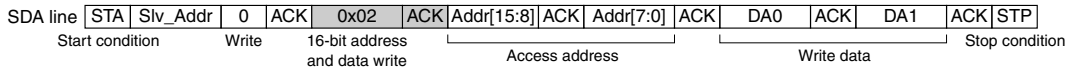


Figure 19.9.2 I²C Master Write (Data Receiving from Master)

The control byte specifies the access address size and writing operations. The received data that follow the control byte should be used as the address and the data to be written according to the access address size.

I²C master read (data transmission to master)

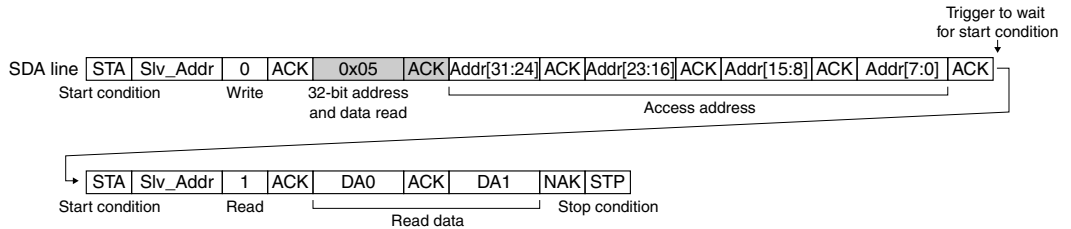


Figure 19.9.3 I²C Master Read (Data Transmission to Master)

The master sends the access address following the control byte. Perform data reception for the control byte and address data to determine the address from which transmit data is read. After sending an ACK for Addr 0, set ISTGMOD[2:0]/USIL_ISTG register to 0x0 and ISTG/USIL_ISTG register to 1 to wait for a start condition that will be sent from the master for reading data (for the slave to sent the read data).

20 General-Purpose Serial Interface (FSIO)

20.1 FSIO Module Overview

The S1C33L26 contains two channels (Ch.0 and Ch.1) of serial interfaces, the features of which are described below.

- A clock-synchronized or asynchronous mode can be selected for the transfer method.
 - Clock-synchronized mode
 - Data length: 8 bits, fixed (No start, stop, and parity bits)
 - Receive error: An overrun error can be detected.
 - Asynchronous mode
 - Data length: 7 or 8 bits, selectable
 - Receive error: Overrun, framing, or parity errors can be detected.
 - Start bit: 1 bit, fixed
 - Stop bit: 1 or 2 bits, selectable
 - Parity bit: Even, odd, or none, selectable
- Since the transmit and receive units are independent, full-duplex communication is possible. Supports IrDA1.0-equivalent communications by software control or using an external IrDA driver. Internal clock or external clock is selectable.
- 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 DMAC are possible.
- Three types of interrupts (transmit buffer empty, receive buffer full, and receive error) can be generated.

Figure 20.1.1 shows the configuration of the FSIO module (one channel).

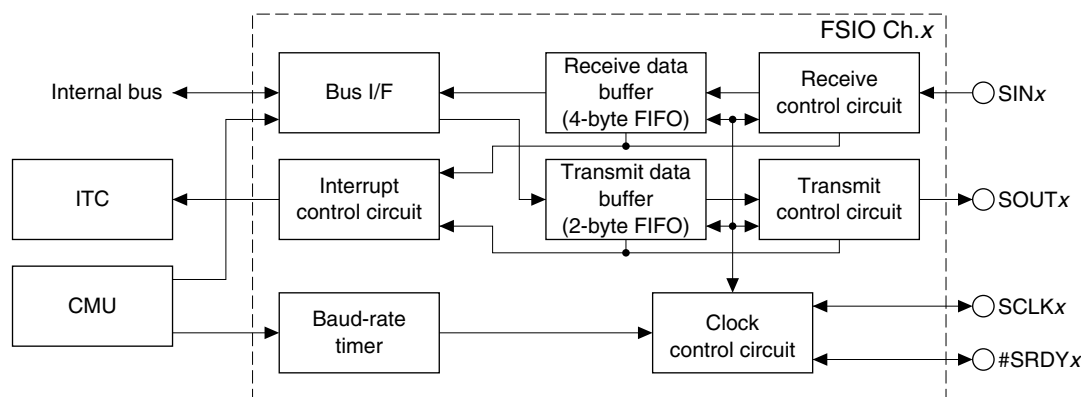


Figure 20.1.1 FSIO Configuration

Note: Two channels in the FSIO module have the same functions except for control register addresses. For this reason, the description in this chapter applies to both FSIO channels. The 'x' in the register name indicates the channel number (0 or 1).

Example: FSIO_CTLx register

Ch.0: FSIO_CTL0 register

Ch.1: FSIO_CTL1 register

20.2 FSIO Pins

Table 20.2.1 lists the I/O pins provided for the FSIO module.

Table 20.2.1 List of FSIO Pins

Pin name	I/O	Qty	Function
SINO SIN1	I	2	FSIO serial data input pin Inputs serial data.
SOUT0 SOUT1	O	2	FSIO serial data output pin Outputs serial data.
SCLK0 SCLK1	I/O	2	FSIO clock input/output pin Inputs or outputs a clock. In clock-synchronized slave mode, it is used as a clock input pin; in clock-synchronized master mode, it is used as a clock output pin. In asynchronous mode, this pin is used as a clock input when an external clock is used. This pin can be used as an I/O port when the internal clock is used.
#SRDY0 #SRDY1	I/O	2	FSIO ready-signal input/output pin Inputs or outputs the ready signal used in clock-synchronized mode. In clock-synchronized slave mode, it is used as the ready-signal output pin; in clock-synchronized master mode, it is used as the ready-signal input pin. This pin can be used as an I/O port in asynchronous mode.

The FSIO pins (SIN_x, SOUT_x, SCLK_x, #SRDY_x) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as FSIO pins.

For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

20.3 FSIO Operating Clock

FSIO Ch.0 and Ch.1 use PCLK1 and PCLK2 as the operating clock, respectively. Therefore, PCLK1 and/or PCLK2 must be supplied from the CMU before starting the FSIO including setting the control registers. For more information on the PCLK1/PCLK2 supply, refer to the “Clock Management Unit (CMU).”

20.4 Mode Settings

20.4.1 Interface Mode and Transfer Mode

The interface type and transfer mode of the serial interface can be configured using SMD[1:0]/FSIO_CTL_x register and IRMD[1:0]/FSIO_IRDA_x register individually for each channel as shown in the table below.

Table 20.4.1.1 Interface and Transfer Mode Settings

IRMD[1:0]	SMD[1:0]	Interface mode	Transfer mode
0x2	0x3	IrDA mode	8-bit asynchronous mode (IrDA I/F)
	0x2		7-bit asynchronous mode (IrDA I/F)
0x0	0x3	Normal mode	8-bit asynchronous mode (normal I/F)
	0x2		7-bit asynchronous mode (normal I/F)
	0x1		Clock-synchronized slave mode
	0x0		Clock-synchronized master mode
Other			Reserved

At initial reset, SMD[1:0] and IRMD[1:0] are both set to 0 (clock-synchronized master mode).

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 20.4.1.2.

Table 20.4.1.2 Pin Configuration by Transfer Mode

Transfer mode	SIN _x	SOUT _x	SCLK _x	#SRDY _x
8-bit asynchronous mode	Data input	Data output	Clock input/P port	P port
7-bit asynchronous mode	Data input	Data output	Clock input/P port	P port
Clock-synchronized slave mode	Data input	Data output	Clock input	Ready output
Clock-synchronized master mode	Data input	Data output	Clock output	Ready input

All four pins are used in the clock-synchronized mode.

In the asynchronous mode, since no ready signal is used, the #SRDY_x pin can be used as an I/O (P) port. In addition, when an external clock is not used, the SCLK_x pin can also be used as an I/O port.

The I/O control and data registers for the I/O ports used in the serial interface can be used as general-purpose read/write registers.

20.4.2 Standard Mode and Advanced Mode

The serial interface in the S1C33L26 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 20.4.2.1 shows differences between standard mode and advanced mode.

Table 20.4.2.1 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/FSIO_ADV_x register to 1. The control bits for the extended functions are enabled to write after this setting. At initial reset, SIOADV is set to 0 and the serial interface enters standard mode.

The following descriptions unless otherwise specified are common contents for both modes. The extended functions in advanced mode are explained assuming that SIOADV has been set to 1.

20.5 Baud-Rate Timer (Baud Rate Setting)

The clock-synchronized master mode uses an internal clock for data transfer. Also in the asynchronous mode, the internal clock can be selected as the operating clock. Each channel has an embedded baud-rate timer (12-bit programmable timer) to generate this clock. The counter initial value can be set via 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.

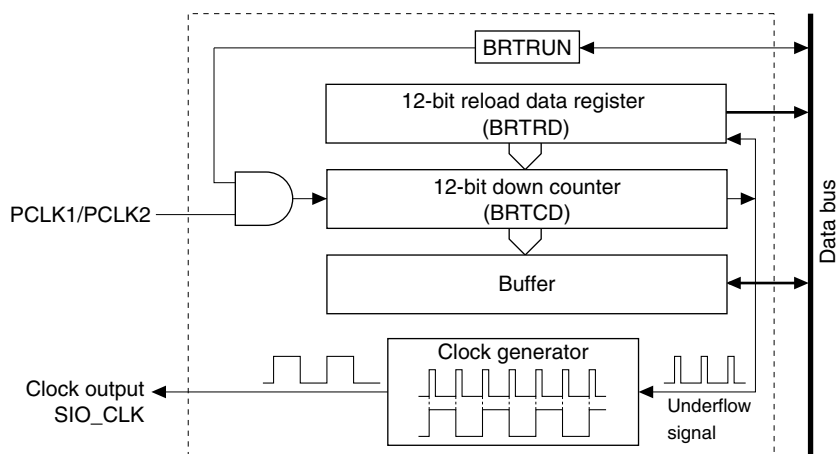


Figure 20.5.1 Transfer Clock Generation by the Baud-Rate Timer

20 GENERAL-PURPOSE SERIAL INTERFACE (FSIO)

The baud-rate timer is configured with a 12-bit presetable down counter (BRTCD[11:0]/FSIO_BRTCDH_x and FSIO_BRTCDL_x registers) and a 12-bit reload data register (BRTRD[11:0]/FSIO_BRTRDH_x and FSIO_BRTRDL_x registers) for setting an initial value to the counter.

The baud-rate timer uses the PCLK1 clock (Ch.0)/PCLK2 clock (Ch.1) supplied from the CMU as the count clock. For details on how to set and control the PCLK1/PCLK2 clock, see the “Clock Management Unit (CMU)” chapter.

The following procedure generates the clock by the baud-rate timer.

1. Set an initial value to the reload data register BRTRD[11:0].
2. Set BRTRUN/FSIO_BRTRUN_x register to 1.

The baud-rate timer loads the initial value set in the reload data register to the counter when 1 is written to BRTRUN, and then starts counting down. When the counter underflows, it outputs an underflow pulse and loads the reload data again to continue counting.

The underflow occurs in the cycle determined by the reload data. The clock generator reverses its output signal level using the underflow signal to generate a clock with 50% duty ratio and 1/2 the frequency of the underflow signal. The baud-rate timer should be stopped (set BRTRUN to 0) when serial communication is not needed to reduce current consumption.

Calculating the reload data

The initial value for the reload data register is determined by the expressions shown below. Note that the expression depends on the transfer mode.

Clock-synchronized master mode

$$\text{BRTRD} = \frac{f_{\text{PCLK}}}{2 \times \text{bps}} - 1$$

BRTRD: Reload data register setup value of the baud-rate timer

f_{PCLK}: Baud-rate timer operating clock (PCLK1 or PCLK2) frequency

bps: Transfer rate (bits/second)

Asynchronous mode

$$\text{BRTRD} = \frac{f_{\text{PCLK}} \times \text{DIVMD}}{2 \times \text{bps}} - 1$$

BRTRD: Reload data register setup value of the baud-rate timer

f_{PCLK}: Baud-rate timer operating clock (PCLK1 or PCLK2) frequency

bps: Transfer rate (bits/second)

DIVMD: Internal division ratio (1/16 or 1/8 selected by DIVMD/FSIO_IRDA_x register)

Note: Be aware that a certain period of time is required before serial data transfer can be started after starting the baud-rate timer, as there is a delay between start of the baud-rate timer and output of the first underflow pulse, especially when a low baud-rate is set.

20.6 Clock-Synchronized Interface

20.6.1 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. This transfer mode supports half-duplex communication, as the clock line is shared between the transmit and receive units.

Master and slave modes

Either the clock-synchronized master mode or the clock-synchronized slave mode can be selected using SMD[1:0]/FSIO_CTL_x register.

Clock-synchronized master mode (SMD[1:0] = 0x0)

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 SCLK_x pin, enabling an external (slave side) serial input/output device to be controlled. The #SRDY_x 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 (SMD[1:0] = 0x1)

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 SCLK_x pin for use as the synchronizing clock of the serial interface. In addition, a #SRDY_x signal indicating whether the serial interface is ready to transmit or receive (when ready in a low level) is output from the #SRDY_x pin.

Figure 20.6.1.1 shows an example of how the input/output pins are connected in the clock-synchronized mode.

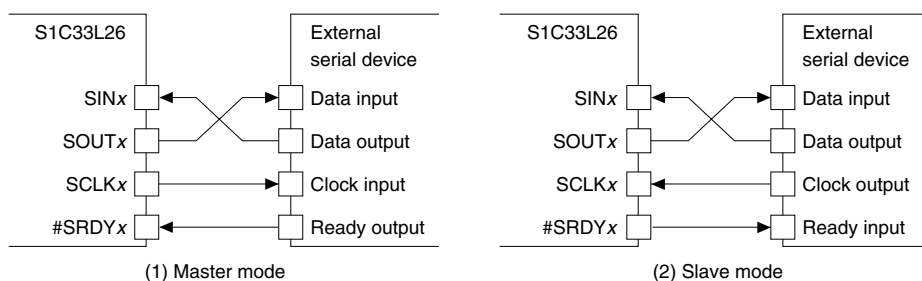


Figure 20.6.1.1 Example of Connection in Clock-Synchronized Mode

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

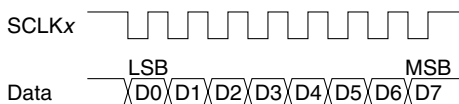


Figure 20.6.1.2 Clock-Synchronized Transfer Data Format

Serial data is transmitted and received starting with the LSB.

20.6.2 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 DMA

The following explains the content of each setting. For details on interrupt/DMA settings, refer to Section 20.9, "FSIO Interrupts and DMA."

Note: Always make sure the serial interface is inactive (TXEN/FSIO_CTL_x register and RXEN/FSIO_CTL_x register = 0) before these settings are made. A change of settings during operation may cause a malfunction.

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 bits to enable these pin functions according to the channel to be used. For details of pin functions and how to switch over, see the “I/O Ports (GPIO)” chapter.

Setting the interface mode

Write 0x0 to IRMD[1:0]/FSIO_IRDA x register to choose the normal interface.

Setting the transfer mode

Use SMD[1:0]/FSIO_CTL x register 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[1:0] to 0x0; when using the serial interface as a slave, set SMD[1:0] to 0x1.

Setting the input clock

Clock-synchronized master mode

This mode operates with 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 20.5, “Baud-Rate Timer (Baud Rate Setting).”

Clock-synchronized slave mode

This mode operates with 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.

Setting the receive FIFO level (advanced mode)

This serial interface incorporates a 4-byte receive FIFO allowing up to 4 bytes of data to 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 FIFOINT[1:0]/FSIO_IRDA x register to set this number of data. Writing 0–3 to FIFOINT[1:0] 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.

20.6.3 Control and Operation of Clock-Synchronized Transfer

Transmit control

(1) Enabling transmit operation

Use the transmit-enable bit TXEN/FSIO_CTL x register 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 SCLK x pin is also enabled (ready for input/output).

Transmit is disabled and the transmit data buffer (FIFO) is cleared by writing 0 to TXEN.

After the port function select bits are set for the serial inputs/outputs, 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, TXEN and the receive-enable bit (RXEN/FSIO_CTL x register) cannot be enabled simultaneously. When transmitting data, fix RXEN at 0 and do not change it during a transmit operation.

In addition, make sure that TXEN is not set to 0 during a transmit operation.

(2) Transmit procedure

The serial interface contains a transmit shift register and a transmit data register, which are provided independently of those used for a receive operation.

Transmit data is written to TXD[7:0]/FSIO_TXD x register. The data written to TXD[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. The next transmit data can be written to the transmit data register, even during data transmission. A transmit data buffer status flag (TDBE/FSIO_STATUS x register) is provided to check whether this buffer is full or not. This flag is set to 1 when the transmit data buffer has a free space for transmit data to be written and reset to 0 when the transmit data buffer becomes full by writing transmit data.

The serial interface starts transmitting when data is written to the transmit data register. The transmit shift register status can be checked using the transmit-completion flag (TEND/FSIO_STATUS x register). This flag goes 1 after the first bit is shifted out from the shift register and goes 0 after the last bit is shifted out.

When data is transmitted successively in clock-synchronized master mode, TEND maintains 1 until all data is shifted out (Figure 20.6.3.1). In slave mode, TEND goes 0 every time 1-byte data is shifted out (Figure 20.6.3.2).

Note: TEND goes 0 at the falling edge of SCLK x to indicate that all the transmit data bits in the transmit shift register are shifted out. Be aware that there is a half SCLK x cycle interval between setting TEND to 0 and latching the last bit by the receiver.

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 by setting the interrupt control bits, the subsequent 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 20.9, “FSIO Interrupts and DMA.”

The following describes 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 20.6.3.1 shows a transmit timing chart in the clock-synchronized master mode.

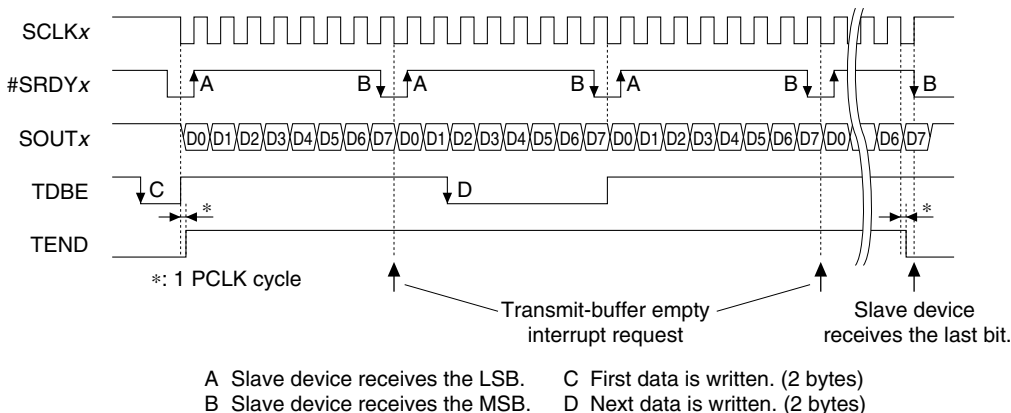


Figure 20.6.3.1 Transmit Timing Chart in Clock-Synchronized Master Mode

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 20.6.3.2 shows a transmit timing chart in the clock-synchronized slave mode.

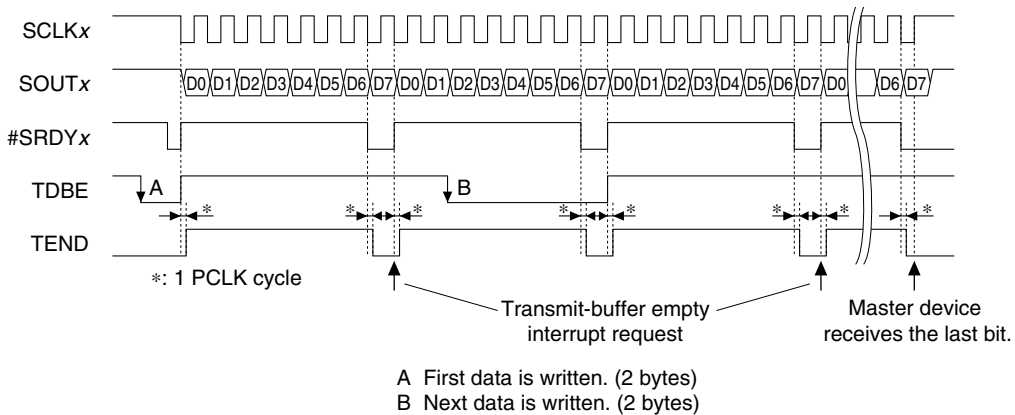


Figure 20.6.3.2 Transmit Timing Chart in Clock-Synchronized Slave Mode

1. After setting the #SRDY_x 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 SCLK_x 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 SOUT_x pin. If the transmit data buffer becomes empty at this point, a transmit-buffer empty interrupt request occurs. The #SRDY_x 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 SOUT_x. This operation is repeated until all 8 bits of data are transmitted.
4. The #SRDY_x signal is set to a low level when the last bit (8th bit) is output from the SOUT_x 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 TXEN 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 TXEN.

Receive control

(1) Enabling receive operation

Use the receive-enable bit RXEN/FSIO_CTL_x register 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.

After the port function select bits are set for the serial inputs/outputs, 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 and the transmit-enable bit (TXEN/FSIO_CTL_x register) cannot be enabled simultaneously. When receiving data, fix TXEN at 0 and do not change it during a receive operation. In addition, make sure RXEN is not set to 0 during a receive operation.

(2) Receive procedure

This serial interface has a receive shift register, receive data buffer and a receive data register that are provided independently of those used for transmit operations.

The received data enters the received data buffer. The receive data buffer is a 4-byte FIFO and can receive data until it becomes full unless the received data is not read out.

The received data in the buffer can be read by accessing RXD[7:0]/FSIO_RXD_x register. The older data is output first and cleared by reading.

The number of data in the receive data buffer can be checked by reading RXDNUM[1:0]/FSIO_STATUS_x register. When RXDNUM[1:0] is 0, the buffer contains 0 or 1 data. When RXDNUM[1:0] is 1–3, the buffer contains 2–4 data.

Furthermore, RDBF/FSIO_STATUS_x register 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.

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 by setting the interrupt control bits, the received data can be read by an interrupt processing routine.

In addition, a DMA can be invoked every time the received data is written to the receive-buffer, allowing the received data to be transferred successively to the specified memory location through DMA transfers.

For details on how to control interrupts/DMA, refer to Section 20.9, “FSIO Interrupts and DMA.”

The following describes a receive operation in the master and slave modes.

Clock-synchronized master mode

Figure 20.6.3.3 shows a receive timing chart in the clock-synchronized master mode.

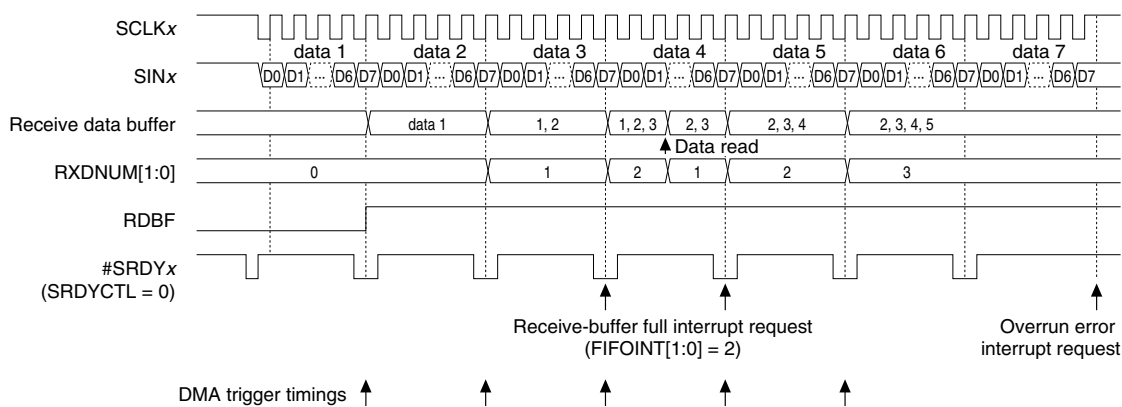


Figure 20.6.3.3 Receive Timing Chart in Clock-Synchronized Master Mode

1. If the #SRDY_x signal from the slave is on a high level, the master waits until it turns to a low level (ready to transmit).
2. If #SRDY_x is on a low level, 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 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 SIN_x 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 20.6.3.4 shows a receive timing chart in the clock-synchronized slave mode.

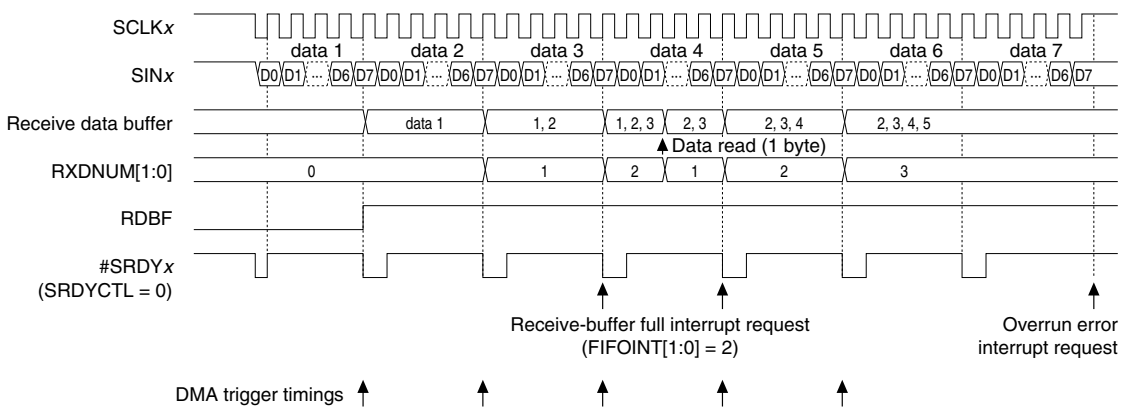


Figure 20.6.3.4 Receive Timing Chart in Clock-Synchronized Slave Mode

1. After setting the #SRDY_x 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 SIN_x input into the shift register at the rising edges of the clock that is input from SCLK_x. 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/FSIO_STATUS_x register) is set to 1. Once the overrun error flag is set to 1, it remains set until it is reset by writing 0 to it in the software. The overrun error is one of the receive-error interrupt causes in the serial interface. An interrupt can be generated for this error by setting the interrupt control bits 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 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 subsequent 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.

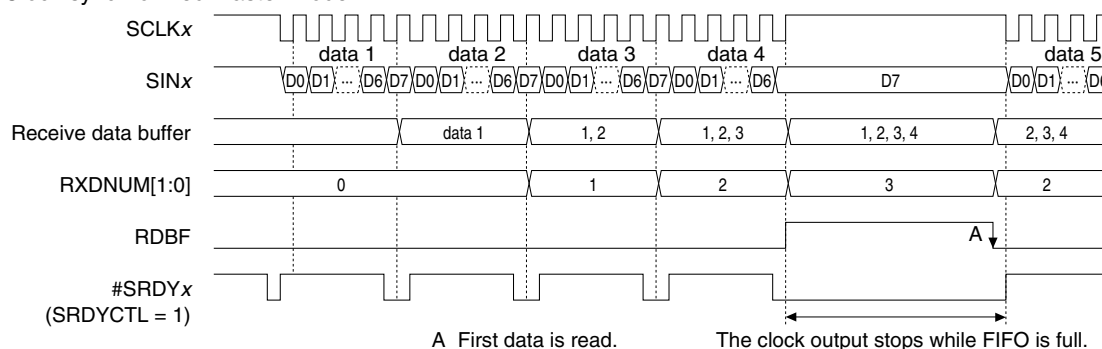
In normal mode, the #SRDY_x signal indicating ready to receive is output even if the receive data buffer is full. If the receive data buffer cannot be read in this case, an overrun error occurs in the next data transfer. To prevent this error, the serial interface provides #SRDY_x high mask mode. In 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/FSIO_IRDA_x register to 1.

This function is effective in 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.

Clock-synchronized master mode



Clock-synchronized slave mode

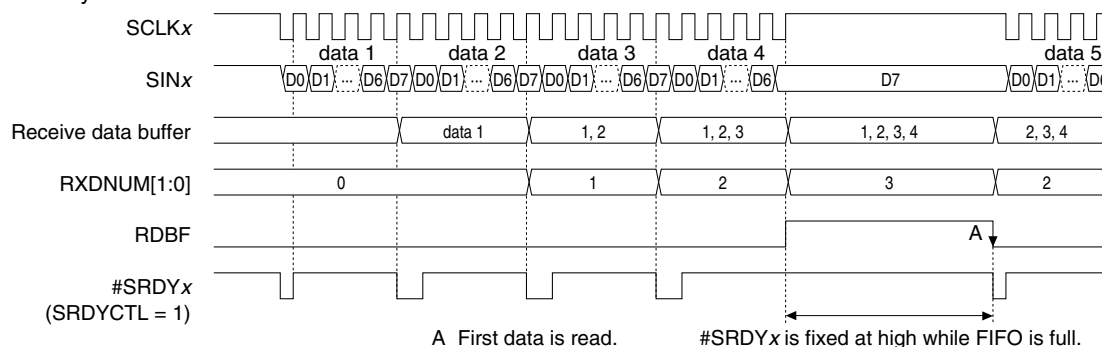


Figure 20.6.3.5 #SRDY_x High Mask Mode

(5) Terminating receive operation

Upon completion of a data receive operation, write 0 to the receive-enable bit RXEN 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 to 0.

20.7 Asynchronous Interface

20.7.1 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 (SMD[1:0]/FSIO_CTLx register = 0x3), 8 bits of data can be transferred; in the 7-bit asynchronous mode (SMD[1:0] = 0x2), 7 bits of data can be transferred.

In either mode, it is possible to select the stop-bit length, add a parity bit, and choose between even and odd parity. The start bit is fixed at 1.

The operating clock can be selected between an internal clock generated by the baud-rate timer or an external clock that is input from the 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 20.7.1.1 shows an example of how input/output pins are connected for transfers in the asynchronous mode.

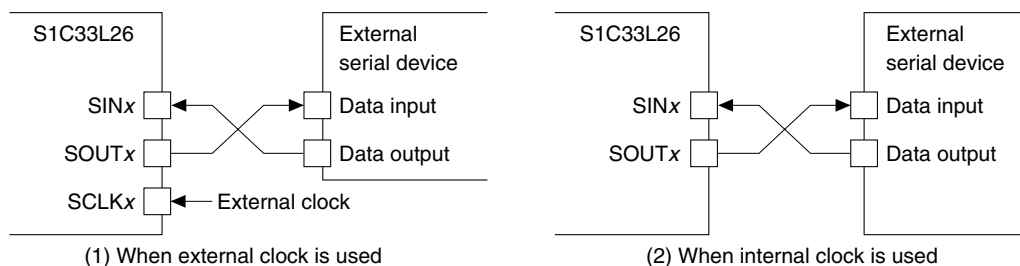


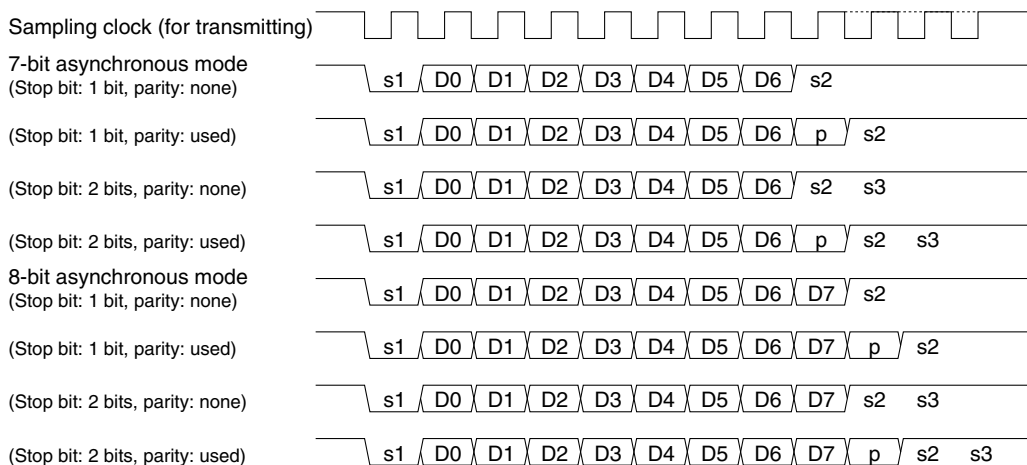
Figure 20.7.1.1 Example of Connection in Asynchronous Mode

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



s1: start bit, s2 & s3: stop bit, p: parity bit

Figure 20.7.1.2 Data Format for Asynchronous Transfer

Serial data is transmitted and received, starting with the LSB.

20.7.2 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 interrupts and DMA

The following describes how to set each of the above. For details on interrupt/DMA settings, refer to Section 20.9, “FSIO Interrupts and DMA.”

Note: Always make sure the serial interface is inactive (TXEN/FSIO_CTLx register and RXEN/FSIO_CTLx register = 0) before making these settings. A change in settings during operation may result in a malfunction.

Setting input/output pins

In the asynchronous mode, two pins—SINx and SOUTx—are used. When an external clock is used, one more pin, SCLKx, is also used. Configure the port function select bits to enable these pin functions according to the channel to be used. For details of pin functions and how to switch over, see the “I/O Ports (GPIO)” chapter.

Setting the interface mode

Initialize IRMD[1:0]/FSIO_IRDAx register by writing 0x0 when using the serial interface as a normal interface, or 0x2 when using the serial interface as an IrDA interface. This setting must be made before a transfer mode is set.

Setting the transfer mode

Use SMD[1:0]/FSIO_CTLx register 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[1:0] to 0x3, when using the serial interface in the 7-bit asynchronous mode, set SMD[1:0] to 0x2.

Setting the input clock

In the asynchronous mode, the operating clock can be selected between the internal clock and an external clock using SSCK/FSIO_CTLx register.

An external clock is selected (input from the SCLKx pin) by writing 1 to SSCK, and an internal clock is selected by writing 0.

Internal clock

When 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 20.5, “Baud-Rate Timer (Baud Rate Setting).”

External clock

When external clock is selected, the serial interface is clocked by a clock input from the SCLKx pin. Therefore, there is no need to control the baud-rate timer.

Any desired clock frequency can be set. The clock input from the SCLKx pin is internally divided by 16 or 8 in the serial interface, in order to create a sampling clock (refer to “Sampling clock”). This division ratio must also be considered when setting the transfer rate.

Sampling clock

In the asynchronous mode, SIO_CLK (the clock output by the baud-rate timer or input from the SCLKx pin) is internally divided in the serial interface, in order to create a sampling clock.

20 GENERAL-PURPOSE SERIAL INTERFACE (FSIO)

A 1/16 division ratio is selected by writing 0 to DIVMD/FSIO_IRDA_x register, and a 1/8 ratio is selected by writing 1.

Note: DIVMD 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

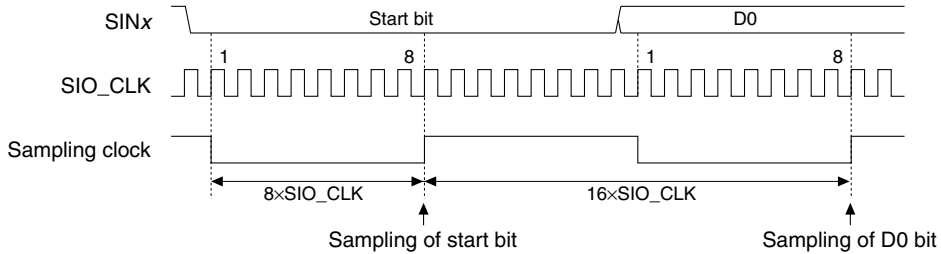


Figure 20.7.2.1 Sampling Clock for Asynchronous Receive Operation (when 1/16 division is selected)

Each bit data is sampled in the timing shown in Figure 20.7.2.1. 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 $8 \times \text{SIO_CLK}$ ($4 \times \text{SIO_CLK}$ 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 $16 \times \text{SIO_CLK}$ cycles ($8 \times \text{SIO_CLK}$ cycles when 1/8 division is selected).

For transmitting

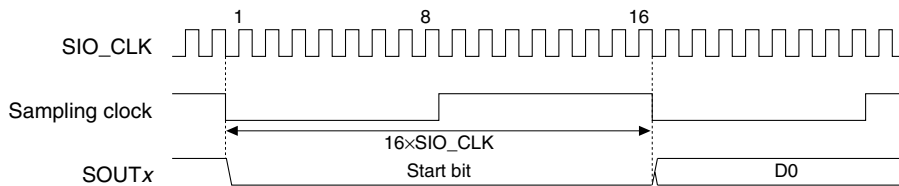


Figure 20.7.2.2 Sampling Clock for Asynchronous Transmit Operation (when 1/16 division is selected)

During transmission, each bit data is output from the SOUT_x pin in $16 \times \text{SIO_CLK}$ cycles ($8 \times \text{SIO_CLK}$ 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 configured as shown in the Table 20.7.2.1 using the control bits listed below.

Stop bit selection: STPB/FSIO_CTL_x register

Parity enable: EPR/FSIO_CTL_x register

Parity mode selection: PMD/FSIO_CTL_x register

Table 20.7.2.1 Stop Bit and Parity Bit Settings

STPB	EPR	PMD	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	None

* PMD settings are ineffective when EPR = 0.

(Default: STPB = EPR = PMD = 0)

Setting the receive FIFO level (advanced mode)

This serial interface incorporates a 4-byte receive FIFO allowing up to 4 bytes of data to 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 FIFOINT[1:0]/FSIO_IRDA_x register to set this number of data. Writing 0–3 to FIFOINT[1:0] 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.

20.7.3 Control and Operation of Asynchronous Transfer

Transmit control

(1) Enabling transmit operation

Use the transmit-enable bit TXEN/FSIO_CTL_x register 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 TXEN.

Note: Do not set TXEN 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 TXD[7:0]/FSIO_TXD_x register.

In the 7-bit asynchronous mode, bit 7 (MSB) in each register is ignored.

The data written to TXD[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. The next transmit data can be written to the transmit data register, even during data transmission. The transmit data buffer status flag (TDBE/FSIO_STATUS_x register) is provided to check whether this buffer is full or not. This flag is set to 1 when the transmit data buffer has a free space for transmit data to be written and reset to 0 when the transmit data buffer becomes full by writing transmit data.

The serial interface starts transmitting when data is written to the transmit data register. The transfer status can be checked using the transmit-completion flag (TEND/FSIO_STATUS_x register). This flag goes 1 when data is being transmitted and goes 0 when the transmission has completed.

When all the data in the transmit data buffer are transferred, a cause of the transmit-data empty interrupt occurs. Since an interrupt can be generated as set by the interrupt control bits, 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 20.9, “FSIO Interrupts and DMA.”

Figure 20.7.3.1 shows a transmit timing chart in the asynchronous mode.

Example: Data length: 8 bits, Stop bit: 1 bit, Parity bit: Included

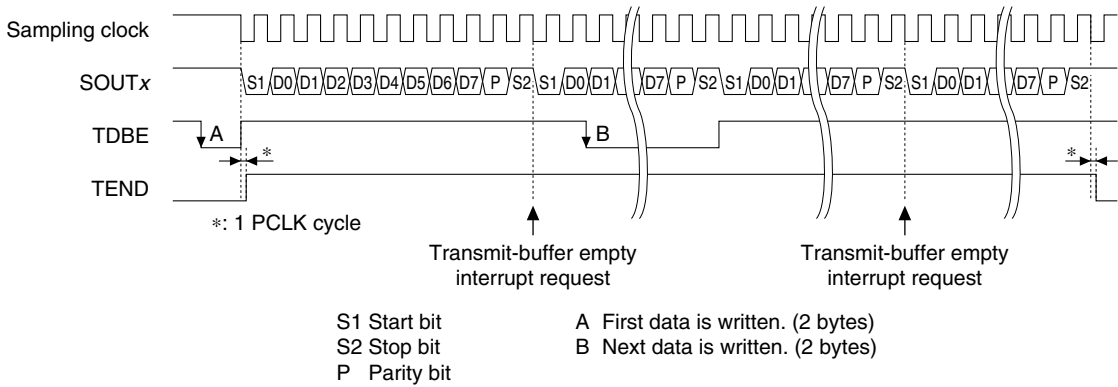


Figure 20.7.3.1 Transmit Timing Chart in Asynchronous Mode

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 SOUTx pin is setting to a low level to send the start bit.
2. Each bit of data in the shift register is transmitted beginning with the LSB at each falling edge of the subsequent sampling clock. This operation is repeated until all 8 (or 7) bits of data are transmitted.
3. After sending the MSB, the parity bit (if EPR = 1) and the stop bit are transmitted in succession.
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 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.

Receive control

(1) Enabling receive operations

Use the receive-enable bit RXEN/FSIO_CTLx register 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 (initialized) by writing 0 to RXEN.

Note: Do not set RXEN 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[7:0]/FSIO_RXDx register. The older data is output first and cleared by reading.

The number of data in the receive data buffer can be checked by reading RXDNUM[1:0]/FSIO_STATUSx register. When RXDNUM[1:0] is 0, the buffer contains 0 or 1 data. When RXDNUM[1:0] is 1–3, the buffer contains 2–4 data.

Furthermore, RDBF/FSIO_STATUSx register 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.

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 by setting the interrupt control bits, the received data can be read by an interrupt processing routine.

In addition, a DMA can be invoked every time the received data is written to the receive-buffer, allowing the received data to be transferred successively to the specified memory location through DMA transfers. For details on how to control interrupts/DMA, refer to Section 20.9, “FSIO Interrupts and DMA.”

Figure 20.7.3.2 shows a receive timing chart in the asynchronous mode.

Example: Data length: 8 bits, Stop bit: 1 bit, Parity bit: Included

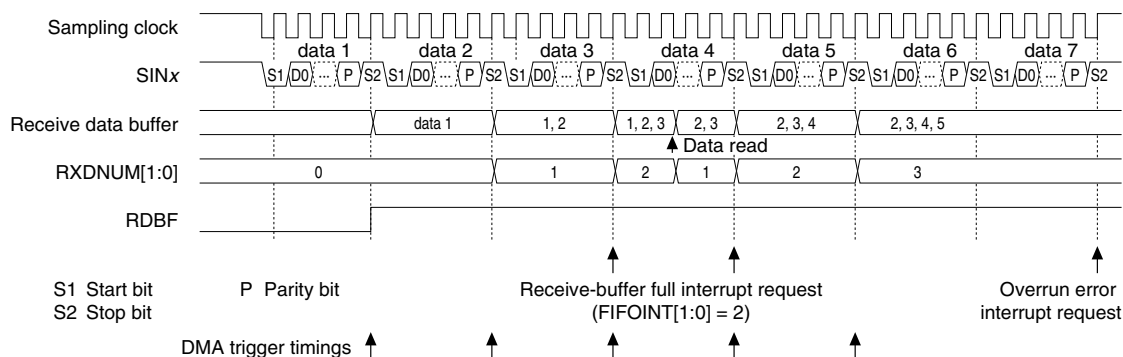


Figure 20.7.3.2 Receive Timing Chart in Asynchronous Mode

1. The serial interface starts sampling when the start bit is input ($SINx = \text{low}$).
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 $EPR = 1$).
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 $EPR = 1$).

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 control bits, the error can be processed using an interrupt processing routine. For details on receive error interrupts, refer to Section 20.9, “FSIO Interrupts and DMA.”

Parity error

If $EPR/FSIO_CTLx$ register 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 the $PMD/FSIO_CTLx$ register setting (odd or even parity).

If any nonconformity is found in this check, a parity error is assumed and the parity error flag $PER/FSIO_STATUSx$ register is set to 1.

Even when this error occurs, the received data in error is transferred to the receive data buffer and the receive operation is continued. However, the content of the received data for which a parity error is flagged cannot be guaranteed.

PER 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 $FER/FSIO_STATUSx$ register is set to 1.

20 GENERAL-PURPOSE SERIAL INTERFACE (FSIO)

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 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/FSIO_STATUS_x register is set to 1.

Even when this error occurs, the receive operation is continued.

OER 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 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 to 0.

20.8 IrDA Interface

20.8.1 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.

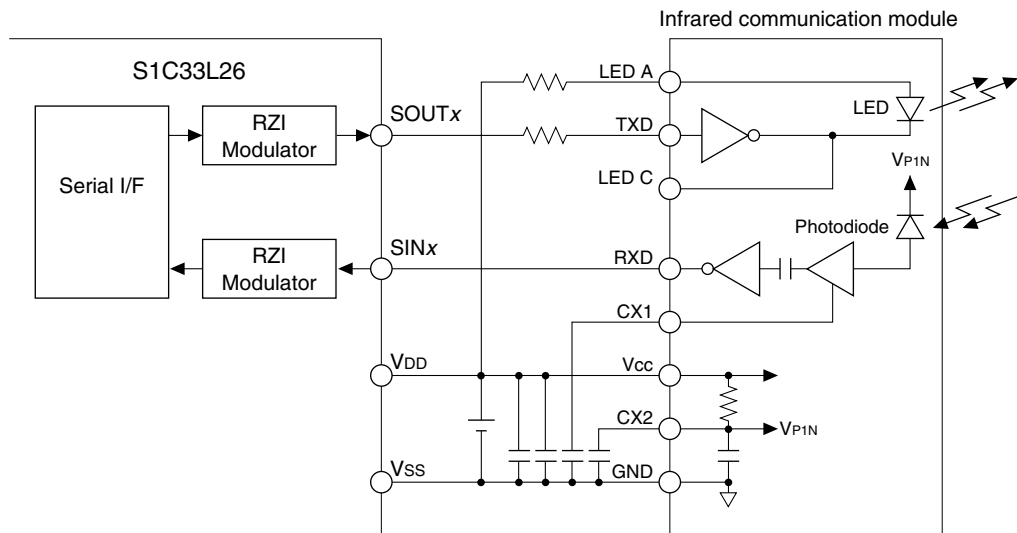


Figure 20.8.1.1 Configuration Example of IrDA Interface

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 20.7, “Asynchronous Interface,” for details on how to set and control the data formats and data transfers.

20.8.2 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 interrupts and DMA
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 20.7, “Asynchronous Interface.” For details on item 7, refer to Section 20.9, “FSIO Interrupts and DMA.”

Note: Before making these settings, always make sure the serial interface is inactive (TXEN/FSIO_CTLx register and RXEN/FSIO_CTLx register 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).

Selecting the IrDA interface function

To use the IrDA interface function, select it using IRMD[1:0]/FSIO_IRDAx register and then set the 8-bit (or 7-bit) asynchronous mode as the transfer mode.

Table 20.8.2.1 Setting of IrDA Interface

IRMD[1:0]	Interface mode
0x3	Setting prohibited (reserved)
0x2	IrDA 1.0 interface
0x1	Setting prohibited (reserved)
0x0	Normal interface

(Default: 0x0)

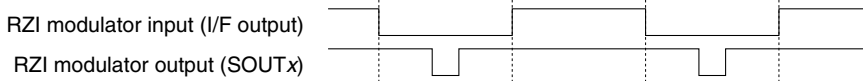
Setting the input/output logic

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 IRRL/FSIO_IRDAx register and IRTL/FSIO_IRDAx register, respectively.

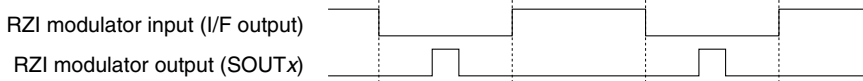
The logic of the input/output signal is inverted by writing 1 to IRRL/IRTL. Logic is not inverted if the bit is set to 0.

When transmitting

(1) IRTL = 0 (default)

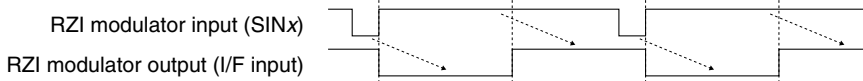


(2) IRTL = 1



When receiving

(1) IRRL = 0 (default)



(2) IRRL = 1

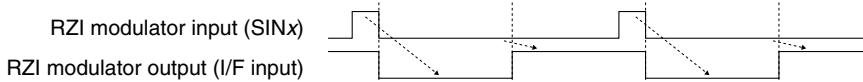


Figure 20.8.2.1 IRRL and IRTL Settings

20.8.3 Control and Operation of IrDA Interface

The transmit/receive procedures have been explained in the section on the asynchronous interface, so refer to Section 20.7.3, “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 SOUTx pin.

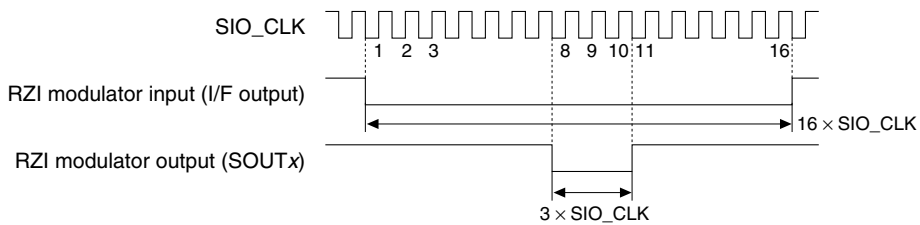


Figure 20.8.3.1 Data Modulation by RZI Circuit

When receiving

During data reception, the pulse width of the input signal from SINx is set to 16/3 before the signal is transferred to the serial interface.

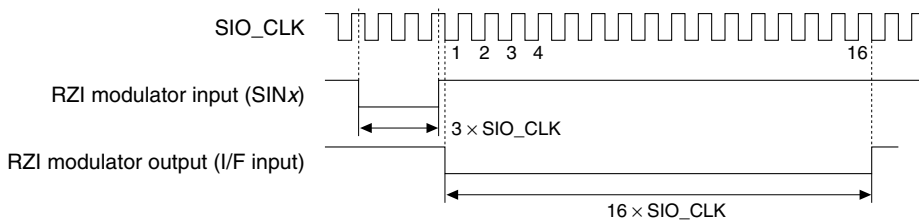


Figure 20.8.3.2 Demodulation by RZI Circuit

- Notes:**
- When using the IrDA interface, set the internal division ratio of the serial interface to 1/16 (DIVMD/FSIO_IRDAx register = 0). Do not set it to 1/8 (DIVMD = 1).
 - Although Figure 20.8.3.2 shows the input signal as a low pulse of a $3 \times$ SIO_CLK width, the RZI circuit recognizes low pulses by means of the signal edge (rising edge when IRRL = 0; falling edge when IRRL = 1). Note that noise may cause a malfunction.

20.9 FSIO Interrupts and DMA

This section describes the FSIO interrupts and DMA.

For more information on interrupt processing and DMA transfer, see the “Interrupt Controller (ITC)” chapter and the “DMA Controller (DMAC)” chapter, respectively.

20.9.1 Interrupts

FSIO includes a function for generating the following three different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

Each FSIO channel outputs one interrupt signal shared by the three above interrupt causes to the interrupt controller (ITC). Inspect the interrupt flags and error flags to determine the interrupt cause occurred.

Transmit buffer empty interrupt

To use this interrupt, set TDBE_IE/FSIO_INTE_x register to 1. If TDBE_IE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When transmit data written to the transmit data buffer is transferred to the shift register, the FSIO module sets TDBE_IF/FSIO_INTF_x register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (TDBE_IE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the TDBE_IF flag in the interrupt handler routine to determine whether the FSIO interrupt is attributable to a transmit buffer empty. If TDBE_IF is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

TDBE_IF is cleared by writing 0.

Note: When TDBE_IF is cleared and no data is written to the transmit data buffer, subsequent interrupt requests will not be issued even if the transmit data buffer is empty.

Receive buffer full interrupt

To use this interrupt, set RDBF_IE/FSIO_INTE_x register to 1. If RDBF_IE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the number of data specified with FIFOINT[1:0]/FSIO_IRDAx register (one data in standard mode) has been received in the receive data buffer, the FSIO module sets RDBF_IF/FSIO_INTF_x register to 1. If receive buffer full interrupts are enabled (RDBF_IE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the RDBF_IF flag in the interrupt handler routine to determine whether the FSIO interrupt is attributable to a receive buffer full. If RDBF_IF is 1, the received data can be read from the receive data buffer by the interrupt handler routine. However, be sure to check whether a receive error has occurred or not.

RDBF_IF is cleared by writing 0.

Note: Before RDBF_IF can be cleared by writing 0, be sure to read out the received data from the receive data buffer.

Receive error interrupt

To use this interrupt, set RERR_IE/FSIO_INTE_x register to 1. If RERR_IE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

20 GENERAL-PURPOSE SERIAL INTERFACE (FSIO)

The FSIO module sets an error flag (PER/FSIO_STATUSx register, FER/FSIO_STATUSx register, or OER/FSIO_STATUSx register) to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are enabled (RERR_IE = 1), an interrupt request is sent simultaneously to the ITC. If other interrupt conditions are satisfied, an interrupt occurs. You can inspect the PER, FER, and OER flags in the interrupt handler routine to determine whether the FSIO interrupt was caused by a receive error. If any of the error flags has the value 1, the interrupt handler routine will proceed with error recovery.

20.9.2 DMA Transfer

In transmit operations, a DMA trigger will be issued when a cause of transmit buffer empty interrupt occurs. In receive operations, a DMA trigger will be issued every time the received data is written to the receive data buffer. These DMA triggers allow continuous data transmission/reception through DMA transfer between memory and transmit/receive data buffers. DMA transfer can be performed without generating any FSIO interrupt. The following lists the DMAC channels that allow selection of an FSIO DMA trigger.

FSIO Ch.0 receive buffer full: DMAC Ch.2
 FSIO Ch.0 transmit buffer empty: DMAC Ch.3
 FSIO Ch.1 receive buffer full: DMAC Ch.4
 FSIO Ch.1 transmit buffer empty: DMAC Ch.5

For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

20.10 Control Register Details

Table 20.10.1 List of FSIO Registers

Address	Register name		Function
0x300700	FSIO_TXD0	FSIO Ch.0 Transmit Data Register	Transmit data
0x300701	FSIO_RXD0	FSIO Ch.0 Receive Data Register	Receive data
0x300702	FSIO_STATUS0	FSIO Ch.0 Status Register	Indicate transfer/error statuses
0x300703	FSIO_CTL0	FSIO Ch.0 Control Register	Set transfer mode and control data transfer
0x300704	FSIO_IRDA0	FSIO Ch.0 IrDA Register	Set IrDA conditions
0x300705	FSIO_BRTRUN0	FSIO Ch.0 Baud-rate Timer Control Register	Control baud-rate timer
0x300706	FSIO_BRTRDL0	FSIO Ch.0 Baud-rate Timer Reload Data L Register	Baud-rate timer initial count data
0x300707	FSIO_BRTRDH0	FSIO Ch.0 Baud-rate Timer Reload Data H Register	
0x300708	FSIO_BRTCDL0	FSIO Ch.0 Baud-rate Timer Count Data L Register	Baud-rate timer count data
0x300709	FSIO_BRTCDH0	FSIO Ch.0 Baud-rate Timer Count Data H Register	
0x30070a	FSIO_INTF0	FSIO Ch.0 Interrupt Flag Register	Indicate FSIO interrupt cause status
0x30070b	FSIO_INTE0	FSIO Ch.0 Interrupt Enable Register	Enable/disable FSIO interrupts
0x30070f	FSIO_ADV0	FSIO Ch.0 STD/ADV Mode Select Register	Select standard/advanced mode
0x300710	FSIO_TXD1	FSIO Ch.1 Transmit Data Register	Transmit data
0x300711	FSIO_RXD1	FSIO Ch.1 Receive Data Register	Receive data
0x300712	FSIO_STATUS1	FSIO Ch.1 Status Register	Indicate transfer/error statuses
0x300713	FSIO_CTL1	FSIO Ch.1 Control Register	Set transfer mode and control data transfer
0x300714	FSIO_IRDA1	FSIO Ch.1 IrDA Register	Set IrDA conditions
0x300715	FSIO_BRTRUN1	FSIO Ch.1 Baud-rate Timer Control Register	Control baud-rate timer
0x300716	FSIO_BRTRDL1	FSIO Ch.1 Baud-rate Timer Reload Data L Register	Baud-rate timer initial count data
0x300717	FSIO_BRTRDH1	FSIO Ch.1 Baud-rate Timer Reload Data H Register	
0x300718	FSIO_BRTCDL1	FSIO Ch.1 Baud-rate Timer Count Data L Register	Baud-rate timer count data
0x300719	FSIO_BRTCDH1	FSIO Ch.1 Baud-rate Timer Count Data H Register	
0x30071a	FSIO_INTF1	FSIO Ch.1 Interrupt Flag Register	Indicate FSIO interrupt cause status
0x30071b	FSIO_INTE1	FSIO Ch.1 Interrupt Enable Register	Enable/disable FSIO interrupts
0x30071f	FSIO_ADV1	FSIO Ch.1 STD/ADV Mode Select Register	Select standard/advanced mode

The FSIO registers are described in detail below. These are 8-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

FSIO Ch.x Transmit Data Registers (FSIO_TXDx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.x Transmit Data Register (FSIO_TXDx)	0x300700	D7-0	TXD[7:0]	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	X	R/W	7-bit asynchronous mode does not use TXD7.
	0x300710 (8 bits)							

D[7:0] TXDx[7:0]: Transmit Data Bits

Sets transmit data. (Default: indeterminate)

When data is written to this register (transmit data buffer) after 1 is written to TXEN/FSIO_CTLx register, a transmit operation begins. The data written to TXD[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, TXD7 (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.

FSIO Ch.x Receive Data Registers (FSIO_RXDx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.x Receive Data Register (FSIO_RXDx)	0x300701	D7-0	RXD[7:0]	Receive data RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	X	R	7-bit asynchronous mode does not use RXD7 (fixed at 0).
	0x300711 (8 bits)							

D[7:0] RXDx[7:0]: 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/FSIO_STATUSx register is provided to indicate that it is necessary to read the receive data buffer. This flag is set to 1 when the receive data buffer contains one or more received data, and is reset to 0 when the receive data buffer becomes empty by reading all the received data.

When the receive data buffer has received the number of data specified with FIFOINT[1:0]/FSIO_IRDAx register (one data in standard mode), a cause of receive buffer full interrupt occurs.

In 7-bit asynchronous mode, 0 is stored in RXD7.

The serial data input from the SINx 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)

FSIO Ch.x Status Registers (FSIO_STATUSx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FSIO Ch.x Status Register (FSIO_STATUSx)	0x300702 0x300712 (8 bits)	D7-6	RXDNUM [1:0]	Receive FIFO data count	RXDNUM[1:0]	Number of data	0x0	R	
					0x3	4			
					0x2	3			
					0x1	2			
					0x0	1 or 0			
	D5	TEND	Transmit status flag	1 Busy	0 End/Idle	0	R	Reset by writing 0.	
	D4	FER	Framing error flag	1 Error	0 Normal	0	R/W		
	D3	PER	Parity error flag	1 Error	0 Normal	0	R/W		
	D2	OER	Overrun error flag	1 Error	0 Normal	0	R/W		
	D1	TDBE	Transmit data buffer empty flag	1 Empty	0 Full	1	R		
D0	RDBF	Receive data buffer status flag	1 Contained	0 Not contained	0	R			

D[7:6] RXDNUM[1:0]: Receive FIFO Data Count Bits

Indicates the number of data in the receive data buffer (FIFO) that have not been read.

Table 20.10.2 Number of Receive Data

RXDNUM[1:0]	Number of data
0x3	4
0x2	3
0x1	2
0x0	1 or 0

(Default: 0x0)

When RXDNUM[1:0] is 0, it indicates that the receive data buffer contains 0 or 1 received data. When RXDNUM[1:0] is 1 to 3, it indicates that the receive data buffer contains 2 to 4 received data.

D5 TEND: Transmit Status Flag Bit

Indicates the transmission status.

1 (R): Busy (during transmitting)

0 (R): End/idle (default)

TEND goes 1 after the first bit is shifted out from the shift register and goes 0 after the last bit is shifted out. When data is transmitted successively in clock-synchronized master mode or asynchronous mode, TEND maintains 1 until all data is shifted out (see Figure 20.6.3.1 and Figure 20.7.3.1). In clock-synchronized slave mode, TEND goes 0 every time 1-byte data is shifted out (see Figure 20.6.3.2).

Note: TEND goes 0 at the falling edge of SCLKx to indicate that all the transmit data bits in the transmit shift register are shifted out. Be aware that there is a half SCLKx cycle interval between setting TEND to 0 and latching the last bit by the receiver.

D4 FER: Framing Error Flag Bit

Indicates whether a framing error has occurred.

1 (R): An error occurred

0 (R): No error occurred (default)

1 (W): Has no effect

0 (W): Reset to 0

FER is an error flag indicating whether a framing error has occurred or not. When an error has occurred, it is set to 1. A framing error occurs when data with a stop bit = 0 is received in asynchronous mode.

FER is reset by writing 0.

D3 PER: Parity Error Flag Bit

Indicates whether a parity error has occurred.

1 (R): An error occurred

0 (R): No error occurred (default)

1 (W): Has no effect

0 (W): Reset to 0

PER is an error flag indicating whether a parity error has occurred or not. When an error has occurred, it is set to 1. Parity checks are valid only in asynchronous mode with EPR set to 1 (parity added). This check is performed when the received data is transferred from the shift register to the receive data buffer.

PER is reset by writing 0.

D2 OER: Overrun Error Flag Bit

Indicates whether an overrun error has occurred.

1 (R): An error occurred

0 (R): No error occurred (default)

1 (W): Has no effect

0 (W): Reset to 0

OER is an error flag indicating whether an overrun error has occurred or not. 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.

OER is reset by writing 0.

D1 **TDBE: Transmit Data Buffer Empty Flag Bit**

Indicates the status of the transmit data buffer.

1 (R): Not full (default)

0 (R): Buffer full

TDBE 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 **RDBF: Receive Data Buffer Status Flag Bit**

Indicates the status of the receive data buffer.

1 (R): Not empty (contains received data)

0 (R): Buffer empty (default)

RDBF 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.

FSIO Ch.x Control Registers (FSIO_CTLx)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
FSIO Ch.x Control Register (FSIO_CTLx)	0x300703 0x300713 (8 bits)	D7	TXEN	Transmit enable	1	Enable	0 Disable	0	R/W	
		D6	RXEN	Receive enable	1	Enable	0 Disable	0	R/W	
		D5	EPR	Parity enable	1	With parity	0 No parity	0	R/W	Valid only in
		D4	PMD	Parity mode select	1	Odd	0 Even	0	R/W	asynchronous
		D3	STPB	Stop bit select	1	2 bits	0 1 bit	0	R/W	mode.
		D2	SSCK	Input clock select	1	SCLK	0 Internal	0	R/W	
		D1-0	SMD[1:0]	Transfer mode select		SMD[1:0]	Transfer mode	0x0	R/W	
					0x3	8-bit async				
					0x2	7-bit async				
					0x1	Clk sync slave				
					0x0	Clk sync master				

D7 **TXEN: Transmit Enable Bit**

Enables transmit operations.

1 (R/W): Transmit enabled

0 (R/W): Transmit disabled (default)

When TXEN for a channel is set to 1, the channel is enabled for transmit operations. When TXEN is set to 0, the channel is disabled for transmit operations.

Always make sure TXEN = 0 before setting the transfer mode and other conditions.

Writing 0 to TXEN clears the transmit data buffer (FIFO) as well as disabling transmit operations.

D6 **RXEN: Receive Enable Bit**

Enables receive operations.

1 (R/W): Receive enabled

0 (R/W): Receive disabled (default)

When RXEN for a channel is set to 1, the channel is enabled for receive operations. When RXEN is set to 0, the channel is disabled for receive operations.

Always make sure RXEN = 0 before setting the transfer mode and other conditions.

Writing 0 to RXEN clears the receive data buffer (FIFO) as well as disabling receive operations.

D5 **EPR: Parity Enable Bit**

Selects a parity function for asynchronous transfer.

1 (R/W): Parity added

0 (R/W): No parity added (default)

20 GENERAL-PURPOSE SERIAL INTERFACE (FSIO)

EPR 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 is set to 1, the receive data is checked for parity. A parity bit is automatically added to the transmit data. When EPR is set to 0, parity is not checked and no parity bit is added.

EPR is only effective in asynchronous mode. Settings of EPR have no effect in clock-synchronized mode.

D4 **PMD: Parity Mode Select Bit**

Selects an odd or even parity for asynchronous transfer.

1 (R/W): Odd parity

0 (R/W): Even parity (default)

Odd parity is selected by writing 1 to PMD, 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 is set to 1. If EPR = 0, PMD is ineffective.

D3 **STPB: Stop Bit Select Bit**

Selects a stop-bit length for asynchronous transfer.

1 (R/W): 2 bits

0 (R/W): 1 bit (default)

STPB is only valid in asynchronous mode. Two stop bits are selected by writing 1 to STPB, and one stop bit is selected by writing 0. The start bit is fixed at 1 bit.

Settings of STPB are ignored in clock-synchronized mode.

D2 **SSCK: Input Clock Select Bit**

Selects the clock source for asynchronous transfer.

1 (R/W): SCLK_x (external clock)

0 (R/W): Internal clock (default)

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[1:0]: Transfer Mode Select Bits**

Sets the transfer mode of the serial interface as shown in the table below.

Table 20.10.3 Setting of Transfer Mode

SMD[1:0]	Transfer mode
0x3	8-bit asynchronous mode
0x2	7-bit asynchronous mode
0x1	Clock-synchronized slave mode
0x0	Clock-synchronized master mode

(Default: 0x0)

SMD[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.

FSIO Ch.x IrDA Registers (FSIO_IRDA_x)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
FSIO Ch.x IrDA Register (FSIO_IRDA _x)	0x300704 0x300714 (8 bits)	D7	SRDYCTL	#SRDY control	1 High mask	0 Normal	0	R/W	Writing is disabled when SLOADV = 0.	
		D6–5	FIFOINT [1:0]	Receive buffer full interrupt timing	FIFOINT[1:0]	Receive level	0x0	R/W		
					0x3	4				
					0x2	3				
					0x1	2				
					0x0	1				
		D4	DIVMD	Async clock division ratio	1 1/8	0 1/16	0	R/W		
		D3	IRTL	IrDA I/F output logic inversion	1 Inverted	0 Direct	0	R/W		Valid only in async mode.
		D2	IRRL	IrDA I/F input logic inversion	1 Inverted	0 Direct	0	R/W		
		D1–0	IRMD[1:0]	Interface mode select	IRMD[1:0]	I/F mode	0x0	R/W		
0x3	reserved									
0x2	IrDA 1.0 reserved									
0x1	reserved									
			0x0	General I/F						

D7 SRDYCTL: #SRDY Control Bit

Selects a control method for the #SRDY_x signal.

1 (R/W): High mask mode

0 (R/W): Normal output (default)

When SRDYCTL is set to 0, the #SRDY_x signal is controlled normally and indicates ready to receive even if the receive data buffer is full. When SRDYCTL is set to 1, high-mask mode is selected. The following shows the #SRDY_x controls in clock-synchronized slave mode and master mode:

Clock-synchronizes slave mode

When 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.

Clock-synchronized master mode

When the receive data buffer is full, the #SRDY_x signal (low) from the slave device is ignored and the serial interface stops outputting the SCLK_x 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, this bit is ignored as it does not use the #SRDY_x signal.

Note: This bit can be rewritten only when SIOADV/FSIO_ADV_x register is set to 1 (advanced mode).

D[6:5] FIFOINT[1:0]: Receive Buffer Full Interrupt Timing Bits

Sets the number of data in the receive data buffer to generate a receive-buffer full interrupt.

Table 20.10.4 Number of Receive Data Buffer

FIFOINT[1:0]	Receive level
0x3	4
0x2	3
0x1	2
0x0	1

(Default: 0x0)

Writing 0–3 to FIFOINT[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 flag RDBF_IF/FSIO_INTF_x register is set to 1.

Note: This bit can be rewritten only when SIOADV is set to 1 (advanced mode).

D4 DIVMD: Async Clock Division Ratio Bit

Selects the division ratio for the sampling clock.

1 (R/W): 1/8

0 (R/W): 1/16 (default)

Select the division ratio to generate the sampling clock for asynchronous transfers. When DIVMD 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 SCLK_x) by dividing it by 8. When DIVMD is set to 0, the input clock is divided by 16.

D3 IRTL: IrDA I/F Output Logic Inversion Bit

Inverts the logic of the IrDA output signal.

1 (R/W): Inverted

0 (R/W): Not inverted (default)

When using the IrDA interface, set the logic of the SOUT_x output signal to suit the infrared-ray communication circuit that is connected external to the chip. If IRTL is set to 1, a high pulse is output when the output data = 0 (held low-level when the output data = 1). If IRTL is set to 0, a low pulse is output when the output data = 0 (held high-level when the output data = 1).

D2 IRRL: IrDA I/F Input Logic Inversion Bit

Inverts the logic of the IrDA input signal.

1 (R/W): Inverted

0 (R/W): Not inverted (default)

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 IRRL is set to 1, a high pulse is input as a logic 0. If IRRL is set to 0, a low pulse is input as a logic 0.

D[1:0] IRMD[1:0]: Interface Mode Select Bits

Selects the IrDA interface function.

Table 20.10.5 IrDA Interface Setting

IRMD[1:0]	Interface mode
0x3	Setting prohibited (reserved)
0x2	IrDA 1.0 interface
0x1	Setting prohibited (reserved)
0x0	Normal interface

(Default: 0x0)

When using the IrDA interface function, write 0x2 to IRMD[1:0] while setting to asynchronous mode for the transfer mode. If the IrDA interface function is not to be used, write 0x0 to IRMD[1:0].

Note: This selection must always be performed before the transfer mode and other conditions are set.

FSIO Ch.x Baud-rate Timer Control Registers (FSIO_BRTRUNx)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.x Baud-rate Timer Control Register (FSIO_ BRTRUNx)	0x300705	D7-1	–	reserved	1 Run 0 Stop	–	–	0 when being read.
	0x300715 (8 bits)		D0	BRTRUN		Baud-rate timer run/stop control	0	

D[7:1] Reserved**D0 BRTRUN: 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 (BRTRD[11:0]/FSIO_BRTRDLx and FSIO_BRTRDLx registers) to its counter and starts counting down when 1 is written to BRTRUN. The baud-rate timer stops counting when 0 is written to BRTRUN.

FSIO Ch.x Baud-rate Timer Reload Data L Registers (FSIO_BRTRDLx)**FSIO Ch.x Baud-rate Timer Reload Data H Registers (FSIO_BRTRDHx)**

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.x Baud-rate Timer Reload Data L Register (FSIO_ BRTRDLx)	0x300706	D7-0	BRTRD[7:0]	Baud-rate timer reload data [7:0]	0x0 to 0xff (BRTRD[11:0] = 0x0 to 0xffff)	0x0	R/W	
	0x300716 (8 bits)		–	reserved				
FSIO Ch.x Baud-rate Timer Reload Data H Register (FSIO_ BRTRDHx)	0x300707	D3-0	–	reserved	0x0 to 0xf (BRTRD[11:0] = 0x0 to 0xffff)	0x0	R/W	
	0x300717 (8 bits)		BRTRD [11:8]	Baud-rate timer reload data [11:8]				

D[7:0]/FSIO_BRTRDLx, D[3:0]/FSIO_BRTRDHx**BRTRD[11:0]: Baud-rate Timer Reload Data [11:0]**

Sets the initial counter value of the baud-rate timer. (Default: 0x0)

The reload data set in these registers are loaded into the counter, and the counter starts counting down beginning with the set 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 BRTRUN/FSIO_BRTRUN_x register, or when data is automatically reloaded upon counter underflow.

FSIO Ch.x Baud-rate Timer Count Data L Registers (FSIO_BRTCDL_x)

FSIO Ch.x Baud-rate Timer Count Data H Registers (FSIO_BRTCDH_x)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.x Baud-rate Timer Count Data L Register (FSIO_ BRTCDL _x)	0x300708 0x300718 (8 bits)	D7-0	BRTCD[7:0]	Baud-rate timer count data [7:0]	0x0 to 0xff (BRTCD[11:0] = 0x0 to 0xffff)	0x0	R	
FSIO Ch.x Baud-rate Timer Count Data H Register (FSIO_ BRTCDH _x)	0x300709 0x300719 (8 bits)	D7-4	—	reserved	—	—	—	0 when being read.
		D3-0	BRTCD [11:8]	Baud-rate timer count data [11:8]	0x0 to 0xf (BRTCD[11:0] = 0x0 to 0xffff)	0x0	R	

D[7:0]/FSIO_BRTCDL_x, D[3:0]/FSIO_BRTCDH_x

BRTCD_x[11:0]: Baud-rate Timer Count Data [11:0]

The baud-rate timer data can be read out from these registers. (Default: 0x0)

These registers function as a buffer that retain the counter data when read out, enabling the data to be read out at any time.

FSIO Ch.x Interrupt Flag Registers (FSIO_INTF_x)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FSIO Ch.x Interrupt Flag Register (FSIO_INTF _x)	0x30070a 0x30071a (8 bits)	D7-2	—	reserved	—	—	—	0 when being read.	
		D1	TDBE_IF	Transmit data buffer empty int. flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 0.
		D0	RDBF_IF	Receive data buffer full int. flag			0	R/W	

D[7:2] Reserved

D1 TDBE_IF: Transmit Data Buffer Empty Interrupt Flag Bit

Indicates whether the cause of transmit data buffer empty interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Ignored
- 0 (W): Flag is reset

TDBE_IF is set to 1 when the transmit data written to the transmit data buffer is transferred to the shift register, indicating that the next transmit data can be written to. At the same time a transmit data buffer empty interrupt request is sent to the ITC if TDBE_IE/FSIO_INTE_x register is 1. TDBE_IF is reset by writing 0.

D0 RDBF_IF: Receive Data Buffer Full Interrupt Flag Bit

Indicates whether the cause of receive data buffer full interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Ignored
- 0 (W): Flag is reset

RDBF_IF is set to 1 when the number of data specified with FIFOINT[1:0]/FSIO_IRDA_x register (one data in standard mode) has been received in the receive data buffer, indicating that received data can be read. At the same time a receive data buffer full interrupt request is sent to the ITC if RDBF_IE/FSIO_INTE_x register is 1. RDBF_IF is reset by writing 0.

FSIO Ch.x Interrupt Enable Registers (FSIO_INTE_x)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.x Interrupt Enable Register (FSIO_INTE _x)	0x30070b 0x30071b (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	RERR_IE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	TDBE_IE	Transmit data buffer empty interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	RDBF_IE	Receive data buffer full int. enable	1 Enable 0 Disable	0	R/W	

D[7:3] Reserved

D2 RERR_IE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when a receive error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process receive errors using interrupts.

D1 TDBE_IE: Transmit Data Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC when data written to the transmit data buffer is sent to the shift register (i.e. when data transmission begins).

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to write data to the transmit data buffer using interrupts.

D0 RDBF_IE: Receive Data Buffer Full Interrupt Enable Bit

Enables interrupt requests to the ITC when received data is loaded to the receive data buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

FSIO Ch.x STD/ADV Mode Select Registers (FSIO_ADV_x)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.x STD/ ADV Mode Select Register (FSIO_ADV _x)	0x30070f 0x30071f (8 bits)	D7-1	–	reserved	–	–	–	Writing 1 not allowed.
		D0	SIOADV	Standard/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 S1C33L26 is extended from that of the C33 STD models. The S1C33L26 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 20.10.6 shows differences between standard mode and advanced mode.

Table 20.10.6 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 (SRDYCTL and FIFOINT[1:0]) for the extended functions are enabled to write after this setting.

21 I²S

21.1 I²S Module Overview

The S1C33L26 has a built-in I²S module that outputs PCM data in the I²S (Inter-IC Sound) format. An audio output circuit can be simply configured by connecting external devices such as an audio DAC to the I²S bus.

The following shows the features of the I²S module:

- Operates as an I²S master device.
- Generates the bit clock, word-select clock, and master clock.
- Supports 16-bit PCM data resolution.
- A 16-byte transmit FIFO (16 bits × 2 channels × 4) is included.
- Stereo, mono (L and R), and mute modes are software selectable.
- FIFO data empty (half empty, whole empty, or one empty) can issue an interrupt request.
- FIFO one empty interrupt cause can invoke DMA.
- Clock polarity is software configurable.
- Data shift direction (MSB first/LSB first) is software selectable.
- Supports I²S mode, left justified mode, and right justified mode.

Figure 21.1.1 shows the configuration of the I²S module.

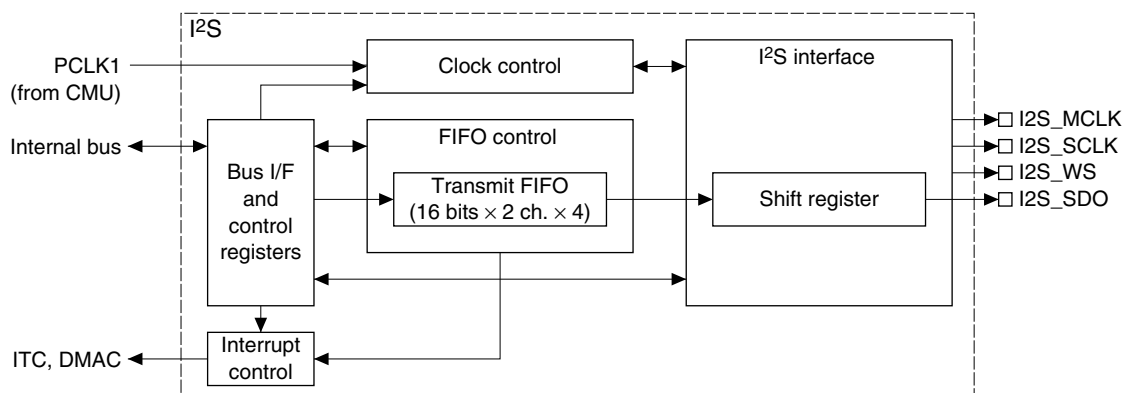


Figure 21.1.1 I²S Module Configuration

21.2 I²S Output Pins

Table 21.2.1 lists the I²S pins.

Table 21.2.1 List of I²S Pins

Pin name	I/O	Qty	Function
I2S_SDO	O	1	I ² S data output pin Outputs serial PCM data.
I2S_WS	O	1	I ² S word-select signal (LRCLK) output pin Outputs the word-select signal that indicates the channel (L or R) of the data being output.
I2S_SCLK	O	1	I ² S synchronous clock (bit clock) output pin Outputs the synchronous clock (bit clock) for serial data.
I2S_MCLK	O	1	I ² S master clock output pin Outputs the I ² S master clock.

The I²S output pins (I2S_SDO, I2S_WS, I2S_SCLK, I2S_MCLK) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as I²S output pins.

For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

21.3 I²S Module Operating Clock

The I²S module uses PCLK1 generated by the CMU as the operating clock.

For more information on the PCLK1 supply control, see the “Clock Management Unit (CMU)” chapter.

21.4 Setting the I²S Module

When performing data transfers via the I²S bus, the following settings must be made before data transfer is actually begun:

1. Setting the output pins
2. Setting the I²S interface clocks
3. Setting the data format and timing
4. Setting interrupt or DMA conditions (see Section 21.6.)

The following describes the settings.

Note: Always make sure the I²S module is not started (I2SSTART/I2S_START register = 0) before these settings are made. A change of settings during operation may cause a malfunction.

Setting the output pins

Configure the port function select bits to enable the I²S output functions. For details of pin functions and how to switch over, see the “I/O Ports (GPIO)” chapter.

Setting the I²S interface clocks

The I²S module outputs the following three clocks:

1. I2S_MCLK (master clock)
2. I2S_SCLK (bit clock)
3. I2S_WS (word-select clock)

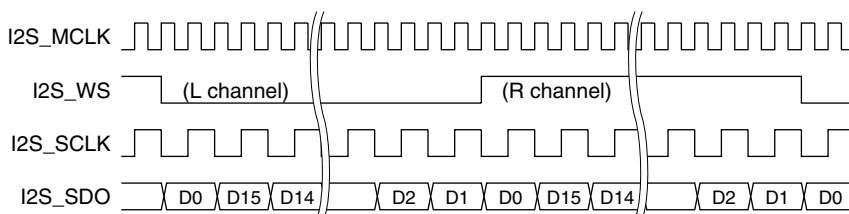


Figure 21.4.1 I²S Interface Clocks

The following shows the configurable clock conditions and their control bits. For more information on clock setting, see Section 21.8, “Setting the I²S Clocks.”

Division ratio for I2S_MCLK (master clock)

The I²S module generates I2S_MCLK to be output from the I2S_MCLK pin by dividing the PCLK1 generated by the CMU. Specify the division ratio using MCLKDIV[5:0]/I2S_DV_MCLK register.

Table 21.4.1 I2S_MCLK (Master Clock) Settings

MCLKDIV[5:0]	PCLK1 division ratio
0x3f	1/64
0x3e	1/63
0x3d	1/62
:	:
0x2	1/3
0x1	1/2
0x0	1/1

(Default: 0x0)

Division ratio for I2S_SCLK (bit clock)

The I²S module generates the bit clock to be output from the I2S_SCLK pin by dividing PCLK1. Specify the division ratio using BCLKDIV[7:0]/I2S_DV_AUDIO_CLK register.

Table 21.4.2 I2S_SCLK (Bit Clock) Settings

BCLKDIV[7:0]	PCLK1 division ratio
0xff	1/512
0xfe	1/510
0xfd	1/508
:	:
0x2	1/6
0x1	1/4
0x0	1/2

(Default: 0x0)

The I²S bit clock frequency is calculated as below.

$$f_{i2s_sclk} = \frac{f_{PCLK1}}{(BCLKDIV + 1) \times 2} \text{ [Hz]}$$

f_{i2s_sclk} : I²S bit clock frequency [Hz]

f_{PCLK1} : PCLK1 clock frequency [Hz]

BCLKDIV: BCLKDIV[7:0] set value (0x0–0xff)

Sample clock (I2S_WS) period

The I²S generates the sample clock (word-select clock) to be output from the I2S_WS pin by counting the bit clock configured with BCLKDIV[7:0]. Specify the half cycle (a high or low level period) of the I2S_WS clock with the number of bit clock cycles using WSCLKCYC[4:0]/I2S_DV_AUDIO_CLK register.

Table 21.4.3 Sample Clock Period Settings

WSCLKCYC[4:0]	Sample clock period (number of bit clock cycles)
0x1f–0x11	Reserved
0x10	32 clocks
0xf	31 clocks
0xe	30 clocks
0xd	29 clocks
0xc	28 clocks
0xb	27 clocks
0xa	26 clocks
0x9	25 clocks
0x8	24 clocks
0x7	23 clocks
0x6	22 clocks
0x5	21 clocks
0x4	20 clocks
0x3	19 clocks
0x2	18 clocks
0x1	17 clocks
0x0	16 clocks

(Default: 0x0)

The sampling clock frequency is calculated as below.

$$f_s = \frac{f_{I2S_SCLK}}{n \times 2} \text{ [Hz]}$$

fs: Sampling clock frequency [Hz]

f_{I2S_SCLK}: Bit clock frequency [Hz] (See Table 21.4.2.)

n: Number of bit clocks selected by WSCLKCYC[4:0] (See Table 21.4.3.)

Note: The value to be set to the WSCLKCYC[4:0] is not the number of audio data bits, but the number of bit clock cycles that is used to adjust the sample clock period. It must be equal to or greater than the number of audio data bits (16 bits).

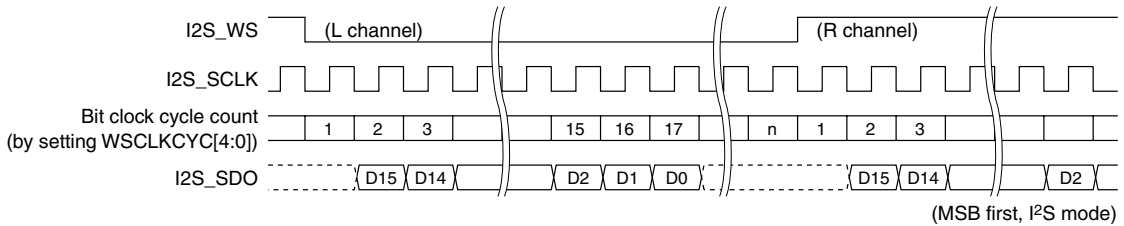


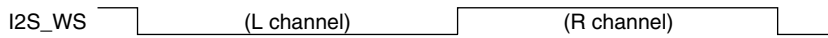
Figure 21.4.2 Sample Clock Period

Selecting the word clock mode

The I2S_WS signal represents the current output channel (L or R) with its level (low or high).

Use WCLKMD/I2S_CTL register to select the relationship between the signal level and the L/R channel.

WCLKMD = 0 (default)



WCLKMD = 1

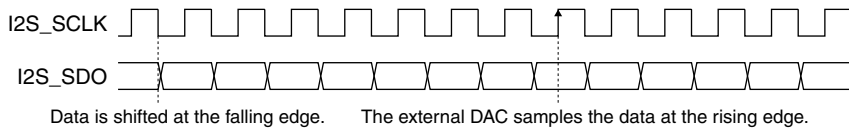


Figure 21.4.3 Word Clock Mode

I2S_SCLK (bit clock) polarity

Use BCLKPOL/I2S_CTL register to select the bit clock polarity.

BCLKPOL = 0 (default)



BCLKPOL = 1

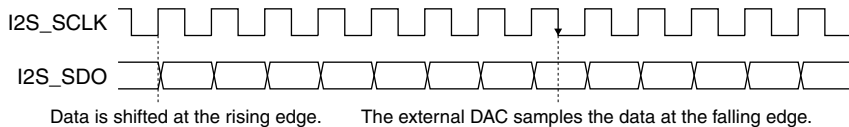


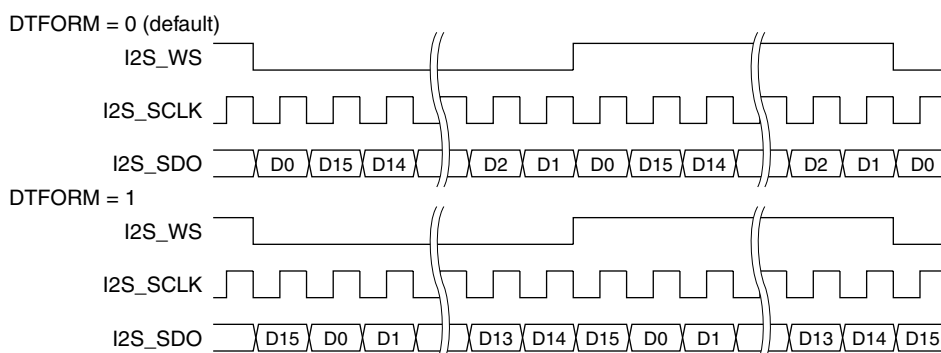
Figure 21.4.4 Bit Clock Polarity

Setting the output data format and timing

Data format (MSB first/LSB first)

Use DTFORM/I2S_CTL register to select either MSB first or LSB first as the data output direction.

Setting DTFORM to 0 (default) selects MSB first and setting 1 selects LSB first.

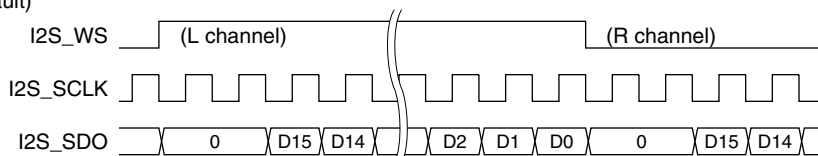
Figure 21.4.5 Output Data Format (Example in I²S Mode)

Signed/unsigned format

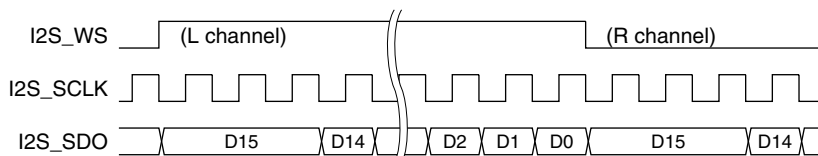
When right justified mode is selected as the data output timing condition, output data can be configured to the signed or unsigned format using DTSIGN/I2S_CTL register.

Setting DTSIGN to 0 (default) selects the unsigned format. The high-order bits that exceed the valid data size are set to 0. Setting 1 selects the signed format. The high-order bits that exceed the valid data size are set to the sign bit value (D15) of the valid data.

DTSIGN = 0 (default)



DTSIGN = 1



(MSB first, right justified mode, number of bit clock cycles = 18)

Figure 21.4.6 Unsigned and Signed Format

This setting is effective only in right justified mode. Set DTSIGN to 0 when another data output timing mode is selected.

Data output timing

Use DTTMG[1:0]/I2S_CTL register to select the data output timing.

Table 21.4.4 Data Output Timing

DTTMG[1:0]	Data output timing mode
0x3	Reserved
0x2	Right justified mode
0x1	Left justified mode
0x0	I ² S mode

(Default: 0x0)

When DTTMG[1:0] is set to 0x0 (default), I²S mode is selected. In this mode, the first bit of each data is output after one I2S_SCLK clock delay from the I2S_WS signal edge.

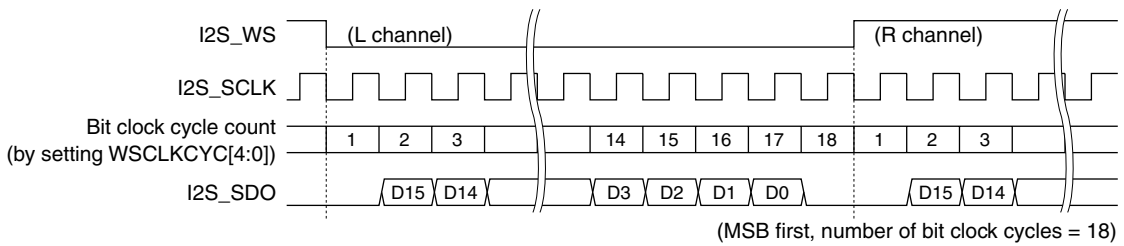


Figure 21.4.7 Data Output Timing 1 (I²S Mode)

When DTTMG[1:0] is set to 0x1, left justified mode is selected. In this mode, each data output starts at the I2S_WS signal edge.

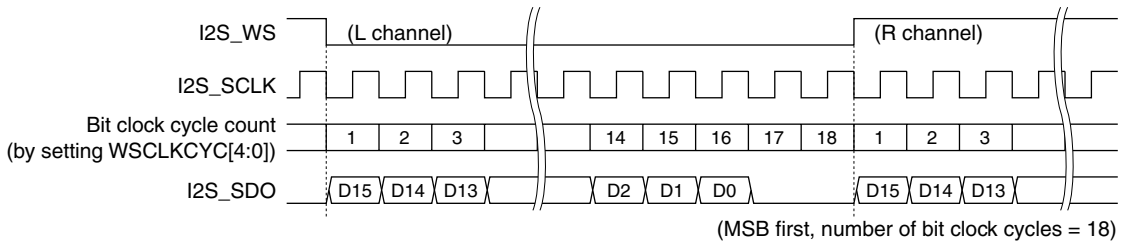


Figure 21.4.8 Data Output Timing 2 (Left Justified Mode)

When DTTMG[1:0] is set to 0x2, right justified mode is selected. In this mode, output data is right justified to the I2S_WS signal edge.

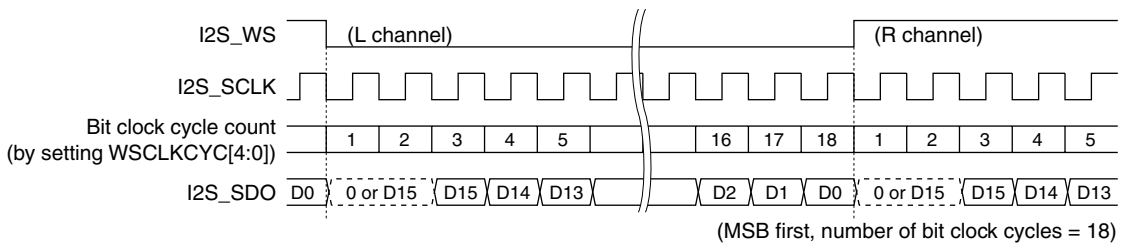


Figure 21.4.9 Data Output Timing 3 (Right Justified Mode)

Note: When using right justified mode, the number of bit clock cycles (sample clock period) must be equal to or greater than [Data bit size + 2].

21.5 Data Output Control

The following shows audio data output procedure:

1. Set up the I²S conditions as described in the previous section.
2. Set up the interrupt or DMA conditions as described in Section 21.6.
3. Set the output channel mode using CHMD[1:0]/I2S_CTL register.

Table 21.5.1 Output Channel Mode Selection

CHMD[1:0]	Output channel mode	L channel	R channel
0x3	Mute	0	0
0x2	Mono (L)	Data output	0
0x1	Mono (R)	0	Data output
0x0	Stereo	Data output	Data output

(Default: 0x0)

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.

4. Set I2SOUTEN/I2S_CTL register.

Write 1 to I2SOUTEN to enable I²S output. When I2SOUTEN is set to 1, all the output pins enter standby status.

Write 0 to I2SOUTEN to disable I²S output. When I2SOUTEN is set to 0, the I2S_MCLK pin still keeps standby status, the I2S_WS pin is fixed at 0. The I2S_SDO pin is left unchanged. The I2S_SCLK pin is fixed at 0 (when BCLKPOL/I2S_CTL register = 0) or 1 (when BCLKPOL = 1).

I2SOUTEN can be altered even if data is being output. This enables the I²S module to pause or resume shifting data (data output).

5. Write 1 to I2SSTART/I2S_START register to start output.

When I2SSTART is set to 1, the I²S module issues an interrupt or DMA request. However, the I²S module idles until the FIFO becomes full (16 bits × 2 channels (L & R) × 4) according to the interrupt mode or DMA mode (as described in Step 6 below and Section 21.6), then loads data (L & R) in the FIFO to the shift register and starts serial output in sync with the I2S_WS signal.

The data in the shift register is shifted at the I2S_SCLK clock edge and is output from the L channel first. When an output of one data (L & R) has finished, the next data is read out from the FIFO and the same operation repeats.

When the number of data according to the interrupt conditions has been read out from the FIFO, an interrupt or DMA request is generated.

When half empty interrupts are enabled, the I²S module generates an interrupt after two stereo data has been read out from the FIFO. In this case, write the next two stereo data (16 bits × 2 channels (L & R) × 2) to the FIFO in the interrupt handler.

When whole empty interrupts are enabled, the I²S module generates an interrupt after all data (four stereo data) has been read out from the FIFO. In this case, write the next four stereo data (16 bits × 2 channels (L & R) × 4) to the FIFO in the interrupt handler.

When one empty interrupts are enabled, the I²S module generates an interrupt after one stereo data has been read out from the FIFO. In this case, write the next one stereo data (16 bits × 2 channels (L & R) × 1) to the FIFO in the interrupt handler. This one empty interrupt cause can also be used to invoke a DMA transfer.

6. Write audio data to the FIFO.

There are two ways to write data.

6.1) Using interrupts

The I²S module includes three different types of interrupts.

• I²S FIFO half empty interrupt

When the half empty interrupt is enabled, the I²S module generates an interrupt after two stereo data has been read out from the FIFO. However, the I²S module continues shifting out the remaining buffer data until the FIFO becomes absolutely empty.

We suggest filling the FIFO with two groups of stereo data (16 bits × 2 channels (L & R) × 2) at once in the half empty interrupt handler.

Note: If the handler fills the FIFO with more than two groups of data, it may overwrite the remaining data in the FIFO. If the handler fills the FIFO with one group of data, a half empty interrupt will be issued again after one remaining data is sent to the shift register. Therefore, the FIFO should be filled with two groups of data in the half empty interrupt handler.

• I²S FIFO whole empty interrupt

When the whole empty interrupt is enabled, the I²S module generates an interrupt after all data (four stereo data) has been read out from the FIFO. In this case, the FIFO becomes absolutely empty. If the I²S module sends the current data completely and the FIFO is still empty, it stops shifting out FIFO data until the FIFO becomes full again.

Be sure to write four stereo data (16 bits × 2 channels (L & R) × 4) to the FIFO at once in the whole empty interrupt handler, otherwise the I²S module continues idle status.

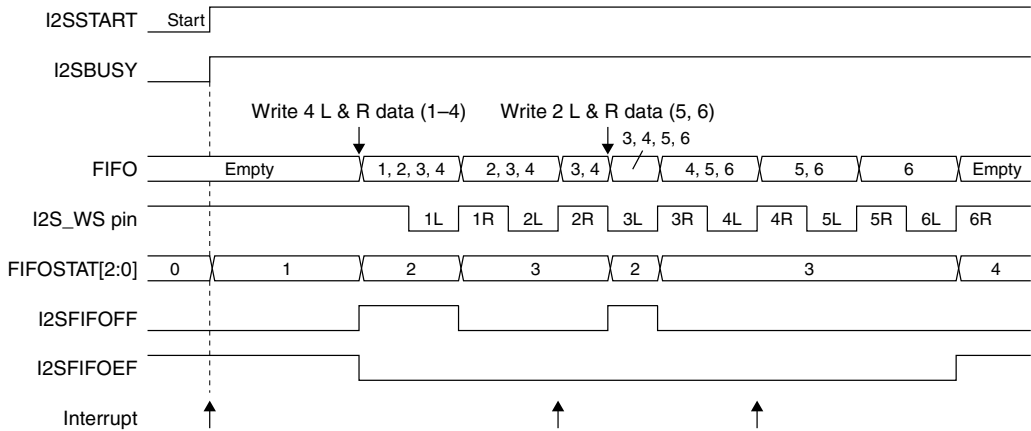
• I²S FIFO one empty interrupt

When the one empty interrupt is enabled, the I²S module generates an interrupt after one stereo data has been read out from the FIFO. However, the I²S module continues shifting out the remaining buffer data until the FIFO becomes absolutely empty.

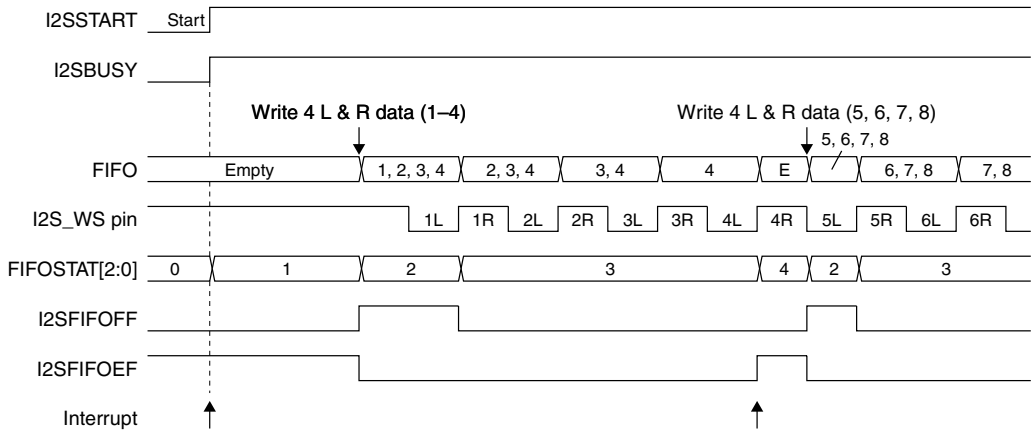
We suggest filling the FIFO with one group of stereo data (16 bits × 2 channels (L & R) × 1) at once in the one empty interrupt handler.

Note: If the handler fills the FIFO with more than one group of data, it may overwrite the remaining data in the FIFO. Therefore, the FIFO should always be filled with one group of data in the one empty interrupt handler.

Half empty interrupt



Whole empty interrupt



One empty interrupt

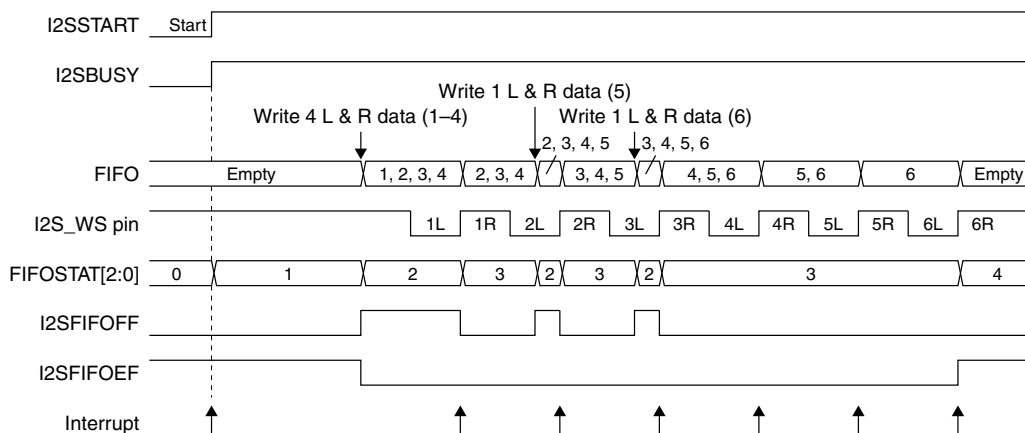


Figure 21.5.1 FIFO Data and Interrupts

The 16-bit register I2S_FIFO is used to write the output data to the FIFO. Up to four stereo data (16 bits \times 2 channels (L & R) \times 4) can be written to the FIFO when it is absolutely empty.

Use a 16-bit or 32-bit memory write instruction for writing data. Note that 8-bit memory write instructions cannot be used.

When a 16-bit memory write instruction is used, first write 16-bit L-channel data to address 0x301410, and then 16-bit R-channel data to address 0x301412. Both channel data must be written as a pair even if “mono” is selected as the output channel mode.

When a 32-bit memory write instruction is used, write both L-channel data (low-order 16-bits) and R-channel data (high-order 16-bits) to address 0x301410 (fixed address).

For more information on the I²S interrupt, see Section 21.6.1.

6.2) Using DMA transfer

The I²S module includes two different types of DMA mode.

DMA transfers are invoked by the cause of one-empty interrupt regardless of the DMA mode set. The interrupt signal is output to both the ITC and DMAC. Therefore, DMA transfer can be performed without generating any I²S interrupt.

• Single channel DMA mode

If L-channel and R-channel audio data are sequentially stored in one memory area, use single channel DMA mode. Use 32-bit data transfer to write both L (low-order 16 bits) and R (high-order 16 bits) data to the FIFO (fixed address 0x301410) for each DMA request.

Note that 16-bit and 8-bit data transfers cannot be specified when single channel DMA mode is used.

• Dual channel DMA mode

If L-channel and R-channel audio data are stored in different locations, use dual channel DMA mode. In this case, perform 16-bit data transfer to write L-channel data to the FIFO (fixed address 0x301410) via DMA Ch.0 and to write R-channel data to the FIFO (fixed address 0x301412) via DMA Ch.1.

Note that 8-bit and 32-bit data transfers cannot be specified when dual channel DMA mode is used. Also be aware that the DMA priority for L-channel data must be higher than that of R-channel data.

For more information on the DMA transfer, see Section 21.6.2.

The I²S module provides two status flags I2SFIFOFF/I2S_FIFO_STAT register and I2SFIFOEF/I2S_FIFO_STAT register to show the FIFO empty or full status.

When four stereo data is written to the FIFO, the FIFO becomes full and I2SFIFOFF is set to 1. Note that the newest data of the FIFO is overwritten if data is written to I2S_FIFO in this status.

When the FIFO becomes empty, I2SFIFOEF is set to 1. When data is written to the FIFO, I2SFIFOEF is reset to 0. Note, however, that the I²S module continues idle status until the FIFO becomes full again.

Furthermore, the I²S provides the status bits FIFOSTAT[2:0]/I2S_FIFO_STAT register that indicate the FIFO state machine.

Table 21.5.2 Monitoring the FIFO State Machine

FIFOSTAT[2:0]	State
0x7–0x6	Reserved
0x5	FLUSH: FIFO is flushing the remained audio data before it stops.
0x4	EMPTY: FIFO is empty.
0x3	LACK: FIFO is not full and not empty.
0x2	FULL: FIFO is full.
0x1	INIT: Initialize all four entries of FIFO.
0x0	STOP: FIFO is idle.

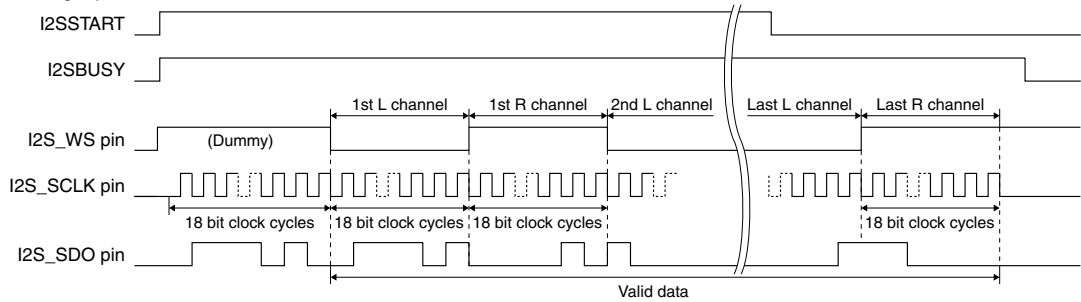
(Default: 0x0)

For more information on FIFOSTAT[2:0], refer to the description of I2S_FIFO_STAT register in Section 21.7.

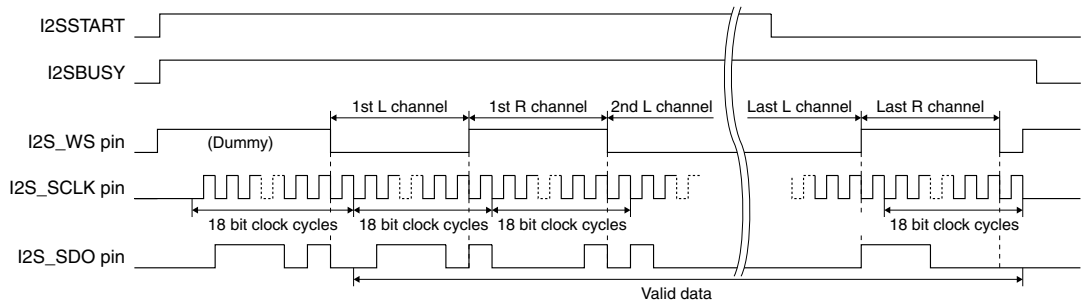
7. To stop output, write 0 to I2SSTART/I2S_START register.

When I2SSTART is set to 0, the I²S module will stop data output after the remaining data stored in the FIFO are all output. The bit clock is stopped with pulled down to low. The word select clock is also stopped with pulled down to low if WCLKMD = 0 or pulled up to high if WCLKMD = 1. When the I²S stops, I2SBUSY is reset to 0.

In left or right justified mode



In I²S mode



Conditions: CHMD[1:0] = 0x0 (stereo), WCLKMD = 0 (L ch = low), BCLKPOL = 0 (falling edge), WCLKCYC[4:0] = 0x2 (18 clocks)

Figure 21.5.2 Data Output Timing Chart

* Output when mute or mono mode is selected

When mute mode is selected using CHMD[1:0]/I2S_CTL register, 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 an interrupt caused by a FIFO empty occurs. If CHMD[1:0] is changed when data is being output, the mode changes after the current L & R data output has finished.

21.6 I²S Interrupt and DMA

This section describes the I²S interrupts and invoking DMA.

For more information on interrupt processing and DMA transfer, see the “Interrupt Controller (ITC)” chapter and the “DMA Controller (DMAC)” chapter, respectively.

21.6.1 Interrupts

The I²S module includes a function for generating the following three different types of interrupts.

- I²S FIFO whole empty interrupt
- I²S FIFO half empty interrupt
- I²S FIFO one empty interrupt

The I²S module outputs one interrupt signal shared by the three above interrupt causes to the interrupt controller (ITC). Select an interrupt mode and set the interrupt enable bit to 1 using the I2S_INT register to generate the corresponding interrupt.

I²S FIFO whole empty interrupt

To use this interrupt, set WEIE/I2S_INT register to 1. If WEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When all data (four stereo data) has been read from the FIFO to output, the I²S module sets WEIF/I2S_INT register to 1, indicating that the FIFO is empty. If whole empty interrupts are enabled (WEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. If WEIF is 1, the application program should fill the FIFO with four stereo data (16 bits × 2 channels (L & R) × 4). When writing each group of audio data using a 16-bit memory access instruction, first fill L-channel data, then fill R-channel data. When a 32-bit memory access instruction is used, fill L-channel data (16-bit low order) and R-channel data (16-bit high order) to the FIFO simultaneously.

I²S FIFO half empty interrupt

To use this interrupt, set HEIE/I2S_INT register to 1. If HEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When a free space for two stereo data becomes available in the FIFO, the I²S module sets HEIF/I2S_INT register to 1. If half empty interrupts are enabled (HEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. If HEIF is 1, the application program should fill the FIFO with two stereo data (16 bits × 2 channels (L & R) × 2). When writing each group of audio data using a 16-bit memory access instruction, first fill L-channel data, then fill R-channel data. When a 32-bit memory access instruction is used, fill L-channel data (16-bit low order) and R-channel data (16-bit high order) to the FIFO simultaneously.

I²S FIFO one empty interrupt

To use this interrupt, set OEIE/I2S_INT register to 1. If OEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When a free space for one stereo data becomes available in the FIFO, the I²S module sets OEIF/I2S_INT register to 1. If one empty interrupts are enabled (OEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. If OEIF is 1, the application program should fill the FIFO with one stereo data (16 bits × 2 channels (L & R) × 1). When writing audio data using a 16-bit memory access instruction, first fill L-channel data, then fill R-channel data. When a 32-bit memory access instruction is used, fill L-channel data (16-bit low order) and R-channel data (16-bit high order) to the FIFO simultaneously.

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

Note: At the beginning of I²S transfer (by setting I2SSTART to 1), the FIFO is absolutely empty. So the I²S module continues the initial status until the FIFO becomes full with four stereo data (16 bits × 2 channels (L & R) × 4).

The interrupt handler should write one, two or four stereo data to the FIFO according to the interrupt mode selected.

- When filling the FIFO before enabling the interrupt
The application program should write four stereo data to the FIFO first, and then enable the interrupt.
- When filling the FIFO after enabling the interrupt
The application program can enable the interrupt directly. The FIFO becomes full when the first whole-empty interrupt, second half-empty interrupt, or fourth one-empty interrupt occurs after the interrupt is enabled.

21.6.2 DMA Transfer

The cause of one empty interrupt can invoke a DMA. This allows continuous data output through DMA transfer between memory and the FIFO. The interrupt signal is output to both the ITC and DMAC. Therefore, DMA transfer can be performed without generating any I²S interrupt.

The following lists the DMAC channels that allow selection of the I²S one-empty interrupt cause as the trigger.

DMAC Ch.0: Used for L and R data transfer with single DMA, or L data transfer with dual DMA.

DMAC Ch.1: Used for L and R data transfer with single DMA, or R data transfer with dual DMA.

Use single or dual DMAC channels according to the audio data storing method.

Single channel DMA mode

When L-channel and R-channel audio data are sequentially stored in a memory area, use a DMAC channel and perform 32-bit data transfer to write both L (low-order 16 bits) and R (high-order 16 bits) data to the FIFO (fixed address 0x301410) for each DMA request. Note that 16-bit and 8-bit data transfer cannot be specified.

Dual channel DMA mode

When L-channel and R-channel audio data are stored in different locations, use DMAC Ch.0 and Ch.1. In this case, perform 16-bit data transfer to write L-channel data to the FIFO (fixed address 0x301410) using DMAC Ch.0 and to write R-channel data to the FIFO (fixed address 0x301412) using DMAC Ch.1. The I²S one-empty DMA request is sent to DMAC Ch.0 and Ch.1 simultaneously. However, DMAC Ch.0 starts a DMA transfer first as it priority over Ch.1. Therefore, DMAC Ch.0 must be used for L-channel data transfer. Since DMAC Ch.0 transfers L-channel data first, then DMAC Ch.1 transfers R-channel data, enable DMAC Ch.1 end-of-transfer interrupt only. The DMAC Ch.0 end-of-transfer interrupt should be disabled. In the DMAC Ch.1 interrupt handler, configure the DMAC transfer counter, memory access address and enable DMAC for both Ch.0 and Ch.1. Note that 8-bit and 32-bit data transfer cannot be specified when dual DMA channels are used.

For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

21.7 Control Register Details

Table 21.7.1 List of I²S Registers

Address	Register name		Function
0x301400	I2S_CTL	I ² S Control Register	Sets the I ² S output conditions.
0x301404	I2S_DV_MCLK	I ² S Master Clock Division ratio Register	Configures the master clock.
0x301406	I2S_DV_AUDIO_CLK	I ² S Audio Clock Division ratio Register	Configures the audio clock.
0x301408	I2S_START	I ² S Start/Stop Register	Controls/indicates I ² S start/stop status.
0x30140a	I2S_FIFO_STAT	I ² S FIFO Status Register	Indicates the FIFO status.
0x30140c	I2S_INT	I ² S Interrupt Control Register	Controls I ² S interrupts.
0x301410	I2S_FIFO	I ² S FIFO Register	L-channel output data
0x301412			R-channel output data

The following describes each I²S register. These are all 16-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

I²S Control Register (I2S_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S Control Register (I2S_CTL)	0x301400 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.	
		D8	DTSIGN	I ² S signed/unsigned data format select	1 Signed 0 Unsigned	0	R/W		
		D7	WCLKMD	I ² S output word clock mode select	1 L: High R: Low 0 L: Low R: High	0	R/W		
		D6	BCLKPOL	I ² S output bit clock polarity select	1 Negative 0 Positive	0	R/W		
		D5	DTFORM	I ² S output data format select	1 LSB first 0 MSB first	0	R/W		
		D4	I2SOUTEN	I ² S output enable	1 Enable 0 Disable	0	R/W		
		D3–2	DTTMG[1:0]	I ² S output data timing select	DTTMG[1:0] 0x3 reserved 0x2 Right justified 0x1 Left justified 0x0 I ² S	Timing mode	0x0	R/W	
		D1–0	CHMD[1:0]	I ² S output channel mode select	CHMD[1:0] 0x3 Mute 0x2 Mono left 0x1 Mono right 0x0 Stereo	Channel mode	0x0	R/W	

Note: All the data transfer conditions must be set using this register before setting I2SSTART/I2S_START register to start data output from the I²S module.

D[15:9] Reserved

D8 **DTSIGN: I²S Signed/Unsigned Data Format Select Bit**

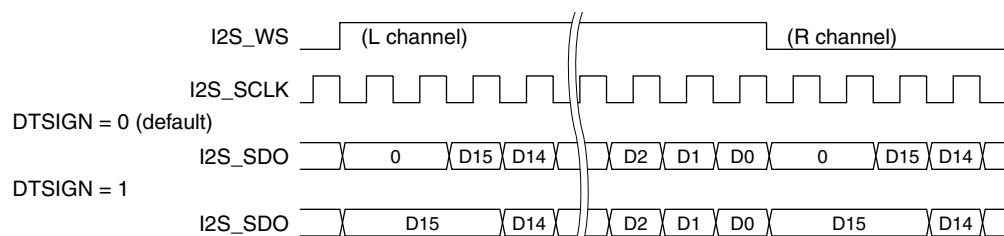
Selects the data format in right justified mode.

1 (R/W): Signed

0 (R/W): Unsigned (default)

Setting DTSIGN to 0 (default) selects the unsigned format. The high-order bits that exceed the valid data size are set to 0. Setting 1 selects the signed format. The high-order bits that exceed the valid data size are set to the sign bit value (D15) of the valid data.

This setting is effective only in right justified mode. Set DTSIGN to 0 when another data output timing mode is selected.



(MSB first, right justified mode, number of bit clock cycles = 18)

Figure 21.7.1 Unsigned and Signed Format

D7 **WCLKMD: I²S Output Word Clock Mode Select Bit**

Selects the I2S_WS output 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)

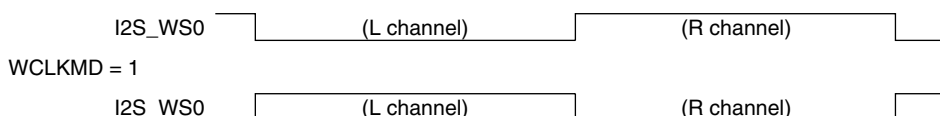


Figure 21.7.2 Word Clock Mode

D6 **BCLKPOL: I²S Output Bit Clock Polarity Select Bit**

Selects the bit clock polarity.

1 (R/W): Negative

0 (R/W): Positive (default)

When BCLKPOL is 0, the I2S_SDO output changes at the falling edge of the I2S_SCLK clock (bit clock) and the external DAC samples the data bit at the rising edge of I2S_SCLK.

When BCLKPOL is set to 1, the I2S_SDO output changes at the rising edge of I2S_SCLK and the external DAC samples the data bit at the falling edge of I2S_SCLK.

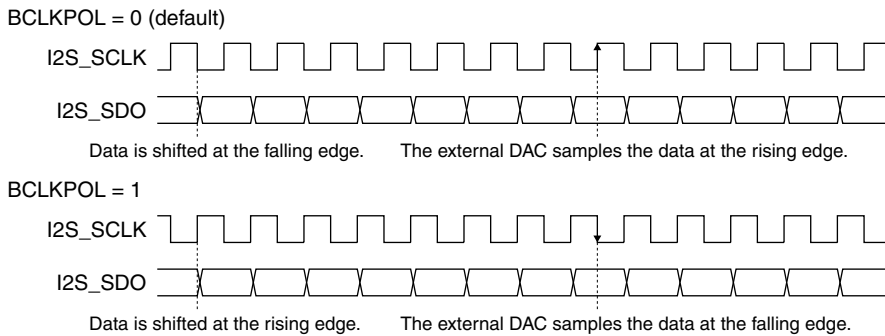


Figure 21.7.3 Bit Clock Polarity

D5 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)

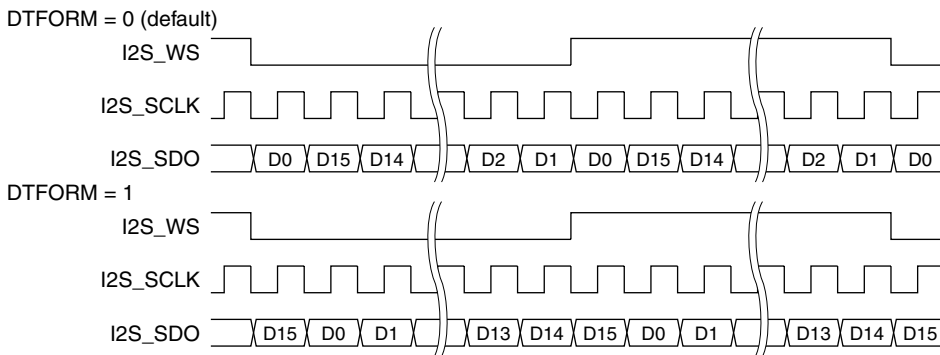


Figure 21.7.4 Output Data Format (Example in I²S Mode)

D4 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 pin is maintained at standby status and the I2S_WS pin is fixed at 0. The I2S_SDO pin is left unchanged. The I2S_SCLK pin is fixed at 0 (when BCLKPOL = 0) or 1 (when BCLKPOL = 1).

When I2SOUTEN is set to 1, all output pins enter standby status.

I2SOUTEN can be altered even if data is being output. This enables the I²S module to pause or resume shifting data (data output).

D[3:2] DTTMG[1:0]: I²S Output Data Timing Select Bits

Selects the data bit output timing.

Table 21.7.2 Data Output Timing

DTTMG[1:0]	Data output timing mode
0x3	Reserved
0x2	Right justified mode
0x1	Left justified mode
0x0	I ² S mode

(Default: 0x0)

When DTTMG[1:0] is set to 0x0 (default), I²S mode is selected. In this mode, the first bit of each data is output after one I2S_SCLK clock delay from the I2S_WS signal edge.

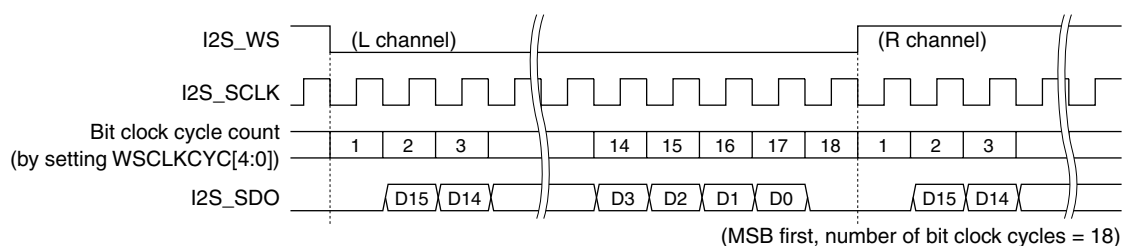


Figure 21.7.5 Data Output Timing 1 (I²S Mode)

When DTTMG[1:0] is set to 0x1, left justified mode is selected. In this mode, each data output will start at the I2S_WS signal edge.

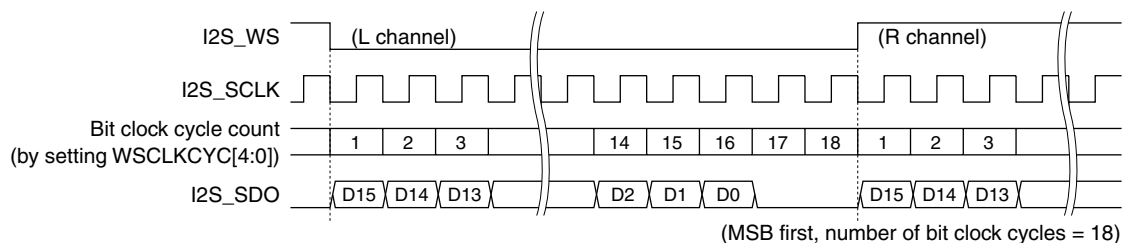


Figure 21.7.6 Data Output Timing 2 (Left Justified Mode)

When DTTMG[1:0] is set to 0x2, right justified mode is selected. In this mode, output data will be right justified to the I2S_WS signal edge.

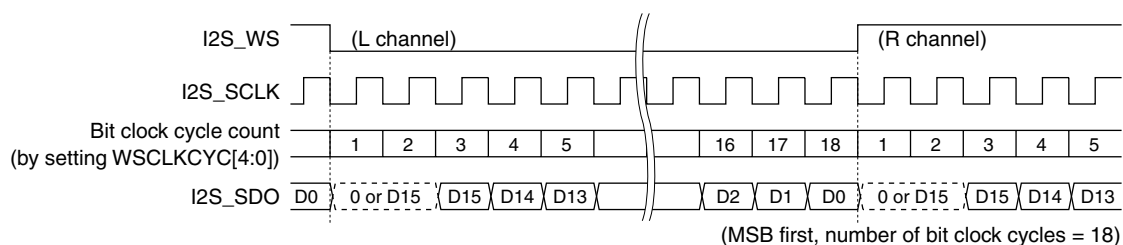


Figure 21.7.7 Data Output Timing 3 (Right Justified Mode)

Note: When using right justified mode, the number of bit clock cycles (sample clock period) must be equal to or greater than [Data bit size + 2].

D[1:0] CHMD[1:0]: I²S Output Channel Mode Select Bits

Selects the I²S output channel mode.

Table 21.7.3 Output Channel Mode Selection

CHMD[1:0]	Output channel mode	L channel	R channel
0x3	Mute	0	0
0x2	Mono (L)	Data output	0
0x1	Mono (R)	0	Data output
0x0	Stereo	Data output	Data output

(Default: 0x0)

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 an interrupt occurs.

I²S Master Clock Division Ratio Register (I2S_DV_MCLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S Master Clock Division Ratio Register (I2S_DV_MCLK)	0x301404 (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.	
		D5–0	MCLKDIV[5:0]	I2S_MCLK division ratio select	MCLKDIV[5:0] Division ratio	0x0	R/W	Source clock = PCLK1	
					0x3f	1/64			
					0x3e	1/63			
					0x3d	1/62			
					:	:			
					0x2	1/3			
					0x1	1/2			
					0x0	1/1			

D[15:6] Reserved

D[5:0] MCLKDIV[5:0]: I2S_MCLK Division Ratio Select Bits

Configures the I²S master clock (I2S_MCLK) to be output from the I2S_MCLK pin.

The I²S module generates the I2S_MCLK by dividing the operating clock (PCLK1 generated by the CMU). Specify the division ratio using MCLKDIV[5:0].

Table 21.7.4 I2S_MCLK (Master Clock) Settings

MCLKDIV[5:0]	PCLK1 division ratio
0x3f	1/64
0x3e	1/63
0x3d	1/62
:	:
0x2	1/3
0x1	1/2
0x0	1/1

(Default: 0x0)

I²S Audio Clock Division Ratio Register (I2S_DV_AUDIO_CLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S Audio Clock Division Ratio Register (I2S_DV_AUDIO_CLK)	0x301406 (16 bits)	D15–13	–	reserved	–	–	–	0 when being read.	
		D12–8	WSCLKCYC[4:0]	I ² S WS clock cycle setup	WSCLKCYC[4:0] Clock period	0x0	R/W		
					Other	reserved			
					0x10	32 clocks			
					0xf	31 clocks			
					0xe	30 clocks			
					0xd	29 clocks			
					0xc	28 clocks			
					0xb	27 clocks			
					0xa	26 clocks			
					0x9	25 clocks			
					0x8	24 clocks			
					0x7	23 clocks			
					0x6	22 clocks			
					0x5	21 clocks			
					0x4	20 clocks			
					0x3	19 clocks			
					0x2	18 clocks			
					0x1	17 clocks			
					0x0	16 clocks			
		D7–0	BCLKDIV[7:0]	I ² S bit clock division ratio select	BCLKDIV[7:0] Division ratio	0x0	R/W	Source clock = PCLK1	
					0xff	1/512			
					0xfe	1/510			
					0xfd	1/508			
					:	:			
					0x2	1/6			
					0x1	1/4			
					0x0	1/2			

D[15:13] Reserved

D[12:8] WSCLKCYC[4:0]: I²S WS Clock Cycle Setup Bits

Specifies the sample clock (I2S_WS signal) period.

The I²S generates the sample clock to be output from the I2S_WS pin by counting the bit clock configured with BCLKDIV[7:0]. Specify the half cycle (a high or low level period) of the I2S_WS clock with the number of bit clock cycles using WSCLKCYC[4:0].

Table 21.7.5 Sample Clock Settings

WSCLKCYC[4:0]	Sample clock period (number of bit clock cycles)
0x1f–0x11	Reserved
0x10	32 clocks
0xf	31 clocks
0xe	30 clocks
0xd	29 clocks
0xc	28 clocks
0xb	27 clocks
0xa	26 clocks
0x9	25 clocks
0x8	24 clocks
0x7	23 clocks
0x6	22 clocks
0x5	21 clocks
0x4	20 clocks
0x3	19 clocks
0x2	18 clocks
0x1	17 clocks
0x0	16 clocks

(Default: 0x0)

The sampling clock frequency is calculated as below.

$$f_s = \frac{f_{I2S_SCLK}}{n \times 2} \text{ [Hz]}$$

f_s : Sampling clock frequency [Hz]

f_{I2S_SCLK} : Bit clock frequency [Hz] (See Table 21.7.6.)

n : Number of bit clocks selected by WSCLKCYC[4:0] (See Table 21.7.5.)

Note: The value to be set to the WSCLKCYC[4:0] is not the number of audio data bits, but the number of bit clock cycles that is used to adjust the sample clock period. It must be equal to or greater than the number of audio data bits (16 bits).

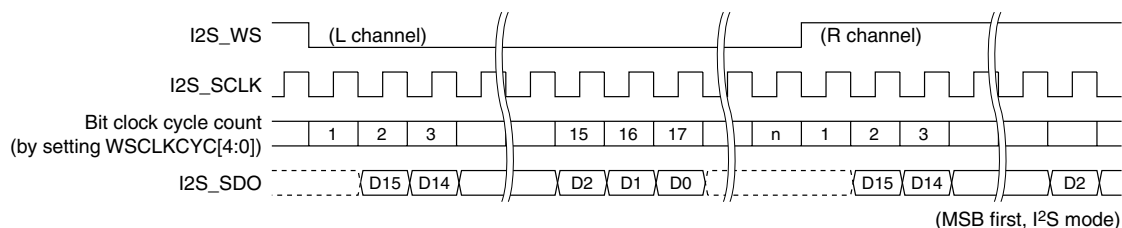


Figure 21.7.8 Sample Clock Period

D[7:0] BCLKDIV[7:0]: I²S Bit Clock Division ratio Select Bits

Configures the bit clock to be output.

The I²S module generates the bit clock to be output from the I2S_SCLK pin of the I²S by dividing PCLK1. Specify the division ratio using BCLKDIV[7:0].

Table 21.7.6 I2S_SCLK (Bit Clock) Settings

BCLKDIV[7:0]	PCLK1 division ratio
0xff	1/512
0xfe	1/510
0xfd	1/508
:	:
0x2	1/6
0x1	1/4
0x0	1/2

(Default: 0x0)

The I²S bit clock frequency is calculated as below.

$$f_{i2s_SCLK} = \frac{f_{PCLK1}}{(BCLKDIV + 1) \times 2} \text{ [Hz]}$$

f_{i2s_SCLK} : I²S bit clock frequency [Hz]

f_{PCLK1} : PCLK1 clock frequency [Hz]

BCLKDIV: BCLKDIV[7:0] set value (0x0–0xff)

I²S Start/Stop Register (I2S_START)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S Start/Stop Register (I2S_START)	0x301408 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7	I2SBUSY	I ² S busy flag	1 Busy 0 Idle	0	R	0 when being read.
		D6–1	–	reserved	–	–	–	0 when being read.
		D0	I2SSTART	I ² S start/stop control	1 Start (run) 0 Stop	0	R/W	

D[15:8] Reserved

D7 I2SBUSY: I²S Busy Flag Bit

Indicates the data output status of the I²S module.

1 (R): Busy

0 (R): Idle (default)

I2SBUSY is set to 1 when 1 is written to I2SSTART 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 data output of the I²S.

1 (R/W): Start

0 (R/W): Stop (default)

Writing 1 to I2SSTART starts serial data transmission through the I2S_SDO pin.

Writing 0 to I2SSTART stops data transmission. Note, however, that the data currently stored in the FIFO will be continuously transmitted through the I2S_SDO pin until the FIFO becomes empty. After I2SSTART is set to 0, new transmit data cannot be written to the FIFO.

Note: Be sure to avoid altering the I2S_DV_MCLK and I2S_DV_AUDIO_CLK registers when I2SSTART is 1.

I²S FIFO Status Register (I2S_FIFO_STAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S FIFO Status Register (I2S_FIFO_STAT)	0x30140a (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.	
		D4–2	FIFOSTAT [2:0]	I ² S FIFO state machine	FIFOSTAT[2:0]	State	0x0	R	
					0x7–0x6	reserved			
					0x5	FLUSH			
	0x4	EMPTY							
	0x3	LACK							
	0x2	FULL							
	0x1	INIT							
	0x0	STOP							
	D1	I2SFIFOFF	I ² S FIFO full flag	1 Full 0 Not full	0	R			
	D0	I2SFIFOE	I ² S FIFO empty flag	1 Empty 0 Not empty	1	R			

D[15:5] Reserved

D[4:2] FIFOSTAT[2:0]: I²S FIFO State Machine Bits

Indicates the transmit FIFO status.

Table 21.7.7 Monitoring the FIFO State Machine

FIFOSTAT[2:0]	State
0x7–0x6	Reserved
0x5	<p>FLUSH: FIFO is flushing the remained audio data before it stops.</p> <p>This means that the I²S module is stopped by setting I2SSTART to 0, but there are some audio data remained in the FIFO. The I²S module enters FLUSH state until all the remained audio data has been shifting out. After that, the FIFOSTAT[2:0] changes from FLUSH to STOP.</p> <p>If the application program restarts the I²S module by setting I2SSTART to 1 again in FLUSH state, FIFOSTAT[2:0] changes back to FULL or LACK according to the current FIFO status.</p>
0x4	<p>EMPTY: FIFO is empty.</p> <p>This means that the I²S FIFO became absolutely empty, but it has not been filled to full yet.</p> <p>Once the FIFO becomes absolutely empty, FIFOSTAT[2:0] is set to EMPTY. When the application program fills the FIFO with less than four stereo data (16 bits × 2 channels (L & R) × 4), FIFOSTAT[2:0] retains EMPTY and the I²S module stops shifting out data until the FIFO becomes full again. After the FIFO becomes full, FIFOSTAT[2:0] is set to FULL and the I²S module starts shifting out data again.</p>
0x3	<p>LACK: FIFO is not full and not empty.</p> <p>This means that the I²S FIFO has data, but is not full and also not empty.</p>
0x2	<p>FULL: FIFO is full.</p> <p>This means that the I²S FIFO becomes full with four stereo data. The I²S module will start shifting out the buffered data.</p>
0x1	<p>INIT: Initialize all four entries of FIFO.</p> <p>This means that the I²S module is started and waits for filling the FIFO with the first four stereo data.</p>
0x0	<p>STOP: FIFO is idle.</p> <p>This means that the I²S module is stopped.</p>

(Default: 0x0)

D1 I2SFIFOFF: I²S FIFO Full Flag Bit

Indicates whether the transmit FIFO is full or not.

1 (R): Full

0 (R): Not full (default)

I2SFIFOFF is set to 1 when the FIFO becomes full of the written data (16 bits × 2 channels (L & R) × 4) to indicate that no more data can be written.

I2SFIFOFF is reset to 0 when the stored data is read out to transmit.

D0 I2SFIFOEF: I²S FIFO Empty Flag Bit

Indicates whether the transmit FIFO is empty or not.

1 (R): Empty (default)

0 (R): Not empty

I2SFIFOEF is reset to 0 when a transmit data is written to the FIFO and is set to 1 when all the stored data have been transmitted.

When the FIFO becomes absolutely empty (I2SFIFOEF = 1), the I²S module stops shifting data and enters EMPTY state. In this case, the I²S module resumes output when the FIFO becomes full again.

I²S Interrupt Control Register (I2S_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
I ² S Interrupt Control Register (I2S_INT)	0x30140c (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.	
		D10	WEIF	I ² S FIFO whole empty int. flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	HEIF	I ² S FIFO half empty interrupt flag	1	0	0	R/W	
		D8	OEIF	I ² S FIFO one empty interrupt flag	1	0	0	R/W	
		D7–3	–	reserved	–	–	–	–	0 when being read.
		D2	WEIE	I ² S FIFO whole empty int. enable	1 Enable	0 Disable	0	R/W	
		D1	HEIE	I ² S FIFO half empty int. enable	1 Enable	0 Disable	0	R/W	
		D0	OEIE	I ² S FIFO one empty int. enable	1 Enable	0 Disable	0	R/W	

D[15:11] Reserved**D10 WEIF: I²S FIFO Whole Empty Interrupt Flag Bit**

Indicates whether the cause of I²S FIFO whole empty interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

When all data (four stereo data) has been read from the FIFO to transmit, the I²S module sets WEIF to 1, indicating that the FIFO is empty. If I²S FIFO whole empty interrupts are enabled (WEIE = 1), an interrupt request is sent simultaneously to the ITC. The interrupt handler needs to fill the FIFO with four stereo data (16 bits × 2 channels (L & R) × 4). Then reset WEIF by writing 1 at the end of the interrupt handler.

D9 HEIF: I²S FIFO Half Empty Interrupt Flag Bit

Indicates whether the cause of I²S FIFO half empty interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

When a free space for two stereo data becomes available in the FIFO, the I²S module sets HEIF to 1. If I²S FIFO half empty interrupts are enabled (HEIE = 1), an interrupt request is sent simultaneously to the ITC. The interrupt handler needs to fill the FIFO with two stereo data (16 bits × 2 channels (L & R) × 2). Then reset HEIF by writing 1 at the end of the interrupt handler.

D8 OEIF: I²S FIFO One Empty Interrupt Flag Bit

Indicates whether the cause of I²S FIFO one empty interrupt has occurred or not.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

When a free space for one stereo data becomes available in the FIFO, the I²S module sets OEIF to 1. If I²S FIFO one empty interrupts are enabled (OEIE = 1), an interrupt request is sent simultaneously to the ITC. The interrupt handler needs to fill the FIFO with one stereo data (16 bits × 2 channels (L & R) × 1). Then reset OEIF by writing 1 at the end of the interrupt handler.

D[7:3] Reserved**D2 WEIE: I²S FIFO Whole Empty Interrupt Enable Bit**

Enables or disables I²S FIFO whole empty interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

Setting WEIE to 1 enables I²S FIFO whole empty interrupt requests to the ITC. Setting it to 0 disables interrupts.

D1 HEIE: I²S FIFO Half Empty Interrupt Enable Bit

Enables or disables I²S FIFO half empty interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

Setting HEIE to 1 enables I²S FIFO half empty interrupt requests to the ITC. Setting it to 0 disables interrupts.

D0 OEIE: I²S FIFO One Empty Interrupt Enable Bit

Enables or disables I²S FIFO one empty interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

Setting OEIE to 1 enables I²S FIFO one empty interrupt requests to the ITC. Setting it to 0 disables interrupts.

Note: Always be sure to select only one interrupt mode by setting one of the interrupt enable bit (WEIE, HEIE, or OEIE).

I²S FIFO Register (I2S_FIFO)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S FIFO Register (I2S_FIFO)	0x301410 (16 bits)	D15–0	I2SFIFO [31:0]	I ² S FIFO (L-channel output data)	0 to 0xffffffff	0x0	W	0 when being read.
	0x301412 (16 bits)	D15–0		I ² S FIFO (R-channel output data)				

D[15:0] I2SFIFO[15:0]: I²S FIFO (Output Data) Bits

Write audio data to the FIFO through this address.

Up to four stereo data (16 bits × 2 channels (L & R) × 4) can be written to the FIFO. Once the FIFO becomes absolutely empty, the I²S module stops shifting audio data and waits until the FIFO becomes full with four stereo data.

Example:

After the I²S module starts by setting I2SSTART to 1, the FIFO becomes absolutely empty.

During audio playing, the interrupt handler may not be invoked in time and the FIFO may become absolutely empty.

In these cases, the I²S module enters INIT/EMPTY state and does not shift out data until the FIFO becomes full by the interrupt handler or DMA transfer.

When writing data in the interrupt handler routine

When writing data in the interrupt handler routine, use a 16-bit memory write (`ld.h [%rb], %rs`) instruction or a 32-bit memory write (`ld.w [%rb], %rs`) instruction. Note that 8-bit memory write instructions cannot be used.

When using a 16-bit memory write (`ld.h [%rb], %rs`) instruction, first write L-channel data to address 0x301410, then R-channel data to address 0x301412. Both channel data must be written as a pair even if “mono” is selected as the output channel mode.

Write the first to fourth data to the same addresses (0x301410, 0x301412) without changing.

When using a 32-bit memory write (`ld.w [%rb], %rs`) instruction, write both L-channel data (low-order 16 bits) and R-channel data (high-order 16 bits) to address 0x301410 with a single 32-bit memory write instruction.

Write the first to fourth data to the same address (0x301410) without changing.

When writing data via DMAC

When L-channel and R-channel audio data are sequentially stored in a memory area, use a DMAC channel (Ch.0 or Ch.1) and perform 32-bit data transfer to write both L (low-order 16 bits) and R (high-order 16 bits) data to address 0x301410 (fixed) for each DMA request. Note that 16-bit and 8-bit data transfer cannot be specified.

When L-channel and R-channel audio data are stored in different locations, use DMAC Ch.0 and Ch.1. In this case, perform 16-bit data transfer to write L-channel data to address 0x301410 (fixed) using DMAC Ch.0 and to write R-channel data to address 0x301412 (fixed) using DMAC Ch.1. The I²S one-empty DMA request is sent to DMAC Ch.0 and Ch.1 simultaneously. However, DMAC Ch.0 starts a DMA transfer first as it priority over Ch.1. Therefore, DMAC Ch.0 must be used for L-channel data transfer. Note that 8-bit and 32-bit data transfer cannot be specified when dual DMA channels are used.

21.8 Setting the I²S Clocks

This section explains how to configure the I2S_MCLK, I2S_WS, and I2S_SCLK clocks.

The following shows how to determine the clock setting values from the sampling rate. The example below assumes that the system clock frequency is 33 MHz and the sampling rate of audio data is 44.1 kHz.

The sample clock (I2S_WS) is in sync with the master clock (I2S_MCLK), so the following equation is formulated:

$$\frac{f_{i2s_mclk}}{f_{i2s_ws}} = \text{Integer}$$

where f_{i2s_mclk} is the output master clock (I2S_MCLK) frequency and f_{i2s_ws} is the sample clock (I2S_WS) frequency.

$$f_{i2s_mclk} = \frac{33 \text{ MHz}}{\text{MCLKDIV}[5:0] + 1} \quad (\text{eq1})$$

$$f_{i2s_ws} = \frac{33 \text{ MHz}}{(\text{BCLKDIV}[7:0] + 1) \times 2 \times (\text{WSCLKCYC}[4:0] + 16) \times 2} \quad (\text{eq2})$$

$$\frac{(\text{BCLKDIV}[7:0] + 1) \times 2 \times (\text{WSCLKCYC}[4:0] + 16) \times 2}{\text{MCLKDIV}[5:0] + 1} = \text{Integer} \quad (\text{eq3})$$

Table 21.8.1 I2S_MCLK (Master Clock) Settings

MCLKDIV[5:0]	PCLK1 division ratio
0x3f	1/64
0x3e	1/63
0x3d	1/62
:	:
0x2	1/3
0x1	1/2
0x0	1/1

Table 21.8.2 I2S_SCLK (Bit Clock) Settings

BCLKDIV[7:0]	PCLK1 division ratio
0xff	1/512
0xfe	1/510
0xfd	1/508
:	:
0x2	1/6
0x1	1/4
0x0	1/2

Table 21.8.3 Sample Clock Period Settings

WSCLKCYC[4:0]	Sample clock period (number of bit clock cycles)
0x1f–0x11	Reserved
0x10	32 clocks
0xf	31 clocks
0xe	30 clocks
0xd	29 clocks
0xc	28 clocks
0xb	27 clocks
0xa	26 clocks
0x9	25 clocks
0x8	24 clocks
0x7	23 clocks
0x6	22 clocks
0x5	21 clocks
0x4	20 clocks
0x3	19 clocks
0x2	18 clocks
0x1	17 clocks
0x0	16 clocks

The table below is made from Equation 2 (eq2) using Excel.

Table 21.8.4 List of Sample Clock Frequencies

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	SysClk	BCLKDIV[7:0]	WSCLKCYC[4:0]																
2	[kHz]		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	33000	0	515.63	485.29	458.33	434.21	412.50	392.86	375.00	358.70	343.75	330.00	317.31	305.56	294.64	284.48	275.00	266.13	257.81
4		1	257.81	242.65	229.17	217.11	206.25	196.43	187.50	179.35	171.88	165.00	158.65	152.78	147.32	142.24	137.50	133.06	128.91
5		2	171.88	161.76	152.78	144.74	137.50	130.95	125.00	119.57	114.58	110.00	105.77	101.85	98.21	94.83	91.67	88.71	85.94
6		3	128.91	121.32	114.58	108.55	103.13	98.21	93.75	89.67	85.94	82.50	79.33	76.39	73.66	71.12	68.75	66.53	64.45
7		4	103.13	97.06	91.67	86.84	82.50	78.57	75.00	71.74	68.75	66.00	63.46	61.11	58.93	56.90	55.00	53.23	51.56
8		5	85.94	80.88	76.39	72.37	68.75	65.48	62.50	59.78	57.29	55.00	52.88	50.93	49.11	47.41	45.83	44.35	42.97
9		6	73.66	69.33	65.48	62.03	58.93	56.12	53.57	51.24	49.11	47.14	45.33	43.65	42.09	40.64	39.29	38.02	36.83
10		7	64.45	60.66	57.29	54.28	51.56	49.11	46.88	44.84	42.97	41.25	39.66	38.19	36.83	35.56	34.38	33.27	32.23
11		8	57.29	53.92	50.93	48.25	45.83	43.65	41.67	39.86	38.19	36.67	35.26	33.95	32.74	31.61	30.56	29.57	28.65
12		9	51.56	48.53	45.83	43.42	41.25	39.29	37.50	35.87	34.38	33.00	31.73	30.56	29.46	28.45	27.50	26.61	25.78
13		10	46.88	44.12	41.67	39.47	37.50	35.71	34.09	32.61	31.25	30.00	28.85	27.78	26.79	25.86	25.00	24.19	23.44
14		11	42.97	40.44	38.19	36.18	34.38	32.74	31.25	29.89	28.65	27.50	26.44	25.46	24.55	23.71	22.92	22.18	21.48
15		12	39.66	37.33	35.26	33.40	31.73	30.22	28.85	27.59	26.44	25.38	24.41	23.50	22.66	21.88	21.15	20.47	19.83
16		13	36.83	34.66	32.74	31.02	29.46	28.06	26.79	25.62	24.55	23.57	22.66	21.83	21.05	20.32	19.64	19.01	18.42
17		14	34.38	32.35	30.56	28.95	27.50	26.19	25.00	23.91	22.92	22.00	21.15	20.37	19.64	18.97	18.33	17.74	17.19
18		15	32.23	30.33	28.65	27.14	25.78	24.55	23.44	22.42	21.48	20.63	19.83	19.10	18.42	17.78	17.19	16.63	16.11
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
257		254	2.02	1.90	1.80	1.70	1.62	1.54	1.47	1.41	1.35	1.29	1.24	1.20	1.16	1.12	1.08	1.04	1.01
258		255	2.01	1.90	1.79	1.70	1.61	1.53	1.46	1.40	1.34	1.29	1.24	1.19	1.15	1.11	1.07	1.04	1.01

Cell [A3]: System clock frequency (33000 kHz)

Cells [B3:B258]: BCLKDIV[7:0] settings (0 to 255)

Cells [C2:S2]: WSCLKCYC[4:0] settings (0 to 16)

Cells [C3:S258]: f_{12S_ws} calculated by Equation 2 (eq2) according to the BCLKDIV[7:0] and WSCLKCYC[4:0] settings

Cell [C3] = $\$A\$3/((\$B\$3+1)*2*(\$C\$2+16)*2)$ Cell [S3] = $\$A\$3/((\$B\$3+1)*2*(\$S\$2+16)*2)$

:

Cell [C258] = $\$A\$3/((\$B\$258+1)*2*(\$C\$2+16)*2)$ Cell [S258] = $\$A\$3/((\$B\$258+1)*2*(\$S\$2+16)*2)$

If you use another system clock frequency, enter the frequency in kHz to Cell [A3]. Cells [C3:S258] will be corrected according to the entered value.

Find “44.1 (kHz)” or an approximate value from the table. You may choose “44.12” in Cell [D13].

You may get the BCLKDIV[7:0] and WSCLKCYC[4:0] values as 10 and 1, respectively.

Substituting these values in Equation 3 (eq3) yields the MCLKDIV[5:0] values.

$$\frac{(10 + 1) \times 2 \times (1 + 16) \times 2}{\text{MCLKDIV}[5:0] + 1} = \text{Integer}$$

The table below is made from Equation 3 (eq3) using Excel.

Table 21.8.5 MCLKDIV[5:0] Valid Values

	U	V
1	BCLKDIV[7:0]	WSCLKCYC[4:0]
2	10	1
3		
4	MCLKDIV[5:0]	Results
5	0	Integer
6	1	Integer
7	2	–
8	3	Integer
9	4	–
:	:	–
15	10	Integer
:	:	–
21	16	Integer
:	:	–
26	21	Integer
:	:	–
38	33	Integer
:	:	–
48	43	Integer
:	:	–
68	63	–

Cell [U2]: BCLKDIV[7:0] setting (10)

Cell [V2]: WSCLKCYC[4:0] setting (1)

Cells [V5:V68]: Results of Equation 3 (eq3)

Cell [V5] = IF(MOD((U\$2+1)*2*(V\$2+16)*2, (U5+1))=0, "Integer", "-")

⋮

Cell [V68] = IF(MOD((U\$2+1)*2*(V\$2+16)*2, (U68+1))=0, "Integer", "-")

Enter the selected BCLKDIV[7:0] and WSCLKCYC[4:0] values to Cells [U2] and [V2], respectively.

"Integer" appears in the cell corresponding to the MCLKDIV[5:0] value that can be set.

MCLKDIV[5:0] = 0, 1, 3, 10, 16, 21, 33, 43

Table 21.8.6 Master Clock Frequency

MCLKDIV[5:0]	f _{12S_MCLK}
0	33 MHz (748 fs)
1	16.5 MHz (374 fs)
3	8.25 MHz (187 fs)
10	3 MHz (68 fs)
16	1.941 MHz (44 fs)
21	1.5 MHz (34 fs)
33	970.588 kHz (22 fs)
43	750 kHz (17 fs)

Select an appropriate MCLKDIV[5:0] value according to the value listed in the above table.

22 Remote Controller (REMC)

22.1 REMC Module Overview

The S1C33L26 incorporates a remote controller (REMC) module for generating infrared remote control communication signals.

The following shows the features of the REMC module:

- Supports input and output infrared remote control communication signals.
- Incorporates a carrier generator for generating a carrier signal using the prescaler output clock.
- Incorporates an 8-bit down-counter for counting the transfer data length.
- Incorporates a modulator for generating transmission data of the specified carrier length.
- Incorporates an edge detector for detecting input signal rising and falling edges.
- Can generate counter underflow interrupts indicating that the specified data length has been transmitted and input rising/falling edge detection interrupts for data receive processing.

Figure 22.1.1 shows the configuration of the REMC module.

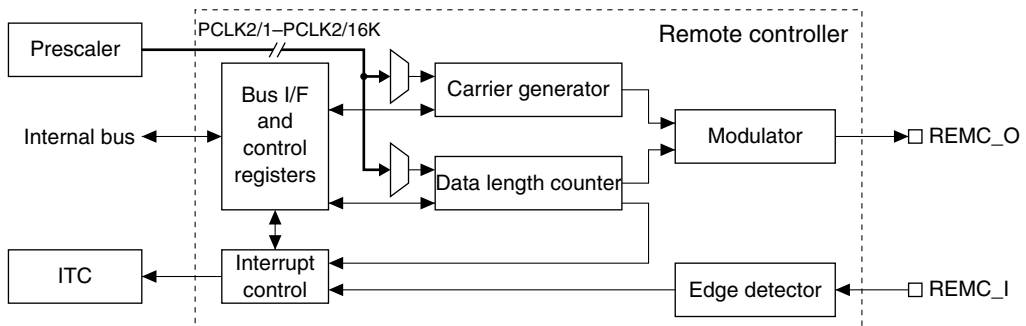


Figure 22.1.1 REMC Module Configuration

22.2 REMC Input/Output Pins

Table 22.2.1 lists the REMC input/output pins.

Table 22.2.1 List of REMC Pins

Pin name	I/O	Qty	Function
REMC_I	I	1	Remote control receive data input pin Inputs receive data.
REMC_O	O	1	Remote control transmit data output pin Outputs modulated remote control transmit data.

The REMC input/output pins (REMC_I, REMC_O) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as REMC input/output pins. For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

22.3 Carrier Generation

The REMC module incorporates a carrier generator that generates a carrier signal for transmission in accordance with the clock set by software and carrier H and L section lengths.

The prescaler (PSC Ch.1) output clock is used for the carrier signal generation clock. The prescaler generates 15 different clocks, dividing the PCLK2 clock by 1 to 16K. One of these clocks is selected by CGCLK[3:0]/REMC_CFG register.

22 REMOTE CONTROLLER (REMC)

Table 22.3.1 Carrier Generation Clock (PCLK2 Division Ratio) Selection

CGCLK[3:0]	Division ratio	CGCLK[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

For more information on prescaler control, see the “Prescaler (PSC)” chapter.

Note: The prescaler must be run before the REMC module can operate.

The carrier H and L section lengths are set by REMCH[5:0]/REMC_CAR register and REMCL[5:0]/REMC_CAR register, respectively. Set a value corresponding to the number of clock (selected as above) cycles + 1 to these registers.

The carrier H and L section lengths can be calculated as follows:

$$\text{Carrier H section length} = \frac{\text{REMCH} + 1}{\text{clk}_{\text{in}}} \text{ [s]}$$

$$\text{Carrier L section length} = \frac{\text{REMCL} + 1}{\text{clk}_{\text{in}}} \text{ [s]}$$

REMCH: Carrier H section length data value

REMCL: Carrier L section length data value

clk_{in}: Prescaler (PSC Ch.1) output clock frequency

The carrier signal is generated from these settings as shown in Figure 22.3.1.

Example: CGCLK[3:0] = 0x2 (PCLK2/4), REMCH[5:0] = 2, REMCL[5:0] = 1

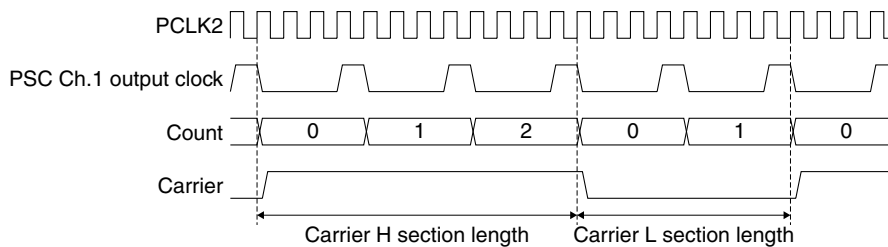


Figure 22.3.1 Carrier Signal Generation

22.4 Data Length Counter Clock Settings

The data length counter is an 8-bit counter for setting data lengths when transmitting data.

When a value corresponding to the data pulse width is written during data transmission, the data length counter begins counting down from that value and stops after generating an underflow interrupt cause when the counter reaches 0. The subsequent transmit data is set using this interrupt.

This counter is also used for data receiving, enabling measurement of the received data length. Interrupts can be generated at the input signal rising or falling edges when receiving data. The data pulse length can be obtained from the difference between data pulse edges by setting the data length counter to 0xff using the interrupt when the input changes and by reading out the count value when a subsequent interrupt occurs due to input changes.

This data length counter count clock also uses a prescaler output clock and can select one of 15 different types. The prescaler output clock is selected by LCCLK[3:0]/REMC_CFG register provided separately to the carrier generation clock select bits.

Table 22.4.1 Data Length Counter Clock (PCLK2 Division Ratio) Selection

LCCLK[3:0]	Division ratio	LCCLK[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

The data length counter can count up to 256. The count clock should be selected to ensure that the data length fits within this range.

22.5 Data Transfer Control

Make the following settings before starting data transfers.

- (1) Configure the carrier signal. (See Section 22.3.)
- (2) Select the data length counter clock. (See Section 22.4.)
- (3) Set the interrupt conditions. (See Section 22.6.)

Note: Make sure the REMC module is halted (REMCN/REMC_CFG register = 0) before changing the above settings.

Data transmission control

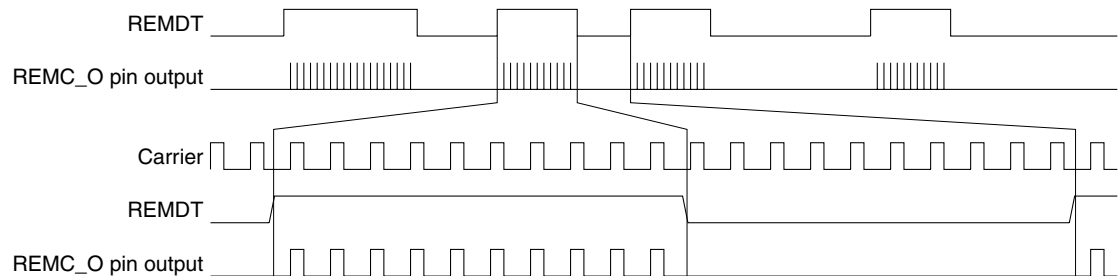


Figure 22.5.1 Data Transmission

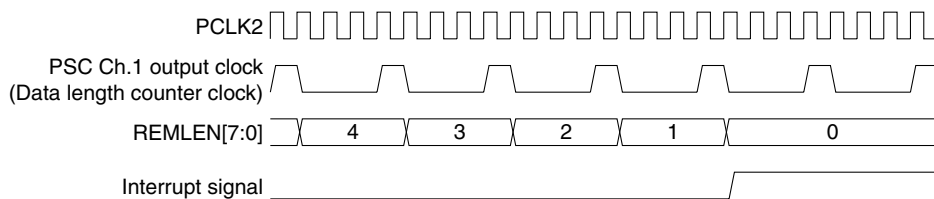


Figure 22.5.2 Underflow Interrupt Generation Timing

(1) Data transmit mode setting

Set REMC to transmit mode by writing 0 to REMMD/REMC_CFG register.

(2) Enabling data transmission

Enable REMC operation by setting REMEN/REMC_CFG register to 1. This initiates REMC transmission.

Set REMDT/REMC_LCNT register to 0 and REMLEN[7:0]/REMC_LCNT register to 0x0 before setting REMEN to 1 to prevent unnecessary data transmission.

(3) Transmission data setting

Set the data to be transmitted (High or Low) to REMDT/REMC_LCNT register.

Setting REMDT to 1 outputs High; setting it to 0 outputs Low from the REMC_O pin after being modulated by the carrier signal.

22 REMOTE CONTROLLER (REMC)

(4) Data pulse length setting

Set the value corresponding to the data pulse length (High or Low section) to `REMLEN[7:0]/REMC_LCNT` register to set to the data length counter.

Given below is the value to which the data length counter is set:

$$\text{Setting value} = \text{Data pulse length (seconds)} \times \text{Prescaler output clock frequency (Hz)}$$

The data length counter starts counting down from the value written using the prescaler output clock selected. A cause of underflow interrupt occurs when the data length counter value reaches 0. If the interrupt is enabled, an REMC interrupt request is output to the interrupt controller (ITC). The data length counter stops counting at the same time with the counter value 0 maintained.

(5) Interrupt handling

To transmit the subsequent data, set the subsequent data (Step 3) and set the data pulse length (Step 4) in the interrupt handler routine executed by the data length counter underflow.

(6) Terminating data transmission

To terminate data transmission, set `REMEN` to 0 after the final data transmission has completed (after an underflow interrupt has occurred).

Data reception control

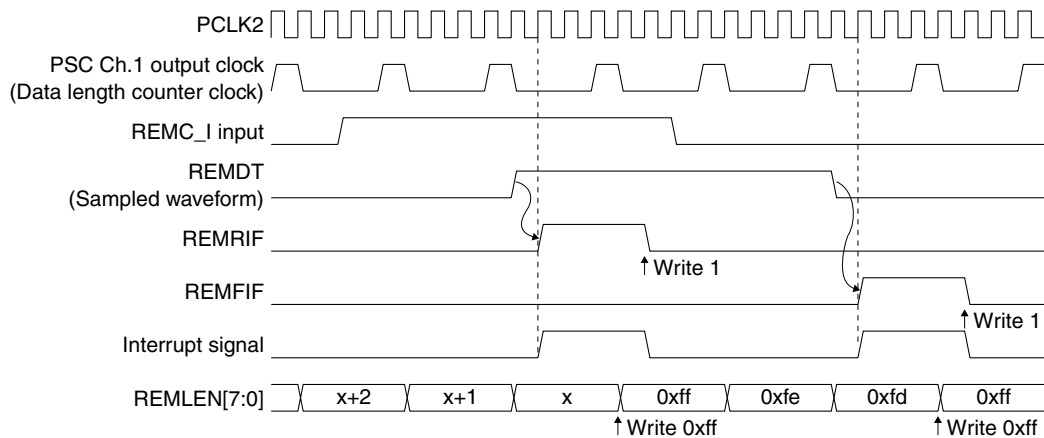


Figure 22.5.3 Data Reception

(1) Data receive mode setting

Set REMC to receive mode by writing 1 to `REMMD/REMC_CFG` register.

(2) Enabling data reception

Enable REMC operation by setting `REMEN/REMC_CFG` register to 1. This initiates REMC reception (input edge detection).

REMC detects an input transition (signal rising or falling edges) by sampling the input signal from the `REMC_I` pin using the prescaler output clock selected for carrier generation. If a signal edge is detected, a cause of rising or falling edge interrupt is generated. An REMC interrupt request is output to the ITC if the interrupt is enabled. Rising edge and falling edge interrupts can be individually enabled or disabled.

Note that if the signal level after the input has changed is not detected for at least two continuous sampling clock cycles, the input signal transition is interpreted as noise, and no rising or falling edge interrupt is generated.

(3) Interrupt handling

When a rising edge or falling edge interrupt occurs, write 0xff to `REMLEN[7:0]/REMC_LCNT` register in the interrupt handler routine to set the value to the data length counter.

The data length counter starts counting down using the selected prescaler output clock from the value written.

The data received can be read out from REMDT/REMC_LCNT register.

The subsequent falling or rising edge interrupt is generated at the termination of the data pulse. Read the data length counter at that point. The data length can be calculated from the difference between 0xff and the value read. To receive the subsequent data, set the data length counter to 0xff once again, then wait for the subsequent interrupt.

If the data length counter becomes 0 after being set to 0xff without the occurrence of an edge interrupt, either no more data is left or a receive error has occurred. Data length counter underflow interrupts are generated even when receiving data and should be used for terminate/error handling.

(4) Terminating data reception

To terminate data reception, write 0 to REMEN after the final data has been received.

22.6 REMC Interrupts

The REMC module includes a function for generating the following three different types of interrupts.

- Underflow interrupt
- Rising edge interrupt
- Falling edge interrupt

The REMC module outputs one interrupt signal shared by the three interrupt causes above to the interrupt controller (ITC). To identify the cause of interrupt occurred, check the interrupt flag status in the REMC module.

Underflow interrupt

Generated when the data length counter has counted down to 0, this interrupt cause sets the interrupt flag REMUIF/REMC_INT register inside the REMC to 1.

When data is being transmitted, the underflow interrupt indicates that the specified data length has been transmitted. When receiving data, the underflow interrupt indicates that data has been received or a receive error has occurred.

To use this interrupt, set REMUIE/REMC_INT register to 1. If REMUIE is set to 0 (default), the interrupt request attributable to this cause will not be sent to the ITC.

When REMUIF is set to 1, REMC outputs an interrupt request to the ITC. An interrupt will be generated if the ITC and C33 PE Core interrupt conditions are met.

REMUIF should be inspected in the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to data length counter underflow.

The interrupt cause should be cleared in the interrupt handler routine by resetting (writing 1 to) REMUIF.

Rising edge interrupt

Generated when the REMC_I pin input signal changes from Low to High, this interrupt cause sets the interrupt flag REMRIF/REMC_INT register to 1 within the REMC.

By running the data length counter between this interrupt and a falling edge interrupt when data is being received, the received data pulse width can be calculated from that count value.

To use this interrupt, set REMRIE/REMC_INT register to 1. If REMRIE is set to 0 (default), the interrupt request attributable to this cause will not be sent to the ITC.

When REMRIF is set to 1, REMC outputs an interrupt request to the ITC. An interrupt will be generated if the ITC and C33 PE Core interrupt conditions are met.

REMRIF should be inspected in the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to input signal rising edge.

The interrupt cause should be cleared in the interrupt handler routine by resetting (writing 1 to) REMRIF.

Falling edge interrupt

Generated when the REMC_I pin input signal changes from High to Low, this interrupt cause sets the interrupt flag REMFIF/REMC_INT register to 1 within the REMC.

By running the data length counter between this interrupt and a rising edge interrupt when data is being received, the received data pulse width can be calculated from that count value.

22 REMOTE CONTROLLER (REMC)

To use this interrupt, set REMFIE/REMC_INT register to 1. If REMFIE is set to 0 (default), the interrupt request attributable to this cause will not be sent to the ITC.

When REMFIF is set to 1, REMC outputs an interrupt request to the ITC. An interrupt will be generated if the ITC and C33 PE Core interrupt conditions are met.

REMFIF should be inspected in the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to input signal falling edge.

The interrupt cause should be cleared in the interrupt handler routine by resetting (writing 1 to) REMFIF.

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

22.7 Control Register Details

Table 22.7.1 List of REMC Registers

Address	Register name		Function
0x301500	REMC_CFG	REMC Configuration Register	Controls the clock and data transfer.
0x301502	REMC_CAR	REMC Carrier Length Setup Register	Sets the carrier H/L section lengths.
0x301504	REMC_LCNT	REMC Length Counter Register	Sets the transmit/receive data length.
0x301506	REMC_INT	REMC Interrupt Control Register	Controls interrupts.

The REMC registers are described in detail below. These are 16-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

REMC Configuration Register (REMC_CFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
REMC Configuration Register (REMC_CFG)	0x301500 (16 bits)	D15–12	CGCLK[3:0]	Carrier generator clock division ratio select (Prescaler output clock)	CGCLK[3:0] LCCLK[3:0] 0xf reserved 0xe 1/16384 0xd 1/8192 0xc 1/4096 0xb 1/2048 0xa 1/1024 0x9 1/512 0x8 1/256	0x0	R/W	Source clock = PCLK2		
		D11–8	LCCLK[3:0]	Length counter clock division ratio select (Prescaler output clock)	0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W			
		D7–2	–	reserved	–	–	–		–	0 when being read.
		D1	REMMD	REMC mode select	1 Receive	0 Transmit	0		R/W	
		D0	REMEM	REMC enable	1 Enable	0 Disable	0		R/W	

D[15:12] CGCLK[3:0]: Carrier Generator Clock Division Ratio Select Bits

Selects a carrier generation clock from the 15 prescaler (PSC Ch.1) output clocks (PCLK2 division ratio).

Table 22.7.2 Carrier Generation Clock (PCLK2 Division Ratio) Selection

CGCLK[3:0]	Division ratio	CGCLK[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

D[11:8] LCCLK[3:0]: Length Counter Clock Division Ratio Select Bits

Selects a data length counter clock from the 15 prescaler (PSC Ch.1) output clocks (PCLK2 division ratio).

Table 22.7.3 Data Length Counter Clock (PCLK2 Division Ratio) Selection

LCCLK[3:0]	Division ratio	LCCLK[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

Note: The clock should be set only while the REMC module is stopped (REMEM = 0).

D[7:2] Reserved**D1 REMMD: REMC Mode Select Bit**

Selects the transfer direction.

1 (R/W): Reception

0 (R/W): Transmission (default)

D0 REMEN: REMC Enable Bit

Enables or disables data transfer by the REMC module.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Setting REMEN to 1 starts transmission or receiving in accordance with REMMD settings.

Setting REMEN to 0 disables REMC module operations.

REMC Carrier Length Setup Register (REMC_CAR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Carrier Length Setup Register (REMC_CAR)	0x301502 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–8	REMCL[5:0]	Carrier L length setup	0x0 to 0x3f	0x0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	REMCH[5:0]	Carrier H length setup	0x0 to 0x3f	0x0	R/W	

D[15:14] Reserved**D[13:8] REMCL[5:0]: Carrier L Length Setup Bits**

Sets the carrier signal L section length. (Default: 0x0)

Specify a value corresponding to the number of carrier generation clock cycles selected by CGCLK[3:0]/REMC_CFG register + 1. Calculate carrier L section length as follows:

$$\text{Carrier L section length} = \frac{\text{REMCL} + 1}{\text{clk}_{\text{in}}} [\text{s}]$$

REMCL: REMCL[5:0] setting

clk_{in}: Prescaler (PSC Ch.1) output clock frequency

The H section length is specified by REMCH[5:0]. The carrier signal is generated from these settings as shown in Figure 22.7.1.

D[7:6] Reserved**D[5:0] REMCH[5:0]: Carrier H Length Setup Bits**

Sets the carrier signal H section length. (Default: 0x0)

Specify a value corresponding to the number of carrier generation clock cycles selected by CGCLK[3:0]/REMC_CFG register + 1. Calculate carrier H section length as follows:

$$\text{Carrier H section length} = \frac{\text{REMCH} + 1}{\text{clk}_{\text{in}}} [\text{s}]$$

REMCH: REMCH[5:0] setting

clk_{in}: Prescaler (PSC Ch.1) output clock frequency

22 REMOTE CONTROLLER (REMC)

The L section length is specified by REMCL[5:0]. The carrier signal is generated from these settings as shown in Figure 22.7.1.

Example: CGCLK[3:0] = 0x2 (PCLK2/4), REMCH[5:0] = 2, REMCL[5:0] = 1

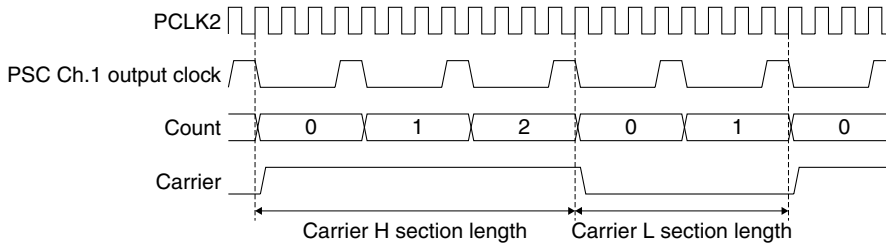


Figure 22.7.1 Carrier Signal Generation

REMC Length Counter Register (REMC_LCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Length Counter Register (REMC_LCNT)	0x301504 (16 bits)	D15–8	REMLEN[7:0]	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	REMDT	Transmit/receive data	1 1 (H) 0 0 (L)	0	R/W	

D[15:8] REMLEN[7:0]: Transmit/Receive Data Length Count Bits

Sets the data length counter value and starts counting. (Default: 0x0)

The counter stops when it reaches 0 and generates a cause of underflow interrupt.

For data transmission

Set the transmit data length for data transmission.

When a value corresponding to the data pulse width is written, the data length counter starts counting down from that value. The counter stops counting and generates a cause of underflow interrupt when it reaches 0. Set the subsequent transmit data using this interrupt.

For data receiving

Interrupts can be generated at the input signal rising or falling edges when receiving data. The data pulse length can be obtained from the difference between 0xff set to the data length counter using the interrupt when the input changes and the count value read out when the next interrupt occurs due to an input change.

D[7:1] Reserved

D0 REMDT: Transmit/Receive Data Bit

Sets the transmit data for data transmission. Receive data can be read when receiving data.

1 (R/W): 1 (H)

0 (R/W): 0 (L) (default)

If REMEN/REMC_CFG register is set to 1, the REMDT setting is modulated by the carrier signal for data transmission and output from the REMC_O pin. For data receiving, this bit is set to the value corresponding to the signal level of the data pulse input.

REMC Interrupt Control Register (REMC_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
REMC Interrupt Control Register (REMC_INT)	0x301506 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.	
		D10	REMFIF	Falling edge interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	REMRIF	Rising edge interrupt flag	–	–	0	R/W	
		D8	REMUIF	Underflow interrupt flag	–	–	0	R/W	
		D7–3	–	reserved	–	–	–	–	0 when being read.
		D2	REMFIE	Falling edge interrupt enable	1 Enable	0 Disable	0	R/W	
		D1	REMRIE	Rising edge interrupt enable	1 Enable	0 Disable	0	R/W	
		D0	REMUIE	Underflow interrupt enable	1 Enable	0 Disable	0	R/W	

This register controls the data length counter underflow, input signal rising edge, and input signal falling edge interrupts. The interrupt flag is set to 1 when the data length counter underflows, or when an input signal rising edge or falling edge is detected. If the corresponding interrupt enable bit has been set to 1, the REMC outputs an interrupt request signal to the ITC at the same time. An interrupt will be generated if the ITC and C33 PE Core interrupt conditions are met. When an REMC interrupt occurs, check the interrupt flag status in this register to identify the cause of interrupt occurred. If the interrupt enable bit is set to 0, the interrupt is disabled.

- Notes:**
- To prevent interrupt recurrences, the REMC module interrupt flag must be reset in the interrupt handler routine after an REMC interrupt has occurred.
 - To prevent generating unnecessary interrupts, reset the interrupt flag before enabling interrupts by the interrupt enable bit.

D[15:11] Reserved

D10 REMFIF: Falling Edge Interrupt Flag Bit

Indicates the falling edge interrupt cause occurrence status.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

REMFIF is set to 1 at the input signal falling edge. REMFIF is reset to 0 by writing 1.

D9 REMRIF: Rising Edge Interrupt Flag Bit

Indicates the rising edge interrupt cause occurrence status.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

REMRIF is set to 1 at the input signal rising edge. REMRIF is reset to 0 by writing 1.

D8 REMUIF: Underflow Interrupt Flag Bit

Indicates the underflow interrupt cause occurrence status.

- 1 (R): Cause of interrupt has occurred
- 0 (R): No cause of interrupt has occurred (default)
- 1 (W): Flag is reset
- 0 (W): Ignored

REMUIF is set to 1 when a data length counter underflow occurs. REMUIF is reset to 0 by writing 1.

D[7:3] Reserved

D2 REMFIE: Falling Edge Interrupt Enable Bit

Enables or disables input signal falling edge interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

D1 REMRIE: Rising Edge Interrupt Enable Bit

Enables or disables input signal rising edge interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

D0 REMUIE: Underflow Interrupt Enable Bit

Enables or disables data length counter underflow interrupts.

- 1 (R/W): Interrupt enabled
- 0 (R/W): Interrupt disabled (default)

23 Card Interface (CARD)

23.1 CARD Module Overview

The S1C33L26 includes a card interface (CARD) module to connect a NAND Flash or SmartMedia cards. The following shows the features of the CARD module:

- Generates the #NAND_RD and #NAND_WR signals.
(Use general-purpose input/output ports to control the signals specific to NAND Flash or SmartMedia card.)
- Supports 8-bit and 16-bit NAND Flash devices and SmartMedia cards.
- A NAND Flash or SmartMedia cards can be connected to the #CE9 area (Area 9 or Area 22).
- The data and address signals of the device can be connected directly to the external bus of the SRAMC.

23.2 CARD Output Pins

Table 23.2.1 lists the card interface output pins.

Table 23.2.1 List of Card Interface Pins

Pin name	I/O	Qty	Function
#NAND_RD	O	1	SmartMedia read signal output pin This pin outputs the read signal for NAND Flash and SmartMedia card.
#NAND_WR	O	1	SmartMedia write signal output pin This pin outputs the write signal for NAND Flash and SmartMedia card.

The CARD output pins (#NAND_RD, #NAND_WR) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as CARD output pins. For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

23.3 Card Interface Control Signals

Figure 23.3.1 shows the logic used to generate SmartMedia interface signals. Figure 23.3.2 shows an example of connecting the S1C33L26 and a SmartMedia card (NAND Flash).

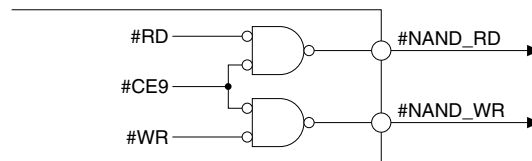


Figure 23.3.1 Card Interface Signal Generation Circuit

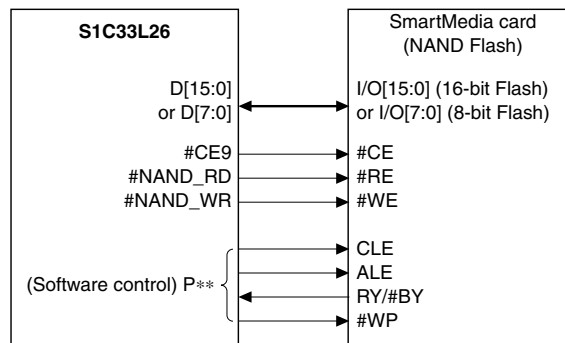


Figure 23.3.2 Example of Connecting a SmartMedia Card

24 I/O Ports (GPIO)

24.1 GPIO Module Overview

The S1C33L26 includes general-purpose I/O ports that allow software to switch input/output direction. These share internal peripheral module input/output pins, but pins not used for peripheral modules can be used as general-purpose I/O ports.

The following shows the features of the GPIO module:

- Maximum 71 I/O ports (P0[7:0], P1[7:0], P2[1:0], P3[6:0], P4[2:0], P5[6:0], P6, P8[3:0], P9[7:0], PA[6:0], PB[7:0], PC[7:0]) and six input ports (P7[5:0]) are available in the TQFP24-144pin or PFBGA12U-180 package model.

Maximum 56 I/O ports (P0[7:0], P1[7:0], P2[1:0], P3[6:0], P4[2:0], P5[6:0], P6, P8[3:0], P9[7:0], PC[7:0]) and six input ports (P7[5:0]) are available in the TQFP15-128pin package model.

* The GPIO ports are shared with other peripheral function pins (USI, PWM etc.). Therefore, the number of GPIO ports depends on the peripheral functions used.

- Can generate input interrupts from the 16 of 64 ports selected via software.
- Interrupt input signal conditions (level or edge trigger, and polarity) can be specified.
- The input interrupt circuit includes chattering filters.
- All port provide a port function select bit to configure the pin function (for GPIO or peripheral functions).

Figure 24.1.1 shows the I/O port configuration.

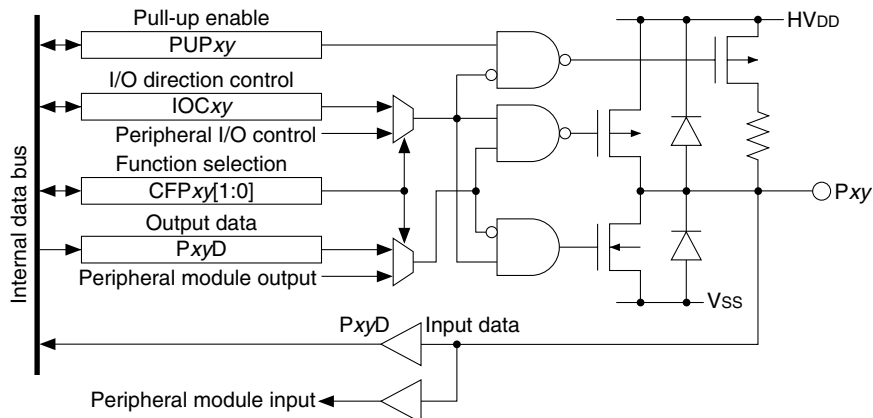


Figure 24.1.1 I/O Port Configuration

Notes:

- The PCLK2 clock must be supplied from the CMU to access the I/O port. The prescaler (PSC Ch.1) output clock is also needed to operate the chattering filters. Turn on the prescaler when using this function.

- The “xy” in the register and bit names refers to the port number (Pxy, x = 0 to C, y = 0 to 7).

Example: PxyD/GPIO_Px_DAT register

P00: P00D/GPIO_P0_DAT register

P17: P17D/GPIO_P1_DAT register

24.2 Input/Output Pin Function Selection (Port MUX)

The I/O port pins share peripheral module input/output pins. Each pin can be configured for use as an I/O port or for a peripheral module function via the corresponding port function select bits. Pins not used for peripheral modules can be used as general-purpose I/O ports.

Table 24.2.1 Input/Output Pin Function Selection

128pin package	Pin function 1 CFPxy[1:0] = 0x0 (default)	Pin function 2 CFPxy[1:0] = 0x1	Pin function 3 CFPxy[1:0] = 0x2	Pin function 4 CFPxy[1:0] = 0x3	Port function select bit
○	P00	USI_DI	SIN1	#NAND_WR	CFP00[1:0]/PMUX_P0_03 register
○	P01	USI_DO	SOUT1	#NAND_RD	CFP01[1:0]/PMUX_P0_03 register
○	P02	USI_CS	SCLK1	REMC_O	CFP02[1:0]/PMUX_P0_03 register
○	P03	USI_CK	#SRDY1	REMC_I	CFP03[1:0]/PMUX_P0_03 register
○	P04	SIN0	I2S_SDO	T16A_ATMA_0	CFP04[1:0]/PMUX_P0_47 register
○	P05	SOUT0	I2S_WS	T16A_ATMB_0	CFP05[1:0]/PMUX_P0_47 register
○	P06	SCLK0	I2S_SCLK	PWM_H	CFP06[1:0]/PMUX_P0_47 register
○	P07	#SRDY0	I2S_MCLK	PWM_L	CFP07[1:0]/PMUX_P0_47 register
○	P10	USIL_DI	FPDAT8		CFP10[1:0]/PMUX_P1_03 register
○	P11	USIL_DO	FPDAT9		CFP11[1:0]/PMUX_P1_03 register
○	P12	USIL_CS	FPDAT10	T16A_ATMA_1	CFP12[1:0]/PMUX_P1_03 register
○	P13	USIL_CK	FPDAT11	T16A_ATMB_1	CFP13[1:0]/PMUX_P1_03 register
○	P14	FPDAT19	FPDAT12	CMU_CLK	CFP14[1:0]/PMUX_P1_47 register
○	DST0	P15	FPDAT13		CFP15[1:0]/PMUX_P1_47 register
○	DST1	P16	FPDAT14		CFP16[1:0]/PMUX_P1_47 register
○	DPCO	P17	FPDAT15		CFP17[1:0]/PMUX_P1_47 register
○	SDCKE	P20			CFP20[1:0]/PMUX_P2_01 register
○	SDCLK	P21			CFP21[1:0]/PMUX_P2_01 register
○	P30		TFT_CTL0	T16A_ATMA_0	CFP30[1:0]/PMUX_P3_03 register
○	P31		TFT_CTL1	T16A_ATMB_0	CFP31[1:0]/PMUX_P3_03 register
○	P32		TFT_CTL2	REMC_O	CFP32[1:0]/PMUX_P3_03 register
○	P33		TFT_CTL3	REMC_I	CFP33[1:0]/PMUX_P3_03 register
○	DCLK	P34			CFP34[1:0]/PMUX_P3_46 register
○	DSIO	P35			CFP35[1:0]/PMUX_P3_46 register
○	DST2	P36			CFP36[1:0]/PMUX_P3_46 register
○	A21	P40	FPDAT18	#NAND_RD	CFP40[1:0]/PMUX_P4_02 register
○	A22	P41	FPDAT17	#NAND_WR	CFP41[1:0]/PMUX_P4_02 register
○	A23	P42	FPDAT16		CFP42[1:0]/PMUX_P4_02 register
○	#CE7	P50	#SDCS		CFP50[1:0]/PMUX_P5_03 register
○	#CE8	P51	#CE4		CFP51[1:0]/PMUX_P5_03 register
○	#CE9	P52	#CE5		CFP52[1:0]/PMUX_P5_03 register
○	#CE10	P53			CFP53[1:0]/PMUX_P5_03 register
○	#RD	P54			CFP54[1:0]/PMUX_P5_46 register
○	#WRL	P55			CFP55[1:0]/PMUX_P5_46 register
○	#WRH/#BSH	P56			CFP56[1:0]/PMUX_P5_46 register
○	P60	#WAIT	WDT_CLK	#WDT_NMI	CFP60[1:0]/PMUX_P6_0 register
○	P70	AIN0	T16A_EXCL_0		CFP70[1:0]/PMUX_P7_03 register
○	P71	AIN1	T16A_EXCL_1		CFP71[1:0]/PMUX_P7_03 register
○	P72	AIN2	PWM_EXCL		CFP72[1:0]/PMUX_P7_03 register
○	P73	AIN3			CFP73[1:0]/PMUX_P7_03 register
○	P74	AIN4			CFP74[1:0]/PMUX_P7_45 register
○	P75	AIN5	#WAIT	#ADTRIG	CFP75[1:0]/PMUX_P7_45 register
○	P80	FPFRAME	USIL_CS		CFP80[1:0]/PMUX_P8_03 register
○	P81	FPLINE	USIL_CK		CFP81[1:0]/PMUX_P8_03 register
○	P82	FPSHIFT	USIL_DI		CFP82[1:0]/PMUX_P8_03 register
○	P83	FPDRDY	USIL_DO		CFP83[1:0]/PMUX_P8_03 register
○	P90	FPDAT0	LCD_D0	SIN0	CFP90[1:0]/PMUX_P9_03 register
○	P91	FPDAT1	LCD_D1	SOUT0	CFP91[1:0]/PMUX_P9_03 register
○	P92	FPDAT2	LCD_D2	SCLK0	CFP92[1:0]/PMUX_P9_03 register
○	P93	FPDAT3	LCD_D3	#SRDY0	CFP93[1:0]/PMUX_P9_03 register
○	P94	FPDAT4	LCD_D4		CFP94[1:0]/PMUX_P9_47 register
○	P95	FPDAT5	LCD_D5		CFP95[1:0]/PMUX_P9_47 register
○	P96	FPDAT6	LCD_D6		CFP96[1:0]/PMUX_P9_47 register
○	P97	FPDAT7	LCD_D7		CFP97[1:0]/PMUX_P9_47 register
N/A	PA0	SIN1	FPDAT16	FPDAT20	CFPA0[1:0]/PMUX_PA_03 register

128pin package	Pin function 1 CFP _{xy} [1:0] = 0x0 (default)	Pin function 2 CFP _{xy} [1:0] = 0x1	Pin function 3 CFP _{xy} [1:0] = 0x2	Pin function 4 CFP _{xy} [1:0] = 0x3	Port function select bit
N/A	PA1	SOUT1	FPDAT17	FPDAT21	CFPA1[1:0]/PMUX_PA_03 register
N/A	PA2	SCLK1	FPDAT18	FPDAT22	CFPA2[1:0]/PMUX_PA_03 register
N/A	PA3	#SRDY1	FPDAT19	FPDAT23	CFPA3[1:0]/PMUX_PA_03 register
N/A	A24	PA4	T16A_ATMA_1	REMC_O	CFPA4[1:0]/PMUX_PA_46 register
N/A	A25	PA5	T16A_ATMB_1	REMC_I	CFPA5[1:0]/PMUX_PA_46 register
N/A	PA6	#ADTRIG			CFPA6[1:0]/PMUX_PA_46 register
N/A	PB0	FPDAT8	I2S_SDO		CFPB0[1:0]/PMUX_PB_03 register
N/A	PB1	FPDAT9	I2S_WS		CFPB1[1:0]/PMUX_PB_03 register
N/A	PB2	FPDAT10	I2S_SCLK	PWM_H	CFPB2[1:0]/PMUX_PB_03 register
N/A	PB3	FPDAT11	I2S_MCLK	PWM_L	CFPB3[1:0]/PMUX_PB_03 register
N/A	PB4	FPDAT12		FPDAT20	CFPB4[1:0]/PMUX_PB_47 register
N/A	PB5	FPDAT13		FPDAT21	CFPB5[1:0]/PMUX_PB_47 register
N/A	PB6	FPDAT14		FPDAT22	CFPB6[1:0]/PMUX_PB_47 register
N/A	PB7	FPDAT15		FPDAT23	CFPB7[1:0]/PMUX_PB_47 register
○	D8	PC0			CFPC0[1:0]/PMUX_PC_03 register
○	D9	PC1			CFPC1[1:0]/PMUX_PC_03 register
○	D10	PC2			CFPC2[1:0]/PMUX_PC_03 register
○	D11	PC3			CFPC3[1:0]/PMUX_PC_03 register
○	D12	PC4			CFPC4[1:0]/PMUX_PC_47 register
○	D13	PC5			CFPC5[1:0]/PMUX_PC_47 register
○	D14	PC6			CFPC6[1:0]/PMUX_PC_47 register
○	D15	PC7			CFPC7[1:0]/PMUX_PC_47 register

At initial reset, each I/O port pin (P_{xy}) is initialized for the default function (“Pin function 1” in Table 24.2.1).

For information on functions other than the I/O ports, see “Pin Descriptions” in the “Overview” chapter or the descriptions of peripheral modules.

The sections below describe port functions with the pins set as general-purpose I/O ports.

Note: The port function select registers (PMUX_Px_*) are write-protected. Before these registers can be rewritten, the write protection must be removed by writing data 0x96 to PPROT[7:0]/GPIO_PROTECT register. Note that since unnecessary rewrites to the port function select registers could lead to erratic system operation, PPROT[7:0] should be set to other than 0x96 unless the port function select registers must be rewritten.

24.3 Data Input/Output

Data input/output control

The I/O ports allow selection of the data input/output direction for each bit using IOC_{xy}/GPIO_Px_IOC register. Set IOC_{xy} to 0 (default) to configure the P_{xy} port for input; set IOC_{xy} to 1 to configure the P_{xy} port for output.

The input/output direction of ports with a peripheral module function selected is controlled by the peripheral module. IOC_{xy} settings are ignored.

Data input

To input an external signal and read out the value, set IOC_{xy} to 0 (input mode, default). The I/O port is placed into high-impedance status and it functions as an input port.

In input mode, the external signal level can be read out directly from P_{xy}D/GPIO_Px_DAT register. The value read will be 1 when the input pin is at high level and 0 when it is at low level.

The port pin status can also be read in output mode (IOC_{xy} = 1). In this case, the value actually output the port can be read out from P_{xy}D.

Data output

To output data from the port pin, set IOC_{xy} to 1 (output mode). The I/O port then functions as an output port, and the value set in the P_{xy}D is output from the port pin. The port pin outputs high level when P_{xy}D is set to 1 and low level when set to 0. Writing to P_{xy}D is possible without affecting pin status, even in input mode.

24.4 Pull-up Control

The I/O port contains a pull-up resistor that can be enabled or disabled individually for each bit using PUP_{xy}/GPIO_Px_PUP register. Setting PUP_{xy} to 1 enables the pull-up resistor and pulls up the port pin in input mode. It will not be pulled up if set to 0. The PUP_{xy} setting is ignored and not pulled up in output mode.

Unused I/O ports should be set with pull-up enabled.

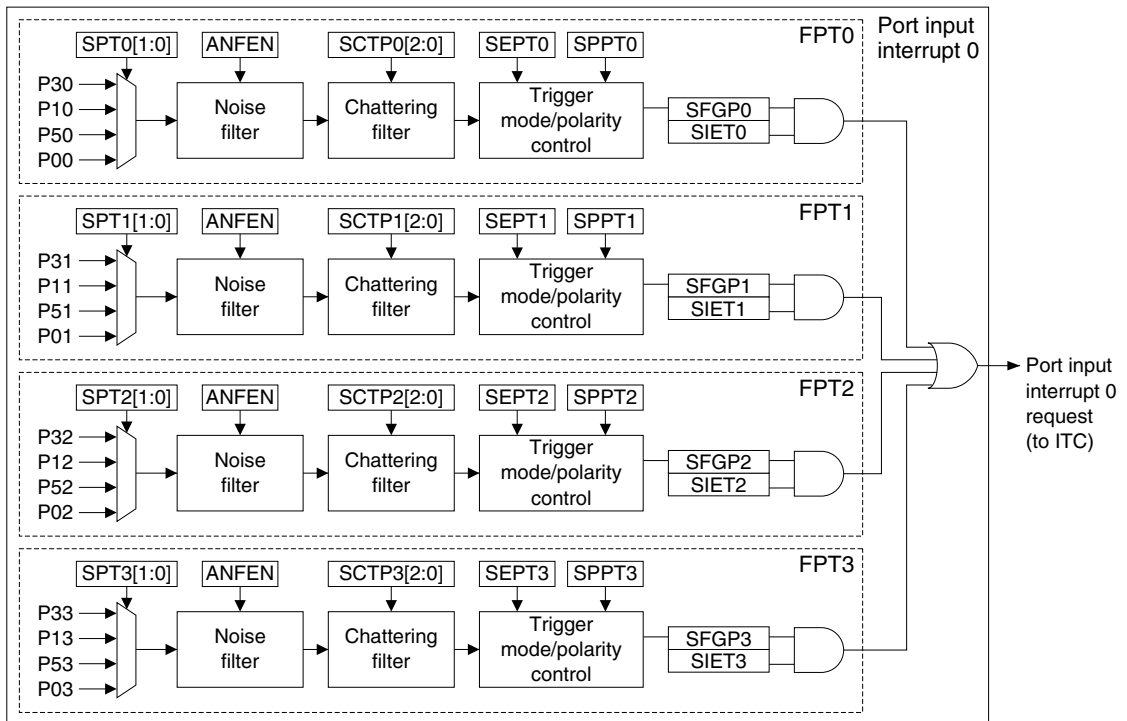
This pull-up setting is also enabled for ports for which the peripheral module function has been selected.

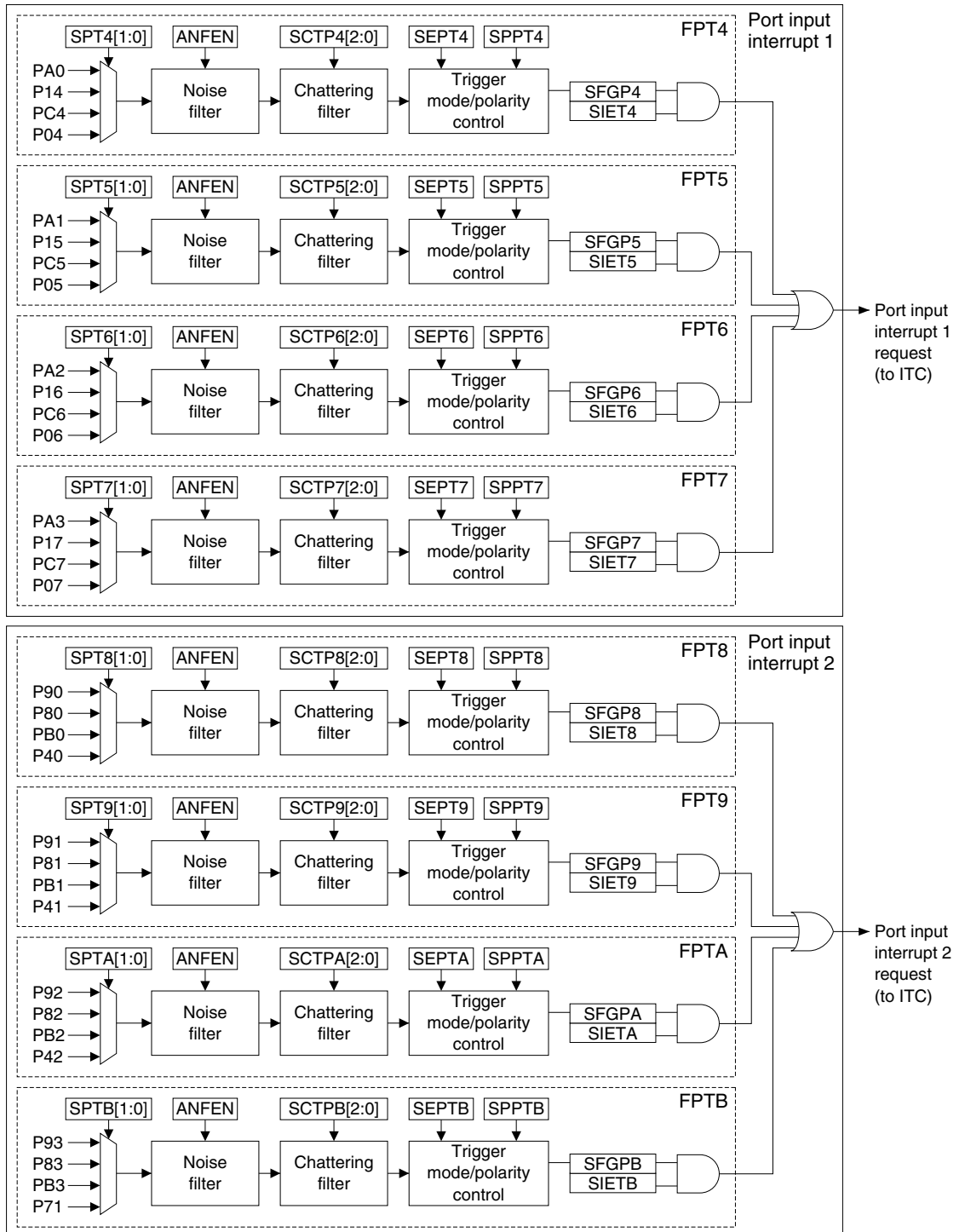
24.5 Port Input Interrupt and DMA

The GPIO module has four interrupt systems (port input interrupts 0 to 3) and the ports can be selected for generating each cause of interrupt.

The interrupt trigger conditions can also be selected from between input signal edge (rising edge or falling edge) and input signal level (high level or low level).

Figure 24.5.1 shows the configuration of the port interrupt circuit.





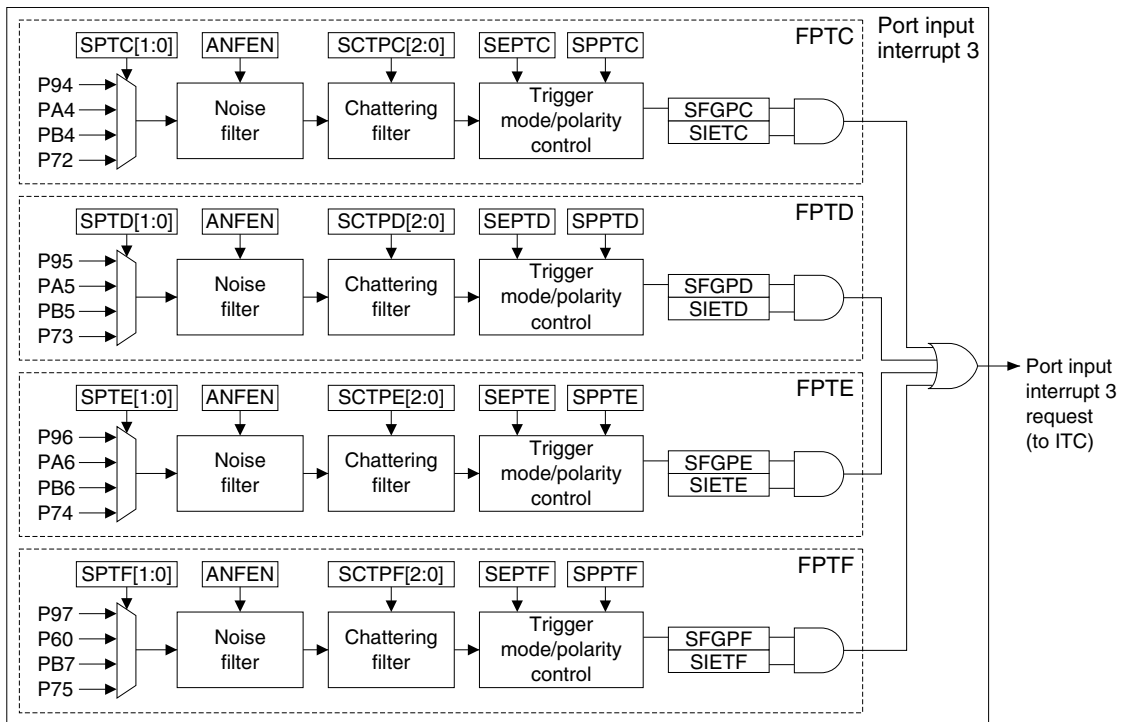


Figure 24.5.1 Port Input Interrupt Circuit Configuration

Interrupt port selection

Each port input interrupt system uses four ports to generate interrupts. Select the ports using SPTn[1:0]/GPIO_FPTnn_SEL register (n = 0 to F, nn = 03, 47, 8B, or CF).

Table 24.5.1 Selecting Ports Used For Port Interrupt 0

SPTn[1:0] setting	FPT0 (SPT0[1:0])	FPT1 (SPT1[1:0])	FPT2 (SPT2[1:0])	FPT3 (SPT3[1:0])
0x3	P30	P31	P32	P33
0x2	P10	P11	P12	P13
0x1	P50	P51	P52	P53
0x0 (default)	P00	P01	P02	P03

Table 24.5.2 Selecting Ports Used For Port Interrupt 1

SPTn[1:0] setting	FPT4 (SPT4[1:0])	FPT5 (SPT5[1:0])	FPT6 (SPT6[1:0])	FPT7 (SPT7[1:0])
0x3	PA0	PA1	PA2	PA3
0x2	P14	P15	P16	P17
0x1	PC4	PC5	PC6	PC7
0x0 (default)	P04	P05	P06	P07

Table 24.5.3 Selecting Ports Used For Port Interrupt 2

SPTn[1:0] setting	FPT8 (SPT8[1:0])	FPT9 (SPT9[1:0])	FPTA (SPTA[1:0])	FPTB (SPTB[1:0])
0x3	P90	P91	P92	P93
0x2	P80	P81	P82	P83
0x1	PB0	PB1	PB2	PB3
0x0 (default)	P40	P41	P42	P71

Table 24.5.4 Selecting Ports Used For Port Interrupt 3

SPTn[1:0] setting	FPTC (SPTC[1:0])	FPTD (SPTD[1:0])	FPTE (SPTE[1:0])	FPTF (SPTF[1:0])
0x3	P94	P95	P96	P97
0x2	PA4	PA5	PA6	P60
0x1	PB4	PB5	PB6	PB7
0x0 (default)	P72	P73	P74	P75

Interrupt mode and polarity selection

The GPIO module provides two interrupt modes to set the interrupt flags: edge trigger mode and level trigger mode. The interrupt mode for each FPT line can be selected using $SEPTn/GPIO_FPTnn_MOD$ register.

When $SEPTn$ bit is set to 1 (default), the corresponding port is set to edge trigger mode. In edge trigger mode, the interrupt bit is set at the active edge of the input signal and it retains 1 until reset via software.

When $SEPTn$ is set to 0, the corresponding port is set to level trigger mode. In level trigger mode, the interrupt flag is set when the input signal goes the active level and it retains 1 until reset via software.

SLEEP mode can be canceled by causing a port input interrupt regardless of how the GPIO interrupt mode (edge trigger/level trigger) is set.

The active level/edge of the input signal can be selected using $SPPTn/GPIO_FPTnn_POL$ register.

When $SPPTn$ is set to 1 (default), high level (in level trigger mode) or rising edge (in edge trigger mode) is selected.

When $SPPTn$ is set to 0, low level (in level trigger mode) or falling edge (in edge trigger mode) is selected.

Table 24.5.5 Port input Interrupt Conditions

$SEPTn$	$SPPTn$	Port input interrupt condition
1	1	Rising edge input
1	0	Falling edge input
0	1	High level input
0	0	Low level input

Interrupt flags

The port interrupt circuit provides 16 interrupt flags ($SFGPn/GPIO_FPTnn_FLG$ register) corresponding to the FPT interrupt ports.

In level trigger mode, the interrupt flag is set according to the input signal level.

In edge trigger mode, the interrupt flag is set at the active edge of the input signal.

The interrupt flag must be reset by writing 1 after an interrupt occurs.

Interrupt enable bits

Each FPT interrupt port can be enabled or disabled to generate interrupts using the corresponding interrupt enable bit ($SIETn/GPIO_FPTnn_MSK$ register).

To use port input interrupts, the interrupt port pins must be configured as an I/O port using the corresponding port function select bits. Before setting $SIETn$ to 1, the corresponding $SFGPn$ must be cleared to 0.

To enable interrupts, set $SIETn$ to 1. To disable interrupts, set $SIETn$ to 0.

When $SFGPn$ is set to 1 while the corresponding $SIETn$ is set to 1, an interrupt request signal is output to the ITC. An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied.

For specific information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

DMA trigger

An FPT interrupt port can be selected as a DMA trigger source using the $SPTRG[3:0]/GPIO_DMA$ register.

Table 24.5.6 DMA Trigger Source Selection

$SPTRG[3:0]$	Trigger source	$SPTRG[3:0]$	Trigger source
0xf	FPTF	0x7	FPT7
0xe	FPT E	0x6	FPT6
0xd	FPTD	0x5	FPT5
0xc	FPTC	0x4	FPT4
0xb	FPTB	0x3	FPT3
0xa	FPTA	0x2	FPT2
0x9	FPT9	0x1	FPT1
0x8	FPT8	0x0	FPT0

(Default: 0x0)

The interrupt signal of the selected FPT line, which is generated according to the interrupt mode and polarity settings regardless of the $SIETn$ setting (even if the interrupt is disabled), is sent to the DMAC to trigger a DMA transfer. For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

Note: After setting SCTPn[2:0]/GPIO_FPTnn_CHAT register, the port may send an undesired DMA trigger to the DMAC. Disable the port DMA request for at least the wait time shown below after the GPIO_FPTnn_CHAT register is set.

$$\text{Wait time } [\mu\text{s}] = \text{Filter sampling time } [\mu\text{s}] \times 4$$

Example: When the filter sampling time is $64/f_{\text{PCLK2}}$ and $f_{\text{PCLK2}} = 32 \text{ MHz}$

$$\text{Wait time} = 64 \times 4 / 32 = 8 \text{ } [\mu\text{s}]$$

The port DMA function can be used after waiting 8 μs or more when the GPIO_FPTnn_CHAT register is set to 0x7.

Chattering Filters

Each FPT line includes a chattering filter circuit for key entry that can be disabled or enabled with a sampling clock specified using SCTPn[2:0]/GPIO_FPTnn_CHAT register ($n = 0$ to F, $nn = 01, 23, 45, 67, 89, AB, CD,$ or EF).

When the chattering filter is enabled, glitches shorter than the filter sampling time $\times 2$ will be filtered as noise. Pulses longer than the filter sampling time $\times 4$ will not be filtered and can generate input port interrupts. If filter sampling time $\times 2 \leq$ pulse width \leq filter sampling time $\times 4$, the pulse may be filtered depending on the input timing.

Table 24.5.7 Chattering Filter Settings

SCTPn[2:0]	Filter sampling time	Invalid pulse (glitch) width that will be filtered	Valid pulse width that will be accepted
0x7	$64/f_{\text{PCLK2}}$	$< 64/f_{\text{PCLK2}} \times 2$	$> 64/f_{\text{PCLK2}} \times 4$
0x6	$32/f_{\text{PCLK2}}$	$< 32/f_{\text{PCLK2}} \times 2$	$> 32/f_{\text{PCLK2}} \times 4$
0x5	$16/f_{\text{PCLK2}}$	$< 16/f_{\text{PCLK2}} \times 2$	$> 16/f_{\text{PCLK2}} \times 4$
0x4	$8/f_{\text{PCLK2}}$	$< 8/f_{\text{PCLK2}} \times 2$	$> 8/f_{\text{PCLK2}} \times 4$
0x3	$4/f_{\text{PCLK2}}$	$< 4/f_{\text{PCLK2}} \times 2$	$> 4/f_{\text{PCLK2}} \times 4$
0x2	$2/f_{\text{PCLK2}}$	$< 2/f_{\text{PCLK2}} \times 2$	$> 2/f_{\text{PCLK2}} \times 4$
0x1	$1/f_{\text{PCLK2}}$	$< 1/f_{\text{PCLK2}} \times 2$	$> 1/f_{\text{PCLK2}} \times 4$
0x0	Not filtered		

(Default: 0x0)

- Notes:**
- The prescaler (PSC Ch.1) output is used as the filter clock. Make sure the prescaler (PSC Ch.1) is turned on before using the chattering filter. Do not enable the chattering filter when the prescaler (PSC Ch.1) is turned off, as undesired port input interrupts may be generated.
 - The chattering filter stops operating in SLEEP mode, as no clock is supplied. In order to cancel SLEEP mode using a port input interrupt, the chattering filter will be automatically bypassed (Not filtered) in SLEEP mode until the CPU exits SLEEP mode even if the chattering filter is set to on.
 - Setting the GPIO_FPTnn_CHAT register while the interrupt is enabled may generate an undesired port input interrupt. Therefore, the port input interrupt must be disabled before setting the GPIO_FPTnn_CHAT register. Furthermore, be sure to clear the port input interrupt flag before enabling the interrupt again after setting the GPIO_FPTnn_CHAT register. In this case, clear the interrupt flag after the wait time shown below has elapsed from the GPIO_FPTnn_CHAT register setting.

$$\text{Wait time } [\mu\text{s}] = \text{Filter sampling time } [\mu\text{s}] \times 4$$

Example: When the filter sampling time is $64/f_{\text{PCLK2}}$ and $f_{\text{PCLK2}} = 32 \text{ MHz}$

$$\text{Wait time} = 64 \times 4 / 32 = 8 \text{ } [\mu\text{s}]$$

The port input interrupt flag should be cleared after waiting 8 μs or more when the GPIO_FPTnn_CHAT register is set to 0x7.

24.6 Input Port Noise Filters

The S1C33L26 provides noise filters to remove noise on the signals input from the ports shown below.

USI: USI_DI, USI_CS, USI_CK
 USIL: USIL_DI, USIL_CS, USIL_CK
 FSIO: SIN0, SIN1, SCLK0, SCLK1, #SRDY0, #SRDY1
 REMC: REMC_I
 T16A5: T16A_EXCL_0, T16A_EXCL_1, T16A_ATMA_0, T16A_ATMA_1, T16A_ATMB_0, T16A_ATMB_1
 ADC10: #ADTRIG
 GPIO: FPT0–FPTF interrupt ports (See note below.)

When using these noise filters, set ANFEN/GPIO_FILTER register to 1. When ANFEN is set to 0 (default), the input signals bypass the noise filters.

- Notes:**
- These noise filters cannot be enabled individually.
 - The noise filters are not effective if these ports are used as general-purpose input port. However, the noise filters for the general-purpose input ports that are selected as FPT interrupt ports (FPT0 to FPTF) are effective.
 - The GPIO_FILTER register is write-protected. Before the register can be rewritten, the write protection must be removed by writing data 0x96 to PPROT[7:0]/GPIO_PROTECT register.

24.7 Bus Drive Control

The external data bus (D[15:0]) pins and the external address bus (A[25:0]) pins can be forcibly set to low level using LDRVDB/GPIO_BUS_DRV register and LDRVAD/GPIO_BUS_DRV register, respectively. This function is useful when turning off the power of the external device connected to the bus.

When the control bit is set to 1, the corresponding bus signals go 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 a general-purpose I/O port (Pxy).
 - If the above 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.

24.8 Control Register Details

Table 24.8.1 List of GPIO and Port MUX Registers

Address	Register name	Function
0x300300	GPIO_P0_DAT	P0 Port Data Register P0 port input/output data
0x300301	GPIO_P0_IOC	P0 Port I/O Control Register Control P0 port input/output direction
0x300302	GPIO_P1_DAT	P1 Port Data Register P1 port input/output data
0x300303	GPIO_P1_IOC	P1 Port I/O Control Register Control P1 port input/output direction
0x300304	GPIO_P2_DAT	P2 Port Data Register P2 port input/output data
0x300305	GPIO_P2_IOC	P2 Port I/O Control Register Control P2 port input/output direction
0x300306	GPIO_P3_DAT	P3 Port Data Register P3 port input/output data
0x300307	GPIO_P3_IOC	P3 Port I/O Control Register Control P3 port input/output direction
0x300308	GPIO_P4_DAT	P4 Port Data Register P4 port input/output data
0x300309	GPIO_P4_IOC	P4 Port I/O Control Register Control P4 port input/output direction
0x30030a	GPIO_P5_DAT	P5 Port Data Register P5 port input/output data
0x30030b	GPIO_P5_IOC	P5 Port I/O Control Register Control P5 port input/output direction
0x30030c	GPIO_P6_DAT	P6 Port Data Register P6 port input/output data
0x30030d	GPIO_P6_IOC	P6 Port I/O Control Register Control P6 port input/output direction
0x30030e	GPIO_P7_DAT	P7 Port Data Register P7 port input data
0x300310	GPIO_P8_DAT	P8 Port Data Register P8 port input/output data
0x300311	GPIO_P8_IOC	P8 Port I/O Control Register Control P8 port input/output direction
0x300312	GPIO_P9_DAT	P9 Port Data Register P9 port input/output data
0x300313	GPIO_P9_IOC	P9 Port I/O Control Register Control P9 port input/output direction
0x300314	GPIO_PA_DAT	PA Port Data Register PA port input/output data

Address	Register name		Function
0x300315	GPIO_PA_IOC	PA Port I/O Control Register	Control PA port input/output direction
0x300316	GPIO_PB_DAT	PB Port Data Register	PB port input/output data
0x300317	GPIO_PB_IOC	PB Port I/O Control Register	Control PB port input/output direction
0x300318	GPIO_PC_DAT	PC Port Data Register	PC port input/output data
0x300319	GPIO_PC_IOC	PC Port I/O Control Register	Control PC port input/output direction
0x300320	GPIO_BUS_DRV	Bus Drive Control Register	Set external data and address bus signals to low
0x300321	GPIO_P0_PUP	P0 Port Pull-up Control Register	Enable/disable P0 port pull-up resistors
0x300322	GPIO_P1_PUP	P1 Port Pull-up Control Register	Enable/disable P1 port pull-up resistors
0x300323	GPIO_P2_PUP	P2 Port Pull-up Control Register	Enable/disable P2 port pull-up resistors
0x300324	GPIO_P3_PUP	P3 Port Pull-up Control Register	Enable/disable P3 port pull-up resistors
0x300325	GPIO_P4_PUP	P4 Port Pull-up Control Register	Enable/disable P4 port pull-up resistors
0x300326	GPIO_P5_PUP	P5 Port Pull-up Control Register	Enable/disable P5 port pull-up resistors
0x300327	GPIO_P6_PUP	P6 Port Pull-up Control Register	Enable/disable P6 port pull-up resistors
0x300328	GPIO_P7_PUP	P7 Port Pull-up Control Register	Enable/disable P7 port pull-up resistors
0x300329	GPIO_P8_PUP	P8 Port Pull-up Control Register	Enable/disable P8 port pull-up resistors
0x30032a	GPIO_P9_PUP	P9 Port Pull-up Control Register	Enable/disable P9 port pull-up resistors
0x30032b	GPIO_PA_PUP	PA Port Pull-up Control Register	Enable/disable PA port pull-up resistors
0x30032c	GPIO_PB_PUP	PB Port Pull-up Control Register	Enable/disable PB port pull-up resistors
0x300330	GPIO_FPT03_SEL	FPT0–3 Interrupt Port Select Register	Select ports used for FPT0–3 interrupts
0x300331	GPIO_FPT47_SEL	FPT4–7 Interrupt Port Select Register	Select ports used for FPT4–7 interrupts
0x300332	GPIO_FPT8B_SEL	FPT8–B Interrupt Port Select Register	Select ports used for FPT8–B interrupts
0x300333	GPIO_FPTCF_SEL	FPTC–F Interrupt Port Select Register	Select ports used for FPTC–F interrupts
0x300334	GPIO_FPT03_POL	FPT0–3 Interrupt Polarity Select Register	Select input signal polarity for FPT0–3 interrupts
0x300335	GPIO_FPT47_POL	FPT4–7 Interrupt Polarity Select Register	Select input signal polarity for FPT4–7 interrupts
0x300336	GPIO_FPT8B_POL	FPT8–B Interrupt Polarity Select Register	Select input signal polarity for FPT8–B interrupts
0x300337	GPIO_FPTCF_POL	FPTC–F Interrupt Polarity Select Register	Select input signal polarity for FPTC–F interrupts
0x300338	GPIO_FPT03_MOD	FPT0–3 Interrupt Mode Select Register	Select edge/level mode for FPT0–3 interrupts
0x300339	GPIO_FPT47_MOD	FPT4–7 Interrupt Mode Select Register	Select edge/level mode for FPT4–7 interrupts
0x30033a	GPIO_FPT8B_MOD	FPT8–B Interrupt Mode Select Register	Select edge/level mode for FPT8–B interrupts
0x30033b	GPIO_FPTCF_MOD	FPTC–F Interrupt Mode Select Register	Select edge/level mode for FPTC–F interrupts
0x30033c	GPIO_FPT03_MSK	FPT0–3 Interrupt Mask Register	Enable/disable FPT0–3 interrupts
0x30033d	GPIO_FPT47_MSK	FPT4–7 Interrupt Mask Register	Enable/disable FPT4–7 interrupts
0x30033e	GPIO_FPT8B_MSK	FPT8–B Interrupt Mask Register	Enable/disable FPT8–B interrupts
0x30033f	GPIO_FPTCF_MSK	FPTC–F Interrupt Mask Register	Enable/disable FPTC–F interrupts
0x300340	GPIO_FPT03_FLG	FPT0–3 Interrupt Flag Register	Indicate FPT0–3 interrupt cause status
0x300341	GPIO_FPT47_FLG	FPT4–7 Interrupt Flag Register	Indicate FPT4–7 interrupt cause status
0x300342	GPIO_FPT8B_FLG	FPT8–B Interrupt Flag Register	Indicate FPT8–B interrupt cause status
0x300343	GPIO_FPTCF_FLG	FPTC–F Interrupt Flag Register	Indicate FPTC–F interrupt cause status
0x300344	GPIO_FPT01_CHAT	FPT0–1 Interrupt Chattering Filter Control Register	Control FPT0–1 chattering filter
0x300345	GPIO_FPT23_CHAT	FPT2–3 Interrupt Chattering Filter Control Register	Control FPT2–3 chattering filter
0x300346	GPIO_FPT45_CHAT	FPT4–5 Interrupt Chattering Filter Control Register	Control FPT4–5 chattering filter
0x300347	GPIO_FPT67_CHAT	FPT6–7 Interrupt Chattering Filter Control Register	Control FPT6–7 chattering filter
0x300348	GPIO_FPT89_CHAT	FPT8–9 Interrupt Chattering Filter Control Register	Control FPT8–9 chattering filter
0x300349	GPIO_FPTAB_CHAT	FPTA–B Interrupt Chattering Filter Control Register	Control FPTA–B chattering filter
0x30034a	GPIO_FPTCD_CHAT	FPTC–D Interrupt Chattering Filter Control Register	Control FPTC–D chattering filter
0x30034b	GPIO_FPTFE_CHAT	FPTF–F Interrupt Chattering Filter Control Register	Control FPTF–F chattering filter
0x30034c	GPIO_DMA	Port DMA Trigger Source Select Register	Select port DMA trigger source
0x300800	PMUX_P0_03	P0[3:0] Port Function Select Register	Select P0[3:0] port functions
0x300801	PMUX_P0_47	P0[7:4] Port Function Select Register	Select P0[7:4] port functions
0x300802	PMUX_P1_03	P1[3:0] Port Function Select Register	Select P1[3:0] port functions
0x300803	PMUX_P1_47	P1[7:4] Port Function Select Register	Select P1[7:4] port functions
0x300804	PMUX_P2_01	P2[1:0] Port Function Select Register	Select P2[1:0] port functions
0x300806	PMUX_P3_03	P3[3:0] Port Function Select Register	Select P3[3:0] port functions
0x300807	PMUX_P3_46	P3[6:4] Port Function Select Register	Select P3[6:4] port functions
0x300808	PMUX_P4_02	P4[2:0] Port Function Select Register	Select P4[2:0] port functions
0x30080a	PMUX_P5_03	P5[3:0] Port Function Select Register	Select P5[3:0] port functions
0x30080b	PMUX_P5_46	P5[6:4] Port Function Select Register	Select P5[6:4] port functions
0x30080c	PMUX_P6_0	P60 Port Function Select Register	Select P60 port functions
0x30080e	PMUX_P7_03	P7[3:0] Port Function Select Register	Select P7[3:0] port functions
0x30080f	PMUX_P7_45	P7[5:4] Port Function Select Register	Select P7[5:4] port functions
0x300810	PMUX_P8_03	P8[3:0] Port Function Select Register	Select P8[3:0] port functions
0x300812	PMUX_P9_03	P9[3:0] Port Function Select Register	Select P9[3:0] port functions
0x300813	PMUX_P9_47	P9[7:4] Port Function Select Register	Select P9[7:4] port functions
0x300814	PMUX_PA_03	PA[3:0] Port Function Select Register	Select PA[3:0] port functions
0x300815	PMUX_PA_46	PA[6:4] Port Function Select Register	Select PA[6:4] port functions
0x300816	PMUX_PB_03	PB[3:0] Port Function Select Register	Select PB[3:0] port functions
0x300817	PMUX_PB_47	PB[7:4] Port Function Select Register	Select PB[7:4] port functions
0x300818	PMUX_PC_03	PC[3:0] Port Function Select Register	Select PC[3:0] port functions

Address	Register name		Function
0x300819	PMUX_PC_47	PC[7:4] Port Function Select Register	Select PC[7:4] port functions
0x30083e	GPIO_FILTER	Port Noise Filter Control Register	Enable/disable port input noise filter
0x30083f	GPIO_PROTECT	GPIO/PMUX Write Protect Register	Enable/disable write protection for PMUX, GPIO_FILTER, GPIO_BUS_DRV, and GPIO_Px_PUP registers

The I/O port registers are described in detail below. These are 8-bit registers.

- Notes:**
- When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.
 - The GPIO_BUS_DRV register, GPIO_Px_PUP registers, GPIO_FILTER register, and PMUX_Px_yy registers are write-protected. Before these registers can be rewritten, the write protection must be removed by writing data 0x96 to PPROT[7:0]GPIO_PROTECT register. Note that since unnecessary rewrites to these registers could lead to erratic system operation, PPROT[7:0] should be set to other than 0x96 unless the registers above must be rewritten.

Px Port Data Registers (GPIO_Px_DAT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Px Port Data Register (GPIO_Px_DAT)	0x300300	D7-0	Px[7:0]D	Px[7:0] I/O port data	1	1 (High)	0	0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
	0x300302										
	0x300318 (8 bits)										

Note: The PxyD bits for unavailable pins are read only bits from which 0 is always read out.

D[7:0] Px[7:0]D: Px[7:0] I/O Port Data Bits

These bits 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

PxyD corresponds directly to the Pxy pin.

The pin voltage level can be read out (even if the port is set to output mode (IOCx_{xy}/GPIO_Px_IOC register = 1)). The value read out will be 1 when the pin voltage is high and 0 when low.

When the port is set to output mode (IOCx_{xy}/GPIO_Px_IOC register = 1), the data written will be output unchanged from the port pins. The port pin will be high when the data bit is set to 1 and low when set to 0. Port data can also be written in input mode (IOCx_{xy} = 0) (the pin status is unaffected).

Px Port I/O Control Registers (GPIO_Px_IOC)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
Px Port I/O Control Register (GPIO_Px_IOC)	0x300301	D7-0	IOCx[7:0]	Px[7:0] I/O control	1	Output	0	Input	0x0	R/W	
	0x300303										
	0x300319 (8 bits)										

Note: The IOCx_{xy} bits for unavailable pins are read only bits from which 0 is always read out.

D[7:0] IOCx[7:0]: Px[7:0] I/O Control Bits

Sets the port to input or output mode.

1 (R/W): Output mode

0 (R/W): Input mode (default)

IOCx_{xy} is the I/O direction control bit that corresponds directly to Pxy port. Setting to 1 enables output and the data set in PxyD is output from the port pin. Output is disabled when IOCx_{xy} is set to 0, and the port pin is set into high-impedance status for inputting an external signal. The peripheral module determines whether output is enabled or disabled when the port is used for a peripheral module function.

Bus Drive Control Register (GPIO_BUS_DRV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Bus Drive Control Register (GPIO_BUS_DRV)	0x300320 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	LDRVDB	D[15:0] low drive	1 Low drive	0 Normal output	0 R/W	Write-protected
		D0	LDRVAD	A[25:0] low drive			0 R/W	

D[7:2] Reserved

D1 LDRVDB: D[15:0] Low Drive 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 D[15:0] signals are forcibly driven low. When it is set to 0, the signals are controlled by the SRAMC/SDRAMC normally.

D0 LDRVAD: A[25:0] Low Drive 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 A[25:0] signals are forcibly driven low. When it is set to 0, the signals are controlled by the SRAMC/SDRAMC normally.

Px Port Pull-up Control Registers (GPIO_Px_PUP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Px Port Pull-up Control Register (GPIO_Px_PUP)	0x300321	D7-0	PUPx[7:0]	Px[7:0] port pull-up enable	1 Enable	0 Disable	*	R/W	Write-protected
	0x30032c (8 bits)								

Note: The PUP_{xy} bits for unavailable pins are read only bits from which 0 is always read out.

D[7:0] PUPx[7:0]: Px[7:0] Port Pull-Up Enable Bits

Enables or disables the pull-up resistor for the I/O port pin.

1 (R/W): Enabled

0 (R/W): Disabled

PUP_{xy} is the pull-up enable bit that corresponds directly to P_{xy} port. Setting to 1 enables the pull-up resistor so that the port pin will be pulled up when the port is set to input mode.

When the port is in output mode, the port pin is not pulled up even if PUP_{xy} is set to 1.

The pull-up register is disabled when PUP_{xy} is set to 0.

This control is also effective when the port is used for a peripheral module function.

The table below shows the initial pull-up settings.

Table 24.8.2 Initial Pull-Up Status

Port	Px0	Px1	Px2	Px3	Px4	Px5	Px6	Px7
P0	–	–	–	–	–	–	–	–
P1	–	–	–	–	–	–	–	–
P2	–	–	–	–	–	–	–	–
P3	–	–	–	–	–	Enabled	–	–
P4	–	–	–	–	–	–	–	–
P5	–	–	–	Enabled	–	–	–	–
P6	Enabled	–	–	–	–	–	–	–
P7	–	–	–	–	–	Enabled	–	–
P8	–	–	–	–	–	–	–	–
P9	–	–	–	–	–	–	–	–
PA	–	Enabled	–	–	–	–	–	–
PB	–	–	–	–	–	–	–	–
PC	Pull-up resistors not included							

–: Disabled

FPT0–3 Interrupt Port Select Register (GPIO_FPT03_SEL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT0–3 Interrupt Port Select Register (GPIO_FPT03_ SEL)	0x300330 (8 bits)	D7–6	SPT3[1:0]	FPT3 interrupt input port select	SPT3[1:0]	Port	0x0	R/W	
					0x3	P33			
					0x2	P13			
					0x1	P53			
		0x0	P03						
		D5–4	SPT2[1:0]	FPT2 interrupt input port select	SPT2[1:0]	Port	0x0	R/W	
					0x3	P32			
					0x2	P12			
					0x1	P52			
		0x0	P02						
		D3–2	SPT1[1:0]	FPT1 interrupt input port select	SPT1[1:0]	Port	0x0	R/W	
					0x3	P31			
					0x2	P11			
					0x1	P51			
		0x0	P01						
		D1–0	SPT0[1:0]	FPT0 interrupt input port select	SPT0[1:0]	Port	0x0	R/W	
0x3	P30								
0x2	P10								
0x1	P50								
0x0	P00								

D[7:6] SPT3[1:0]: FPT3 Interrupt input Port Select Bits

Selects an FPT3 port used for generating port interrupt 0.

D[5:4] SPT2[1:0]: FPT2 Interrupt input Port Select Bits

Selects an FPT2 port used for generating port interrupt 0.

D[3:2] SPT1[1:0]: FPT1 Interrupt input Port Select Bits

Selects an FPT1 port used for generating port interrupt 0.

D[1:0] SPT0[1:0]: FPT0 Interrupt input Port Select Bits

Selects an FPT0 port used for generating port interrupt 0.

Table 24.8.3 Selecting Ports Used For Port Interrupt 0

SPT n [1:0] setting	FPT0 (SPT0[1:0])	FPT1 (SPT1[1:0])	FPT2 (SPT2[1:0])	FPT3 (SPT3[1:0])
0x3	P30	P31	P32	P33
0x2	P10	P11	P12	P13
0x1	P50	P51	P52	P53
0x0 (default)	P00	P01	P02	P03

FPT4–7 Interrupt Port Select Register (GPIO_FPT47_SEL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT4–7 Interrupt Port Select Register (GPIO_FPT47_ SEL)	0x300331 (8 bits)	D7–6	SPT7[1:0]	FPT7 interrupt input port select	SPT7[1:0]	Port	0x0	R/W	
					0x3	PA3			
					0x2	P17			
					0x1	PC7			
		0x0	P07						
		D5–4	SPT6[1:0]	FPT6 interrupt input port select	SPT6[1:0]	Port	0x0	R/W	
					0x3	PA2			
					0x2	P16			
					0x1	PC6			
		0x0	P06						
		D3–2	SPT5[1:0]	FPT5 interrupt input port select	SPT5[1:0]	Port	0x0	R/W	
					0x3	PA1			
					0x2	P15			
					0x1	PC5			
		0x0	P05						
		D1–0	SPT4[1:0]	FPT4 interrupt input port select	SPT4[1:0]	Port	0x0	R/W	
0x3	PA0								
0x2	P14								
0x1	PC4								
0x0	P04								

D[7:6] SPT7[1:0]: FPT7 Interrupt input Port Select Bits

Selects an FPT7 port used for generating port interrupt 1.

D[5:4] SPT6[1:0]: FPT6 Interrupt input Port Select Bits

Selects an FPT6 port used for generating port interrupt 1.

D[3:2] SPT5[1:0]: FPT5 Interrupt input Port Select Bits
 Selects an FPT5 port used for generating port interrupt 1.

D[1:0] SPT4[1:0]: FPT4 Interrupt input Port Select Bits
 Selects an FPT4 port used for generating port interrupt 1.

Table 24.8.4 Selecting Ports Used For Port Interrupt 1

SPTn[1:0] setting	FPT4 (SPT4[1:0])	FPT5 (SPT5[1:0])	FPT6 (SPT6[1:0])	FPT7 (SPT7[1:0])
0x3	PA0	PA1	PA2	PA3
0x2	P14	P15	P16	P17
0x1	PC4	PC5	PC6	PC7
0x0 (default)	P04	P05	P06	P07

FPT8–B Interrupt Port Select Register (GPIO_FPT8B_SEL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT8–B Interrupt Port Select Register (GPIO_FPT8B_ SEL)	0x300332 (8 bits)	D7–6	SPTB[1:0]	FPTB interrupt input port select	SPTB[1:0]	Port	0x0	R/W	
					0x3	P93			
					0x2	P83			
					0x1	PB3			
		0x0	P71						
		D5–4	SPTA[1:0]	FPTA interrupt input port select	SPTA[1:0]	Port	0x0	R/W	
					0x3	P92			
					0x2	P82			
					0x1	PB2			
		0x0	P42						
		D3–2	SPT9[1:0]	FPT9 interrupt input port select	SPT9[1:0]	Port	0x0	R/W	
					0x3	P91			
					0x2	P81			
					0x1	PB1			
		0x0	P41						
		D1–0	SPT8[1:0]	FPT8 interrupt input port select	SPT8[1:0]	Port	0x0	R/W	
0x3	P90								
0x2	P80								
0x1	PB0								
0x0	P40								

D[7:6] SPTB[1:0]: FPTB Interrupt input Port Select Bits
 Selects an FPTB port used for generating port interrupt 2.

D[5:4] SPTA[1:0]: FPTA Interrupt input Port Select Bits
 Selects an FPTA port used for generating port interrupt 2.

D[3:2] SPT9[1:0]: FPT9 Interrupt input Port Select Bits
 Selects an FPT9 port used for generating port interrupt 2.

D[1:0] SPT8[1:0]: FPT8 Interrupt input Port Select Bits
 Selects an FPT8 port used for generating port interrupt 2.

Table 24.8.5 Selecting Ports Used For Port Interrupt 2

SPTn[1:0] setting	FPT8 (SPT8[1:0])	FPT9 (SPT9[1:0])	FPTA (SPTA[1:0])	FPTB (SPTB[1:0])
0x3	P90	P91	P92	P93
0x2	P80	P81	P82	P83
0x1	PB0	PB1	PB2	PB3
0x0 (default)	P40	P41	P42	P71

FPTC–F Interrupt Port Select Register (GPIO_FPTCF_SEL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPTC–F Interrupt Port Select Register (GPIO_FPTCF_ SEL)	0x300333 (8 bits)	D7–6	SPTF[1:0]	FPTF interrupt input port select	SPTF[1:0]	Port	0x0	R/W	
					0x3	P97			
					0x2	P60			
					0x1	PB7			
		0x0	P75						
		D5–4	SPTF[1:0]	FPTE interrupt input port select	SPTF[1:0]	Port	0x0	R/W	
					0x3	P96			
					0x2	PA6			
					0x1	PB6			
		0x0	P74						
		D3–2	SPTD[1:0]	FPTD interrupt input port select	SPTD[1:0]	Port	0x0	R/W	
					0x3	P95			
					0x2	PA5			
					0x1	PB5			
		0x0	P73						
		D1–0	SPTC[1:0]	FPTC interrupt input port select	SPTC[1:0]	Port	0x0	R/W	
0x3	P94								
0x2	PA4								
0x1	PB4								
0x0	P72								

D[7:6] **SPTF[1:0]: FPTF Interrupt input Port Select Bits**

Selects an FPTF port used for generating port interrupt 3.

D[5:4] **SPTF[1:0]: FPTE Interrupt input Port Select Bits**

Selects an FPTE port used for generating port interrupt 3.

D[3:2] **SPTD[1:0]: FPTD Interrupt input Port Select Bits**

Selects an FPTD port used for generating port interrupt 3.

D[1:0] **SPTC[1:0]: FPTC Interrupt input Port Select Bits**

Selects an FPTC port used for generating port interrupt 3.

Table 24.8.6 Selecting Ports Used For Port Interrupt 3

SPTn[1:0] setting	FPTC (SPTC[1:0])	FPTD (SPTD[1:0])	FPTE (SPTF[1:0])	FPTF (SPTF[1:0])
0x3	P94	P95	P96	P97
0x2	PA4	PA5	PA6	P60
0x1	PB4	PB5	PB6	PB7
0x0 (default)	P72	P73	P74	P75

FPT0–3 Interrupt Polarity Select Register (GPIO_FPT03_POL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT0–3 Interrupt Polarity Select Register (GPIO_FPT03_ POL)	0x300334 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SPPT3	FPT3 input polarity select	1 High / ↑	0 Low / ↓	1	R/W
		D2	SPPT2	FPT2 input polarity select	1 High / ↑	0 Low / ↓	1	R/W
		D1	SPPT1	FPT1 input polarity select	1 High / ↑	0 Low / ↓	1	R/W
		D0	SPPT0	FPT0 input polarity select	1 High / ↑	0 Low / ↓	1	R/W

D[7:4] **Reserved**

D[3:0] **SPPT[3:0]: FPT[3:0] Input Polarity Select Bits**

Selects the interrupt trigger level or edge for the ports used for port interrupt 0 (FPT0–FPT3).

1 (R/W): High level/Rising edge (default)

0 (R/W): Low level/Falling edge

When SPPTn is set to 1 (default), high level (in level trigger mode) or rising edge (in edge trigger mode) is selected as the interrupt generating condition for FPTn port.

When SPPTn is set to 0, low level (in level trigger mode) or falling edge (in edge trigger mode) is selected.

FPT4–7 Interrupt Polarity Select Register (GPIO_FPT47_POL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT4–7 Interrupt Polarity Select Register (GPIO_FPT47_ POL)	0x300335 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SPPT7	FPT7 input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D2	SPPT6	FPT6 input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D1	SPPT5	FPT5 input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D0	SPPT4	FPT4 input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	

D[7:4] Reserved**D[3:0] SPPT[7:4]: FPT[7:4] Input Polarity Select Bits**

Selects the interrupt trigger level or edge for the ports used for port interrupt 1 (FPT4–FPT7).

1 (R/W): High level/Rising edge (default)

0 (R/W): Low level/Falling edge

See the descriptions of SPPT[3:0]/GPIO_FPT03_POL register.

FPT8–B Interrupt Polarity Select Register (GPIO_FPT8B_POL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT8–B Interrupt Polarity Select Register (GPIO_FPT8B_ POL)	0x300336 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SPPTB	FPTB input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D2	SPPTA	FPTA input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D1	SPPT9	FPT9 input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D0	SPPT8	FPT8 input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	

D[7:4] Reserved**D[3:0] SPPT[B:8]: FPT[B:8] Input Polarity Select Bits**

Selects the interrupt trigger level or edge for the ports used for port interrupt 2 (FPT8–FPTB).

1 (R/W): High level/Rising edge (default)

0 (R/W): Low level/Falling edge

See the descriptions of SPPT[3:0]/GPIO_FPT03_POL register.

FPTC–F Interrupt Polarity Select Register (GPIO_FPTCF_POL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPTC–F Interrupt Polarity Select Register (GPIO_FPTCF_ POL)	0x300337 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SPPTF	FPTF input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D2	SPPTE	FPTE input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D1	SPPTD	FPTD input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	
		D0	SPPTC	FPTC input polarity select	1 High / ↑ 0 Low / ↓	1	R/W	

D[7:4] Reserved**D[3:0] SPPT[F:C]: FPT[F:C] Input Polarity Select Bits**

Selects the interrupt trigger level or edge for the ports used for port interrupt 3 (FPTC–FPTF).

1 (R/W): High level/Rising edge (default)

0 (R/W): Low level/Falling edge

See the descriptions of SPPT[3:0]/GPIO_FPT03_POL register.

FPT0–3 Interrupt Mode Select Register (GPIO_FPT03_MOD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT0–3 Interrupt Mode Select Register (GPIO_FPT03_ MOD)	0x300338 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SEPT3	FPT3 interrupt mode select	1 Edge 0 Level	1	R/W	
		D2	SEPT2	FPT2 interrupt mode select	1 Edge 0 Level	1	R/W	
		D1	SEPT1	FPT1 interrupt mode select	1 Edge 0 Level	1	R/W	
		D0	SEPT0	FPT0 interrupt mode select	1 Edge 0 Level	1	R/W	

D[7:4] Reserved

D[3:0] SEPT[3:0]: FPT[3:0] Interrupt Mode Select Bits

Selects trigger modes of the ports used for port interrupt 0 (FPT0–FPT3).

1 (R/W): Edge trigger mode (default)

0 (R/W): Level trigger mode

When $SEPT_n$ is set to 1 (default), the corresponding port is set to edge trigger mode. In edge trigger mode, the interrupt flag is set at the active edge of the input signal and it retains 1 until reset via software.

When $SEPT_n$ is set to 0, the corresponding port is set to level trigger mode. In level trigger mode, the interrupt flag is set when the input signal goes the active level and it retains 1 until reset via software. SLEEP mode can be canceled by causing a port input interrupt regardless of how the GPIO interrupt mode (edge trigger/level trigger) is set.

FPT4–7 Interrupt Mode Select Register (GPIO_FPT47_MOD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT4–7 Interrupt Mode Select Register (GPIO_FPT47_ MOD)	0x300339 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SEPT7	FPT7 interrupt mode select	1 Edge	0 Level	1 R/W	
		D2	SEPT6	FPT6 interrupt mode select	1 Edge	0 Level	1 R/W	
		D1	SEPT5	FPT5 interrupt mode select	1 Edge	0 Level	1 R/W	
		D0	SEPT4	FPT4 interrupt mode select	1 Edge	0 Level	1 R/W	

D[7:4] Reserved**D[3:0] SEPT[7:4]: FPT[7:4] Interrupt Mode Select Bits**

Selects trigger modes of the ports used for port interrupt 1 (FPT4–FPT7).

1 (R/W): Edge trigger mode (default)

0 (R/W): Level trigger mode

See the descriptions of SEPT[3:0]/GPIO_FPT03_MOD register.

FPT8–B Interrupt Mode Select Register (GPIO_FPT8B_MOD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT8–B Interrupt Mode Select Register (GPIO_FPT8B_ MOD)	0x30033a (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SEPTB	FPTB interrupt mode select	1 Edge	0 Level	1 R/W	
		D2	SEPTA	FPTA interrupt mode select	1 Edge	0 Level	1 R/W	
		D1	SEPT9	FPT9 interrupt mode select	1 Edge	0 Level	1 R/W	
		D0	SEPT8	FPT8 interrupt mode select	1 Edge	0 Level	1 R/W	

D[7:4] Reserved**D[3:0] SEPT[B:8]: FPT[B:8] Interrupt Mode Select Bits**

Selects trigger modes of the ports used for port interrupt 2 (FPT8–FPTB).

1 (R/W): Edge trigger mode (default)

0 (R/W): Level trigger mode

See the descriptions of SEPT[3:0]/GPIO_FPT03_MOD register.

FPTC–F Interrupt Mode Select Register (GPIO_FPTCF_MOD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPTC–F Interrupt Mode Select Register (GPIO_FPTCF_ MOD)	0x30033b (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SEPTF	FPTF interrupt mode select	1 Edge	0 Level	1 R/W	
		D2	SEPTE	FPT E interrupt mode select	1 Edge	0 Level	1 R/W	
		D1	SEPTD	FPTD interrupt mode select	1 Edge	0 Level	1 R/W	
		D0	SEPTC	FPTC interrupt mode select	1 Edge	0 Level	1 R/W	

D[7:4] Reserved

D[3:0] SEPT[F:C]: FPT[F:C] Interrupt Mode Select Bits

Selects trigger modes of the ports used for port interrupt 3 (FPTC–FPTF).

1 (R/W): Edge trigger mode (default)

0 (R/W): Level trigger mode

See the descriptions of SEPT[3:0]/GPIO_FPT03_MOD register.

FPT0–3 Interrupt Mask Register (GPIO_FPT03_MSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT0–3 Interrupt Mask Register (GPIO_FPT03_ MSK)	0x30033c (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SIET3	FPT3 interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	SIET2	FPT2 interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SIET1	FPT1 interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	SIET0	FPT0 interrupt enable	1 Enable 0 Disable	0	R/W	

D[7:4] Reserved**D[3:0] SIET[3:0]: FPT[3:0] Interrupt Enable Bits**

Enables or disables the ports to generate port interrupt 0 (FPT0–FPT3).

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

FPT4–7 Interrupt Mask Register (GPIO_FPT47_MSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT4–7 Interrupt Mask Register (GPIO_FPT47_ MSK)	0x30033d (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SIET7	FPT7 interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	SIET6	FPT6 interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SIET5	FPT5 interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	SIET4	FPT4 interrupt enable	1 Enable 0 Disable	0	R/W	

D[7:4] Reserved**D[3:0] SIET[7:4]: FPT[7:4] Interrupt Enable Bits**

Enables or disables the ports to generate port interrupt 1 (FPT4–FPT7).

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

FPT8–B Interrupt Mask Register (GPIO_FPT8B_MSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPT8–B Interrupt Mask Register (GPIO_FPT8B_ MSK)	0x30033e (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SIETB	FPTB interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	SIETA	FPTA interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SIET9	FPT9 interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	SIET8	FPT8 interrupt enable	1 Enable 0 Disable	0	R/W	

D[7:4] Reserved**D[3:0] SIET[B:8]: FPT[B:8] Interrupt Enable Bits**

Enables or disables the ports to generate port interrupt 2 (FPT8–FPTB).

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

FPTC–F Interrupt Mask Register (GPIO_FPTCF_MSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPTC–F Interrupt Mask Register (GPIO_FPTCF_ MSK)	0x30033f (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SIETF	FPTF interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	SIETE	FPTE interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SIETD	FPTD interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	SIETC	FPTC interrupt enable	1 Enable 0 Disable	0	R/W	

D[7:4] Reserved

D[3:0] SIET[F:C]: FPT[F:C] Interrupt Enable Bits

Enables or disables the ports to generate port interrupt 3 (FPTC–FPTF).

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

FPT0–3 Interrupt Flag Register (GPIO_FPT03_FLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT0–3 Interrupt Flag Register (GPIO_FPT03_ FLG)	0x300340 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3	SFGP3	FPT3 interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	X	R/W	Reset by writing 1.
		D2	SFGP2	FPT2 interrupt flag			X	R/W	
		D1	SFGP1	FPT1 interrupt flag			X	R/W	
		D0	SFGP0	FPT0 interrupt flag			X	R/W	

D[7:4] Reserved**D[3:0] SFGP[3:0]: FPT[3:0] Interrupt Flag Bits**

These are interrupt flags indicating the port input interrupt 0 (FPT0–FPT3) occurrence status. (Default: undefined)

1 (R): Interrupt cause occurred

0 (R): No interrupt cause occurred

1 (W): Reset flag

0 (W): Ignored

SFGP_n is the interrupt flag corresponding to the individual ports for port input interrupts and is set to 1 at the specified edge (rising or falling edge) or level (high or low) of the input signal. When the corresponding SIET_n bit has been set to 1, a port interrupt request signal is also output to the ITC at the same time. An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied. SFGP_n is reset by writing 1.

FPT4–7 Interrupt Flag Register (GPIO_FPT47_FLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT4–7 Interrupt Flag Register (GPIO_FPT47_ FLG)	0x300341 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3	SFGP7	FPT7 interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	X	R/W	Reset by writing 1.
		D2	SFGP6	FPT6 interrupt flag			X	R/W	
		D1	SFGP5	FPT5 interrupt flag			X	R/W	
		D0	SFGP4	FPT4 interrupt flag			X	R/W	

D[7:4] Reserved**D[3:0] SFGP[7:4]: FPT[7:4] Interrupt Flag Bits**

These are interrupt flags indicating the port input interrupt 1 (FPT4–FPT7) occurrence status. (Default: undefined)

1 (R): Interrupt cause occurred

0 (R): No interrupt cause occurred

1 (W): Reset flag

0 (W): Ignored

See the descriptions of SFGP[3:0]/GPIO_FPT03_FLG register.

FPT8–B Interrupt Flag Register (GPIO_FPT8B_FLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT8–B Interrupt Flag Register (GPIO_FPT8B_ FLG)	0x300342 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3	SFGPB	FPTB interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	X	R/W	Reset by writing 1.
		D2	SFGPA	FPTA interrupt flag			X	R/W	
		D1	SFGP9	FPT9 interrupt flag			X	R/W	
		D0	SFGP8	FPT8 interrupt flag			X	R/W	

D[7:4] Reserved

D[3:0] SFGP[B:8]: FPT[B:8] Interrupt Flag Bits

These are interrupt flags indicating the port input interrupt 2 (FPT8–FPTB) occurrence status. (Default: undefined)

- 1 (R): Interrupt cause occurred
- 0 (R): No interrupt cause occurred
- 1 (W): Reset flag
- 0 (W): Ignored

See the descriptions of SFGP[3:0]/GPIO_FPT03_FLG register.

FPTC–F Interrupt Flag Register (GPIO_FPTCF_FLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPTC–F Interrupt Flag Register (GPIO_FPTCF_ FLG)	0x300343 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	SFGPF	FPTF interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	X R/W	Reset by writing 1.
		D2	SFGPE	FPTF interrupt flag			X R/W	
		D1	SFGPD	FPTD interrupt flag			X R/W	
		D0	SFGPC	FPTC interrupt flag			X R/W	

D[7:4] Reserved**D[3:0] SFGP[F:C]: FPT[F:C] Interrupt Flag Bits**

These are interrupt flags indicating the port input interrupt 3 (FPTC–FPTF) occurrence status. (Default: undefined)

- 1 (R): Interrupt cause occurred
- 0 (R): No interrupt cause occurred
- 1 (W): Reset flag
- 0 (W): Ignored

See the descriptions of SFGP[3:0]/GPIO_FPT03_FLG register.

FPT0–1 Interrupt Chattering Filter Control Register (GPIO_FPT01_CHAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT0–1 Interrupt Chattering Filter Control Register (GPIO_FPT01_ CHAT)	0x300344 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6–4	SCTP1[2:0]	FPT1 chattering filter time select	SCTP1[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
					0x0	None			
D3	–	reserved	–	–	–	0 when being read.			
D2–0	SCTP0[2:0]	FPT0 chattering filter time select	SCTP0[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
			0x3	4/fPCLK2					
			0x2	2/fPCLK2					
			0x1	1/fPCLK2					
			0x0	None					

D7 Reserved**D[6:4] SCTP1[2:0]: FPT1 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT1 port.

D3 Reserved**D[2:0] SCTP0[2:0]: FPT0 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT0 port.

The FPT interrupt input ports include a chattering filter circuit for key entry that can be disabled or enabled with a filter sampling time specified individually for each FPT port using SCTP_n[2:0].

Table 24.8.7 Chattering Filter Settings

SCTP n [2:0]	Filter sampling time	Invalid pulse (glitch) width that will be filtered	Valid pulse width that will be accepted
0x7	64/fPCLK2	< 64/fPCLK2 × 2	> 64/fPCLK2 × 4
0x6	32/fPCLK2	< 32/fPCLK2 × 2	> 32/fPCLK2 × 4
0x5	16/fPCLK2	< 16/fPCLK2 × 2	> 16/fPCLK2 × 4
0x4	8/fPCLK2	< 8/fPCLK2 × 2	> 8/fPCLK2 × 4
0x3	4/fPCLK2	< 4/fPCLK2 × 2	> 4/fPCLK2 × 4
0x2	2/fPCLK2	< 2/fPCLK2 × 2	> 2/fPCLK2 × 4
0x1	1/fPCLK2	< 1/fPCLK2 × 2	> 1/fPCLK2 × 4
0x0	Not filtered		

(Default: 0x0)

- Notes:**
- The prescaler (PSC Ch.1) output is used as the filter clock. Make sure the prescaler (PSC Ch.1) is turned on before using the chattering filter. Do not enable the chattering filter when the prescaler (PSC Ch.1) is turned off, as undesired port input interrupts may be generated.
 - The chattering filter stops operating in SLEEP mode, as no clock is supplied. In order to cancel SLEEP mode using a port input interrupt, the chattering filter will be automatically bypassed (Not filtered) in SLEEP mode until the CPU exits SLEEP mode even if the chattering filter is set to on.
 - Setting the GPIO_FPT nn _CHAT register while the interrupt is enabled may generate an undesired port input interrupt. Therefore, the port input interrupt must be disabled before setting the GPIO_FPT nn _CHAT register. Furthermore, be sure to clear the port input interrupt flag before enabling the interrupt again after setting the GPIO_FPT nn _CHAT register. In this case, clear the interrupt flag after the wait time shown below has elapsed from the GPIO_FPT nn _CHAT register setting.

$$\text{Wait time } [\mu\text{s}] = \text{Filter sampling time } [\mu\text{s}] \times 4$$

Example: When the filter sampling time is 64/fPCLK2 and fPCLK2 = 32 MHz

$$\text{Wait time} = 64 \times 4 / 32 = 8 \text{ } [\mu\text{s}]$$

The port input interrupt flag should be cleared after waiting 8 μs or more when the GPIO_FPT nn _CHAT register is set to 0x7.

FPT2–3 Interrupt Chattering Filter Control Register (GPIO_FPT23_CHAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT2–3 Interrupt Chattering Filter Control Register (GPIO_FPT23_ CHAT)	0x300345 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6–4	SCTP3[2:0]	FPT3 chattering filter time select	SCTP3[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
		0x0	None						
D3	–	reserved	–	–	–	0 when being read.			
D2–0	SCTP2[2:0]	FPT2 chattering filter time select	SCTP2[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
			0x3	4/fPCLK2					
			0x2	2/fPCLK2					
			0x1	1/fPCLK2					
0x0	None								

D7 **Reserved**

D[6:4] **SCTP3[2:0]: FPT3 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT3 port.

D3 Reserved**D[2:0] SCTP2[2:0]: FPT2 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT2 port.

See the descriptions of SCTP0[2:0]/GPIO_FPT01_CHAT register.

FPT4–5 Interrupt Chattering Filter Control Register (GPIO_FPT45_CHAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT4–5 Interrupt Chattering Filter Control Register (GPIO_FPT45_ CHAT)	0x300346 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6–4	SCTP5[2:0]	FPT5 chattering filter time select	SCTP5[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
		0x0	None						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2–0	SCTP4[2:0]	FPT4 chattering filter time select	SCTP4[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
0x5	16/fPCLK2								
0x4	8/fPCLK2								
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								

D7 Reserved**D[6:4] SCTP5[2:0]: FPT5 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT5 port.

D3 Reserved**D[2:0] SCTP4[2:0]: FPT4 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT4 port.

See the descriptions of SCTP0[2:0]/GPIO_FPT01_CHAT register.

FPT6–7 Interrupt Chattering Filter Control Register (GPIO_FPT67_CHAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT6–7 Interrupt Chattering Filter Control Register (GPIO_FPT67_ CHAT)	0x300347 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6–4	SCTP7[2:0]	FPT7 chattering filter time select	SCTP7[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
		0x0	None						
		D3	–	reserved	–	–	–	–	0 when being read.
		D2–0	SCTP6[2:0]	FPT6 chattering filter time select	SCTP6[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
0x5	16/fPCLK2								
0x4	8/fPCLK2								
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								

D7 Reserved**D[6:4] SCTP7[2:0]: FPT7 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT7 port.

D3 **Reserved**

D[2:0] **SCTP6[2:0]: FPT6 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT6 port.

See the descriptions of SCTP0[2:0]/GPIO_FPT01_CHAT register.

FPT8–9 Interrupt Chattering Filter Control Register (GPIO_FPT89_CHAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT8–9 Interrupt Chattering Filter Control Register (GPIO_FPT89_ CHAT)	0x300348 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6–4	SCTP9[2:0]	FPT9 chattering filter time select	SCTP9[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
		0x1	1/fPCLK2						
		0x0	None						
D3	–	reserved	–	–	–	–	0 when being read.		
D2–0	SCTP8[2:0]	FPT8 chattering filter time select	SCTP8[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
			0x3	4/fPCLK2					
			0x2	2/fPCLK2					
			0x1	1/fPCLK2					
0x0	None								

D7 **Reserved**

D[6:4] **SCTP9[2:0]: FPT9 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT9 port.

D3 **Reserved**

D[2:0] **SCTP8[2:0]: FPT8 Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPT8 port.

See the descriptions of SCTP0[2:0]/GPIO_FPT01_CHAT register.

FPTA–B Interrupt Chattering Filter Control Register (GPIO_FPTAB_CHAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPTA–B Interrupt Chattering Filter Control Register (GPIO_FPTAB_ CHAT)	0x300349 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6–4	SCTPB[2:0]	FPTB chattering filter time select	SCTPB[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
		0x1	1/fPCLK2						
		0x0	None						
D3	–	reserved	–	–	–	–	0 when being read.		
D2–0	SCTPA[2:0]	FPTA chattering filter time select	SCTPA[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
			0x3	4/fPCLK2					
			0x2	2/fPCLK2					
			0x1	1/fPCLK2					
0x0	None								

D7 **Reserved**

D[6:4] **SCTPB[2:0]: FPTB Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPTB port.

D3 Reserved

D[2:0] SCTPA[2:0]: FPTA Chattering Filter Time Select Bits

Configures the chattering filter circuit for the FPTA port.

See the descriptions of SCTP0[2:0]/GPIO_FPT01_CHAT register.

FPTC–D Interrupt Chattering Filter Control Register (GPIO_FPTCD_CHAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
FPTC–D Interrupt Chattering Filter Control Register (GPIO_FPTCD_ CHAT)	0x30034a (8 bits)	D7	–	reserved	–	–	–	–	0 when being read.		
		D6–4	SCTPD[2:0]	FPTD chattering filter time select	SCTPD[2:0]	Filter sampling time	0x0	R/W			
										0x7	64/fPCLK2
										0x6	32/fPCLK2
										0x5	16/fPCLK2
										0x4	8/fPCLK2
										0x3	4/fPCLK2
										0x2	2/fPCLK2
										0x1	1/fPCLK2
		0x0	None								
		D3	–	reserved	–	–	–	–	0 when being read.		
		D2–0	SCTPC[2:0]	FPTC chattering filter time select	SCTPC[2:0]	Filter sampling time	0x0	R/W			
0x7	64/fPCLK2										
0x6	32/fPCLK2										
0x5	16/fPCLK2										
0x4	8/fPCLK2										
0x3	4/fPCLK2										
0x2	2/fPCLK2										
0x1	1/fPCLK2										
0x0	None										

D7 Reserved

D[6:4] SCTPD[2:0]: FPTD Chattering Filter Time Select Bits

Configures the chattering filter circuit for the FPTD port.

D3 Reserved

D[2:0] SCTPC[2:0]: FPTC Chattering Filter Time Select Bits

Configures the chattering filter circuit for the FPTC port.

See the descriptions of SCTP0[2:0]/GPIO_FPT01_CHAT register.

FPTE–F Interrupt Chattering Filter Control Register (GPIO_FPTEF_CHAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
FPTE–F Interrupt Chattering Filter Control Register (GPIO_FPTEF_ CHAT)	0x30034b (8 bits)	D7	–	reserved	–	–	–	–	0 when being read.		
		D6–4	SCTPF[2:0]	FPTF chattering filter time select	SCTPF[2:0]	Filter sampling time	0x0	R/W			
										0x7	64/fPCLK2
										0x6	32/fPCLK2
										0x5	16/fPCLK2
										0x4	8/fPCLK2
										0x3	4/fPCLK2
										0x2	2/fPCLK2
										0x1	1/fPCLK2
		0x0	None								
		D3	–	reserved	–	–	–	–	0 when being read.		
		D2–0	SCTPE[2:0]	FPTE chattering filter time select	SCTPE[2:0]	Filter sampling time	0x0	R/W			
0x7	64/fPCLK2										
0x6	32/fPCLK2										
0x5	16/fPCLK2										
0x4	8/fPCLK2										
0x3	4/fPCLK2										
0x2	2/fPCLK2										
0x1	1/fPCLK2										
0x0	None										

D7 Reserved

D[6:4] SCTPF[2:0]: FPTF Chattering Filter Time Select Bits

Configures the chattering filter circuit for the FPTF port.

D3 **Reserved**

D[2:0] **SCTPE[2:0]: FPTE Chattering Filter Time Select Bits**

Configures the chattering filter circuit for the FPTE port.

See the descriptions of SCTP0[2:0]/GPIO_FPT01_CHAT register.

Port DMA Trigger Source Select Register (GPIO_DMA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Port DMA Trigger Source Select Register (GPIO_DMA)	0x30034c (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.	
		D3-0	SPTRG[3:0]	Port DMA trigger source select	SPTRG[3:0] Trigger source	0x0	R/W		
						0xf	FPTF		
						0xe	FPTE		
						0xd	FPTD		
						0xc	FPTC		
						0xb	FPTB		
						0xa	FPTA		
						0x9	FPT9		
						0x8	FPT8		
						0x7	FPT7		
						0x6	FPT6		
						0x5	FPT5		
						0x4	FPT4		
						0x3	FPT3		
						0x2	FPT2		
				0x1	FPT1				
				0x0	FPT0				

D[7:4] **Reserved**

D[3:0] **SPTRG[3:0]: Port DMA Trigger Source Select Bits**

Selects an FPT port as a DMA trigger source to invoke DMA transfer.

Table 24.8.8 DMA Trigger Source Selection

SPTRG[3:0]	Trigger source	SPTRG[3:0]	Trigger source
0xf	FPTF	0x7	FPT7
0xe	FPTE	0x6	FPT6
0xd	FPTD	0x5	FPT5
0xc	FPTC	0x4	FPT4
0xb	FPTB	0x3	FPT3
0xa	FPTA	0x2	FPT2
0x9	FPT9	0x1	FPT1
0x8	FPT8	0x0	FPT0

(Default: 0x0)

The interrupt signal of the selected FPT line, which is generated according to the interrupt mode and polarity settings regardless of the SIET n setting (even if the interrupt is disabled), is sent to the DMAC to trigger a DMA transfer. For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

Note: After setting GPIO_FPT nn _CHAT register, the port may send an undesired DMA trigger to the DMAC. Disable the port DMA request for at least the wait time shown below after the GPIO_FPT nn _CHAT register is set.

Wait time [μ s] = Filter sampling time [μ s] \times 4

Example: When the filter sampling time is $64/f_{PCLK2}$ and $f_{PCLK2} = 32$ MHz

Wait time = $64 \times 4 / 32 = 8$ [μ s]

The port DMA function can be used after waiting 8 μ s or more when the GPIO_FPT nn _CHAT register is set to 0x7.

P0[3:0] Port Function Select Register (PMUX_P0_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P0[3:0] Port Function Select Register (PMUX_P0_03)	0x300800 (8 bits)	D7–6	CFP03[1:0]	P03 port function select	CFP03[1:0]	Function	0x0	R/W	Write-protected
					0x3	REMC_I			
					0x2	#SRDY1			
					0x1	USI_CK			
		D5–4	CFP02[1:0]	P02 port function select	CFP02[1:0]	Function	0x0	R/W	
					0x3	REMC_O			
					0x2	SCLK1			
					0x1	USI_CS			
		D3–2	CFP01[1:0]	P01 port function select	CFP01[1:0]	Function	0x0	R/W	
					0x3	#NAND_RD			
					0x2	SOUT1			
					0x1	USI_DO			
D1–0	CFP00[1:0]	P00 port function select	CFP00[1:0]	Function	0x0	R/W			
			0x3	#NAND_WR					
			0x2	SIN1					
			0x1	USI_DI					
					0x0				

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

0x3 (R/W): REMC_I (REMC)
 0x2 (R/W): #SRDY1 (FSIO Ch.1)
 0x1 (R/W): USI_CK (USI)
 0x0 (R/W): P03 (GPIO) (default)

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

0x3 (R/W): REMC_O (REMC)
 0x2 (R/W): SCLK1 (FSIO Ch.1)
 0x1 (R/W): USI_CS (USI)
 0x0 (R/W): P02 (GPIO) (default)

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

0x3 (R/W): #NAND_RD (CARD)
 0x2 (R/W): SOUT1 (FSIO Ch.1)
 0x1 (R/W): USI_DO (USI)
 0x0 (R/W): P01 (GPIO) (default)

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

0x3 (R/W): #NAND_WR (CARD)
 0x2 (R/W): SIN1 (FSIO Ch.1)
 0x1 (R/W): USI_DI (USI)
 0x0 (R/W): P00 (GPIO) (default)

P0[7:4] Port Function Select Register (PMUX_P0_47)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P0[7:4] Port Function Select Register (PMUX_P0_47)	0x300801 (8 bits)	D7-6	CFP07[1:0]	P07 port function select	CFP07[1:0]	Function	0x0	R/W	Write-protected
					0x3	PWM_L			
					0x2	I2S_MCLK			
					0x1	#SRDY0			
		0x0	P07						
		D5-4	CFP06[1:0]	P06 port function select	CFP06[1:0]	Function	0x0	R/W	
					0x3	PWM_H			
					0x2	I2S_SCLK			
					0x1	SCLK0			
		0x0	P06						
		D3-2	CFP05[1:0]	P05 port function select	CFP05[1:0]	Function	0x0	R/W	
					0x3	T16A_ATMB_0			
					0x2	I2S_WS			
					0x1	SOUT0			
		0x0	P05						
		D1-0	CFP04[1:0]	P04 port function select	CFP04[1:0]	Function	0x0	R/W	
0x3	T16A_ATMA_0								
0x2	I2S_SDO								
0x1	SIN0								
0x0	P04								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

0x3 (R/W): PWM_L (T16P)
 0x2 (R/W): I2S_MCLK (I2S)
 0x1 (R/W): #SRDY0 (FSIO Ch.0)
 0x0 (R/W): P07 (GPIO) (default)

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

0x3 (R/W): PWM_H (T16P)
 0x2 (R/W): I2S_SCLK (I2S)
 0x1 (R/W): SCLK0 (FSIO Ch.0)
 0x0 (R/W): P06 (GPIO) (default)

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

0x3 (R/W): T16A_ATMB_0 (T16A5 Ch.0)
 0x2 (R/W): I2S_WS (I2S)
 0x1 (R/W): SOUT0 (FSIO Ch.0)
 0x0 (R/W): P05 (GPIO) (default)

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

0x3 (R/W): T16A_ATMA_0 (T16A5 Ch.0)
 0x2 (R/W): I2S_SDO (I2S)
 0x1 (R/W): SIN0 (FSIO Ch.0)
 0x0 (R/W): P04 (GPIO) (default)

P1[3:0] Port Function Select Register (PMUX_P1_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P1[3:0] Port Function Select Register (PMUX_P1_03)	0x300802 (8 bits)	D7-6	CFP13[1:0]	P13 port function select	CFP13[1:0]	Function	0x0	R/W	Write-protected
					0x3	T16A_ATMB_1			
					0x2	FPDAT11			
					0x1	USIL_CK			
		0x0	P13						
		D5-4	CFP12[1:0]	P12 port function select	CFP12[1:0]	Function	0x0	R/W	
					0x3	T16A_ATMA_1			
					0x2	FPDAT10			
					0x1	USIL_CS			
		0x0	P12						
		D3-2	CFP11[1:0]	P11 port function select	CFP11[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	FPDAT9			
					0x1	USIL_DO			
		0x0	P11						
		D1-0	CFP10[1:0]	P10 port function select	CFP10[1:0]	Function	0x0	R/W	
0x3	reserved								
0x2	FPDAT8								
0x1	USIL_DI								
0x0	P10								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

0x3 (R/W): T16A_ATMB_1 (T16A5 Ch.1)
 0x2 (R/W): FPDAT11 (LCDC)
 0x1 (R/W): USIL_CK (USIL)
 0x0 (R/W): P13 (GPIO) (default)

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

0x3 (R/W): T16A_ATMA_1 (T16A5 Ch.1)
 0x2 (R/W): FPDAT10 (LCDC)
 0x1 (R/W): USIL_CS (USIL)
 0x0 (R/W): P12 (GPIO) (default)

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

0x3 (R/W): Reserved
 0x2 (R/W): FPDAT9 (LCDC)
 0x1 (R/W): USIL_DO (USIL)
 0x0 (R/W): P11 (GPIO) (default)

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

0x3 (R/W): Reserved
 0x2 (R/W): FPDAT8 (LCDC)
 0x1 (R/W): USIL_DI (USIL)
 0x0 (R/W): P10 (GPIO) (default)

P1[7:4] Port Function Select Register (PMUX_P1_47)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P1[7:4] Port Function Select Register (PMUX_P1_47)	0x300803 (8 bits)	D7–6	CFP17[1:0]	P17 port function select	CFP17[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	FPDAT15 P17			
					0x1	DPCO			
		D5–4	CFP16[1:0]	P16 port function select	CFP16[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	FPDAT14 P16			
					0x1	DST1			
		D3–2	CFP15[1:0]	P15 port function select	CFP15[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	FPDAT13 P15			
					0x1	DST0			
		D1–0	CFP14[1:0]	P14 port function select	CFP14[1:0]	Function	0x0	R/W	
					0x3	CMU_CLK P14			
					0x2	FPDAT12 P14			
					0x1	FPDAT19 P14			
0x0									

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): FPDAT15 (LCDC)
- 0x1 (R/W): P17 (GPIO)
- 0x0 (R/W): DPCO (DBG) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): FPDAT14 (LCDC)
- 0x1 (R/W): P16 (GPIO)
- 0x0 (R/W): DST1 (DBG) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): FPDAT13 (LCDC)
- 0x1 (R/W): P15 (GPIO)
- 0x0 (R/W): DST0 (DBG) (default)

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

- 0x3 (R/W): CMU_CLK (CMU)
- 0x2 (R/W): FPDAT12 (LCDC)
- 0x1 (R/W): FPDAT19 (LCDC)
- 0x0 (R/W): P14 (GPIO) (default)

P2[1:0] Port Function Select Register (PMUX_P2_01)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P2[1:0] Port Function Select Register (PMUX_P2_01)	0x300804 (8 bits)	D7–4	–	reserved	–	–	–	–	0 when being read.
					D3–2	CFP21[1:0]	P21 port function select	CFP21[1:0]	Function
		0x3	reserved						
		0x2	reserved						
		0x1	P21						
		0x0	SDCLK						
		D1–0	CFP20[1:0]	P20 port function select	CFP20[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P20			
0x0	SDCKE								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:4] Reserved

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): P21 (GPIO)
- 0x0 (R/W): SDCLK (SDRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): P20 (GPIO)
- 0x0 (R/W): SDCKE (SDRAMC) (default)

P3[3:0] Port Function Select Register (PMUX_P3_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P3[3:0] Port Function Select Register (PMUX_P3_03)	0x300806 (8 bits)	D7-6	CFP33[1:0]	P33 port function select	CFP33[1:0]	Function	0x0	R/W	Write-protected
					0x3	REMC_I			
					0x2	TFT_CTL3			
					0x1	reserved			
		0x0	P33						
		D5-4	CFP32[1:0]	P32 port function select	CFP32[1:0]	Function	0x0	R/W	
					0x3	REMC_O			
					0x2	TFT_CTL2			
					0x1	reserved			
		0x0	P32						
		D3-2	CFP31[1:0]	P31 port function select	CFP31[1:0]	Function	0x0	R/W	
					0x3	T16A_ATMB_0			
					0x2	TFT_CTL1			
					0x1	reserved			
		0x0	P31						
		D1-0	CFP30[1:0]	P30 port function select	CFP30[1:0]	Function	0x0	R/W	
0x3	T16A_ATMA_0								
0x2	TFT_CTL0								
0x1	reserved								
0x0	P30								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): REMC_I (REMC)
- 0x2 (R/W): TFT_CTL3 (LCDC)
- 0x1 (R/W): Reserved
- 0x0 (R/W): P33 (GPIO) (default)

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

- 0x3 (R/W): REMC_O (REMC)
- 0x2 (R/W): TFT_CTL2 (LCDC)
- 0x1 (R/W): Reserved
- 0x0 (R/W): P32 (GPIO) (default)

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

- 0x3 (R/W): T16A_ATMB_0 (T16A5 Ch.0)
- 0x2 (R/W): TFT_CTL1 (LCDC)
- 0x1 (R/W): Reserved
- 0x0 (R/W): P31 (GPIO) (default)

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

- 0x3 (R/W): T16A_ATMA_0 (T16A5 Ch.0)
- 0x2 (R/W): TFT_CTL0 (LCDC)
- 0x1 (R/W): Reserved
- 0x0 (R/W): P30 (GPIO) (default)

P3[6:4] Port Function Select Register (PMUX_P3_46)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P3[6:4] Port Function Select Register (PMUX_P3_46)	0x300807 (8 bits)	D7-6	-	reserved	-	-	-	-	0 when being read.
		D5-4	CFP36[1:0]	P36 port function select	CFP36[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	reserved			
					0x1	P36			
		0x0	DST2						
		D3-2	CFP35[1:0]	P35 port function select	CFP35[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P35			
		0x0	DSIO						
		D1-0	CFP34[1:0]	P34 port function select	CFP34[1:0]	Function	0x0	R/W	
0x3	reserved								
0x2	reserved								
0x1	P34								
0x0	DCLK								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] Reserved

D[5:4] CFP36[1:0]: P36 Port Function Select Bits

0x3 (R/W): Reserved

0x2 (R/W): Reserved

0x1 (R/W): P36 (GPIO)

0x0 (R/W): DST2 (DBG) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): Reserved

0x1 (R/W): P35 (GPIO)

0x0 (R/W): DSIO (DBG) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): Reserved

0x1 (R/W): P34 (GPIO)

0x0 (R/W): DCLK (DBG) (default)

P4[2:0] Port Function Select Register (PMUX_P4_02)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P4[2:0] Port Function Select Register (PMUX_P4_02)	0x300808 (8 bits)	D7-6	-	reserved	-	-	-	-	0 when being read.
		D5-4	CFP42[1:0]	P42 port function select	CFP42[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	FPDAT16			
					0x1	P42			
		0x0	A23						
		D3-2	CFP41[1:0]	P41 port function select	CFP41[1:0]	Function	0x0	R/W	
					0x3	#NAND_WR			
					0x2	FPDAT17			
					0x1	P41			
		0x0	A22						
		D1-0	CFP40[1:0]	P40 port function select	CFP40[1:0]	Function	0x0	R/W	
0x3	#NAND_RD								
0x2	FPDAT18								
0x1	P40								
0x0	A21								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] Reserved

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): FPDAT16 (LCDC)
- 0x1 (R/W): P42 (GPIO)
- 0x0 (R/W): A23 (SRAMC) (default)

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

- 0x3 (R/W): #NAND_WR (CARD)
- 0x2 (R/W): FPDAT17 (LCDC)
- 0x1 (R/W): P41 (GPIO)
- 0x0 (R/W): A22 (SRAMC) (default)

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

- 0x3 (R/W): #NAND_RD (CARD)
- 0x2 (R/W): FPDAT18 (LCDC)
- 0x1 (R/W): P40 (GPIO)
- 0x0 (R/W): A21 (SRAMC) (default)

P5[3:0] Port Function Select Register (PMUX_P5_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P5[3:0] Port Function Select Register (PMUX_P5_03)	0x30080a (8 bits)	D7-6	CFP53[1:0]	P53 port function select	CFP53[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	reserved			
					0x1	P53			
		0x0	#CE10						
		D5-4	CFP52[1:0]	P52 port function select	CFP52[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#CE5			
					0x1	P52			
		0x0	#CE9						
		D3-2	CFP51[1:0]	P51 port function select	CFP51[1:0]	Function	0x0	R/W	
					0x3	reserved			
0x2	#CE4								
0x1	P51								
0x0	#CE8								
D1-0	CFP50[1:0]	P50 port function select	CFP50[1:0]	Function	0x0	R/W			
			0x3	reserved					
			0x2	#SDCS					
			0x1	P50					
0x0	#CE7								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): P53 (GPIO)
- 0x0 (R/W): #CE10 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): #CE5 (SRAMC)
- 0x1 (R/W): P52 (GPIO)
- 0x0 (R/W): #CE9 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): #CE4 (SRAMC)
- 0x1 (R/W): P51 (GPIO)
- 0x0 (R/W): #CE8 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): #SDCS (SDRAMC)
- 0x1 (R/W): P50 (GPIO)
- 0x0 (R/W): #CE7 (SRAMC) (default)

P5[6:4] Port Function Select Register (PMUX_P5_46)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P5[6:4] Port Function Select Register (PMUX_P5_46)	0x30080b (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.	
		D5–4	CFP56[1:0]	P56 port function select	CFP56[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	reserved			
					0x1	P56			
		0x0	#WRH/#BSH						
		D3–2	CFP55[1:0]	P55 port function select	CFP55[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P55			
		0x0	#WRL						
		D1–0	CFP54[1:0]	P54 port function select	CFP54[1:0]	Function	0x0	R/W	
0x3	reserved								
0x2	reserved								
0x1	P54								
0x0	#RD								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] Reserved**D[5:4] CFP56[1:0]: P56 Port Function Select Bits**

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): P56 (GPIO)
- 0x0 (R/W): #WRH/#BSH (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): P55 (GPIO)
- 0x0 (R/W): #WRL (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): P54 (GPIO)
- 0x0 (R/W): #RD (SRAMC) (default)

P60 Port Function Select Register (PMUX_P6_0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P60 Port Function Select Register (PMUX_P6_0)	0x30080c (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.	
		D1–0	CFP60[1:0]	P60 port function select	CFP60[1:0]	Function	0x0	R/W	Write-protected
					0x3	#WDT_NMI			
					0x2	WDT_CLK			
					0x1	#WAIT			
					0x0	P60			

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:2] Reserved**D[1:0] CFP60[1:0]: P60 Port Function Select Bits**

- 0x3 (R/W): #WDT_NMI (WDT)
- 0x2 (R/W): WDT_CLK (WDT)
- 0x1 (R/W): #WAIT (SRAMC)
- 0x0 (R/W): P60 (GPIO) (default)

P7[3:0] Port Function Select Register (PMUX_P7_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P7[3:0] Port Function Select Register (PMUX_P7_03)	0x30080e (8 bits)	D7–6	CFP73[1:0]	P73 port function select	CFP73[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	reserved			
					0x1	AIN3			
		D5–4	CFP72[1:0]	P72 port function select	CFP72[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	PWM_EXCL			
					0x1	AIN2			
		D3–2	CFP71[1:0]	P71 port function select	CFP71[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	T16A_EXCL_1			
					0x1	AIN1			
D1–0	CFP70[1:0]	P70 port function select	CFP70[1:0]	Function	0x0	R/W			
			0x3	reserved					
			0x2	T16A_EXCL_0					
			0x1	AIN0					
					0x0				

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

0x3 (R/W): Reserved

0x2 (R/W): Reserved

0x1 (R/W): AIN3 (ADC10)

0x0 (R/W): P73 (GPIO) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): PWM_EXCL (T16P)

0x1 (R/W): AIN2 (ADC10)

0x0 (R/W): P72 (GPIO) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): T16A_EXCL_1 (T16A5 Ch.1)

0x1 (R/W): AIN1 (ADC10)

0x0 (R/W): P71 (GPIO) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): T16A_EXCL_0 (T16A5 Ch.0)

0x1 (R/W): AIN0 (ADC10)

0x0 (R/W): P70 (GPIO) (default)

P7[5:4] Port Function Select Register (PMUX_P7_45)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
P7[5:4] Port Function Select Register (PMUX_P7_45)	0x30080f (8 bits)	D7–4	–	reserved	–	–	–	–	0 when being read.	
		D3–2	CFP75[1:0]	P75 port function select	CFP75[1:0]	Function	0x0	R/W		
					0x3	#ADTRIG				
					0x2	#WAIT				
					0x1	AIN5				
		D1–0	CFP74[1:0]	P74 port function select	CFP74[1:0]	Function	0x0	R/W		
					0x3	reserved				
					0x2	reserved				
					0x1	AIN4				
							0x0			

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:4] Reserved

D[3:2] CFP75[1:0]: P75 Port Function Select Bits

0x3 (R/W): #ADTRIG (ADC10)

0x2 (R/W): #WAIT (SRAMC)

0x1 (R/W): AIN5 (ADC10)

0x0 (R/W): P75 (GPIO) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): Reserved

0x1 (R/W): AIN4 (ADC10)

0x0 (R/W): P74 (GPIO) (default)

P8[3:0] Port Function Select Register (PMUX_P8_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P8[3:0] Port Function Select Register (PMUX_P8_03)	0x300810 (8 bits)	D7-6	CFP83[1:0]	P83 port function select	CFP83[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	USIL_DO			
					0x1	FPDRDY			
		D5-4	CFP82[1:0]	P82 port function select	CFP82[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	USIL_DI			
					0x1	FPSHIFT			
		D3-2	CFP81[1:0]	P81 port function select	CFP81[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	USIL_CK			
					0x1	FPLINE			
		D1-0	CFP80[1:0]	P80 port function select	CFP80[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	USIL_CS			
					0x1	FPFRAME			
					0x0				

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

0x3 (R/W): Reserved

0x2 (R/W): USIL_DO (USIL)

0x1 (R/W): FPDRDY (LCDC)

0x0 (R/W): P83 (GPIO) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): USIL_DI (USIL)

0x1 (R/W): FPSHIFT (LCDC)

0x0 (R/W): P82 (GPIO) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): USIL_CK (USIL)

0x1 (R/W): FPLINE (LCDC)

0x0 (R/W): P81 (GPIO) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): USIL_CS (USIL)

0x1 (R/W): FPFRAME (LCDC)

0x0 (R/W): P80 (GPIO) (default)

P9[3:0] Port Function Select Register (PMUX_P9_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P9[3:0] Port Function Select Register (PMUX_P9_03)	0x300812 (8 bits)	D7-6	CFP93[1:0]	P93 port function select	CFP93[1:0]	Function	0x0	R/W	Write-protected
					0x3	#SRDY0			
					0x2	LCD_D3			
					0x1	FPDAT3			
		D5-4	CFP92[1:0]	P92 port function select	CFP92[1:0]	Function	0x0	R/W	
					0x3	SCLK0			
					0x2	LCD_D2			
					0x1	FPDAT2			
		D3-2	CFP91[1:0]	P91 port function select	CFP91[1:0]	Function	0x0	R/W	
					0x3	SOUT0			
					0x2	LCD_D1			
					0x1	FPDAT1			
		D1-0	CFP90[1:0]	P90 port function select	CFP90[1:0]	Function	0x0	R/W	
					0x3	SIN0			
					0x2	LCD_D0			
					0x1	FPDAT0			
					0x0				
					P90				

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): #SRDY0 (FSIO Ch.0)
- 0x2 (R/W): LCD_D3 (USIL)
- 0x1 (R/W): FPDAT3 (LCDC)
- 0x0 (R/W): P93 (GPIO) (default)

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

- 0x3 (R/W): SCLK0 (FSIO Ch.0)
- 0x2 (R/W): LCD_D2 (USIL)
- 0x1 (R/W): FPDAT2 (LCDC)
- 0x0 (R/W): P92 (GPIO) (default)

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

- 0x3 (R/W): SOUT0 (FSIO Ch.0)
- 0x2 (R/W): LCD_D1 (USIL)
- 0x1 (R/W): FPDAT1 (LCDC)
- 0x0 (R/W): P91 (GPIO) (default)

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

- 0x3 (R/W): SIN0 (FSIO Ch.0)
- 0x2 (R/W): LCD_D0 (USIL)
- 0x1 (R/W): FPDAT0 (LCDC)
- 0x0 (R/W): P90 (GPIO) (default)

P9[7:4] Port Function Select Register (PMUX_P9_47)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P9[7:4] Port Function Select Register (PMUX_P9_47)	0x300813 (8 bits)	D7-6	CFP97[1:0]	P97 port function select	CFP97[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	LCD_D7			
					0x1	FPDAT7			
		0x0	P97						
		D5-4	CFP96[1:0]	P96 port function select	CFP96[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	LCD_D6			
					0x1	FPDAT6			
		0x0	P96						
		D3-2	CFP95[1:0]	P95 port function select	CFP95[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	LCD_D5			
					0x1	FPDAT5			
		0x0	P95						
		D1-0	CFP94[1:0]	P94 port function select	CFP94[1:0]	Function	0x0	R/W	
0x3	reserved								
0x2	LCD_D4								
0x1	FPDAT4								
0x0	P94								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): LCD_D7 (USIL)
- 0x1 (R/W): FPDAT7 (LCDC)
- 0x0 (R/W): P97 (GPIO) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): LCD_D6 (USIL)
- 0x1 (R/W): FPDAT6 (LCDC)
- 0x0 (R/W): P96 (GPIO) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): LCD_D5 (USIL)
- 0x1 (R/W): FPDAT5 (LCDC)
- 0x0 (R/W): P95 (GPIO) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): LCD_D4 (USIL)
- 0x1 (R/W): FPDAT4 (LCDC)
- 0x0 (R/W): P94 (GPIO) (default)

PA[3:0] Port Function Select Register (PMUX_PA_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
PA[3:0] Port Function Select Register (PMUX_PA_03)	0x300814 (8 bits)	D7–6	CFPA3[1:0]	PA3 port function select	CFPA3[1:0]	Function	0x0	R/W	Write-protected		
					0x3	FPDAT23					
					0x2	FPDAT19					
					0x1	#SRDY1					
				0x0	PA3						
		D5–4	CFPA2[1:0]	PA2 port function select	CFPA2[1:0]	Function	0x0	R/W			
					0x3	FPDAT22					
					0x2	FPDAT18					
					0x1	SCLK1					
				0x0	PA2						
		D3–2	CFPA1[1:0]	PA1 port function select	CFPA1[1:0]	Function	0x0	R/W			
					0x3	FPDAT21					
0x2	FPDAT17										
0x1	SOUT1										
		0x0	PA1								
D1–0	CFPA0[1:0]	PA0 port function select	CFPA0[1:0]	Function	0x0	R/W					
			0x3	FPDAT20							
			0x2	FPDAT16							
			0x1	SIN1							
		0x0	PA0								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): FPDAT23 (LCDC)
- 0x2 (R/W): FPDAT19 (LCDC)
- 0x1 (R/W): #SRDY1 (FSIO Ch.1)
- 0x0 (R/W): PA3 (GPIO) (default)

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

- 0x3 (R/W): FPDAT22 (LCDC)
- 0x2 (R/W): FPDAT18 (LCDC)
- 0x1 (R/W): SCLK1 (FSIO Ch.1)
- 0x0 (R/W): PA2 (GPIO) (default)

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

- 0x3 (R/W): FPDAT21 (LCDC)
- 0x2 (R/W): FPDAT17 (LCDC)
- 0x1 (R/W): SOUT1 (FSIO Ch.1)
- 0x0 (R/W): PA1 (GPIO) (default)

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

- 0x3 (R/W): FPDAT20 (LCDC)
- 0x2 (R/W): FPDAT16 (LCDC)
- 0x1 (R/W): SIN1 (FSIO Ch.1)
- 0x0 (R/W): PA0 (GPIO) (default)

PA[6:4] Port Function Select Register (PMUX_PA_46)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
PA[6:4] Port Function Select Register (PMUX_PA_46)	0x300815 (8 bits)	D7–6	–	reserved	–	–	–	–	0 when being read.	
		D5–4	CFPA6[1:0]	PA6 port function select	CFPA6[1:0]	Function	0x0	R/W		
					0x3	reserved				
					0x2	reserved				
					0x1	#ADTRIG				
				0x0	PA6					
		D3–2	CFPA5[1:0]	PA5 port function select	CFPA5[1:0]	Function	0x0	R/W		
					0x3	REMC_I				
					0x2	T16A_ATMB_1				
					0x1	PA5				
				0x0	A25					
D1–0	CFPA4[1:0]	PA4 port function select	CFPA4[1:0]	Function	0x0	R/W				
			0x3	REMC_O						
			0x2	T16A_ATMA_1						
			0x1	PA4						
		0x0	A24							

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] Reserved

D[5:4] CFPA6[1:0]: PA6 Port Function Select Bits

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): #ADTRIG (ADC10)
- 0x0 (R/W): PA6 (GPIO) (default)

D[3:2] CFPA5[1:0]: PA5 Port Function Select Bits

- 0x3 (R/W): REMC_I (REMC)
- 0x2 (R/W): T16A_ATMB_1 (T16A5 Ch.1)
- 0x1 (R/W): PA5 (GPIO)
- 0x0 (R/W): A25 (SRAMC) (default)

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

- 0x3 (R/W): REMC_O (REMC)
- 0x2 (R/W): T16A_ATMA_1 (T16A5 Ch.1)
- 0x1 (R/W): PA4 (GPIO)
- 0x0 (R/W): A24 (SRAMC) (default)

PB[3:0] Port Function Select Register (PMUX_PB_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PB[3:0] Port Function Select Register (PMUX_PB_03)	0x300816 (8 bits)	D7-6	CFPB3[1:0]	PB3 port function select	CFPB3[1:0]	Function	0x0	R/W	Write-protected
					0x3	PWM_L			
					0x2	I2S_MCLK			
					0x1	FPDAT11			
		0x0	PB3						
		D5-4	CFPB2[1:0]	PB2 port function select	CFPB2[1:0]	Function	0x0	R/W	
					0x3	PWM_H			
					0x2	I2S_SCLK			
					0x1	FPDAT10			
		0x0	PB2						
		D3-2	CFPB1[1:0]	PB1 port function select	CFPB1[1:0]	Function	0x0	R/W	
					0x3	reserved			
0x2	I2S_WS								
0x1	FPDAT9								
0x0	PB1								
D1-0	CFPB0[1:0]	PB0 port function select	CFPB0[1:0]	Function	0x0	R/W			
			0x3	reserved					
			0x2	I2S_SDO					
			0x1	FPDAT8					
0x0	PB0								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): PWM_L (T16P)
- 0x2 (R/W): I2S_MCLK (I2S)
- 0x1 (R/W): FPDAT11 (LCDC)
- 0x0 (R/W): PB3 (GPIO) (default)

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

- 0x3 (R/W): PWM_H (T16P)
- 0x2 (R/W): I2S_SCLK (I2S)
- 0x1 (R/W): FPDAT10 (LCDC)
- 0x0 (R/W): PB2 (GPIO) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): I2S_WS (I2S)
- 0x1 (R/W): FPDAT9 (LCDC)
- 0x0 (R/W): PB1 (GPIO) (default)

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

0x3 (R/W): Reserved

0x2 (R/W): I2S_SDO (I2S)

0x1 (R/W): FPDAT8 (LCDC)

0x0 (R/W): PB0 (GPIO) (default)

PB[7:4] Port Function Select Register (PMUX_PB_47)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PB[7:4] Port Function Select Register (PMUX_PB_47)	0x300817 (8 bits)	D7-6	CFPB7[1:0]	PB7 port function select	CFPB7[1:0]	Function	0x0	R/W	Write-protected
					0x3	FPDAT23 reserved			
					0x2	FPDAT15 PB7			
					0x1	FPDAT15 PB7			
		D5-4	CFPB6[1:0]	PB6 port function select	CFPB6[1:0]	Function	0x0	R/W	
					0x3	FPDAT22 reserved			
					0x2	FPDAT14 PB6			
					0x1	FPDAT14 PB6			
		D3-2	CFPB5[1:0]	PB5 port function select	CFPB5[1:0]	Function	0x0	R/W	
					0x3	FPDAT21 reserved			
					0x2	FPDAT13 PB5			
					0x1	FPDAT13 PB5			
		D1-0	CFPB4[1:0]	PB4 port function select	CFPB4[1:0]	Function	0x0	R/W	
					0x3	FPDAT20 reserved			
					0x2	FPDAT12 PB4			
					0x1	FPDAT12 PB4			
					0x0				

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] CFPB7[1:0]: PB7 Port Function Select Bits

0x3 (R/W): FPDAT23 (LCDC)

0x2 (R/W): Reserved

0x1 (R/W): FPDAT15 (LCDC)

0x0 (R/W): PB7 (GPIO) (default)

D[5:4] CFPB6[1:0]: PB6 Port Function Select Bits

0x3 (R/W): FPDAT22 (LCDC)

0x2 (R/W): Reserved

0x1 (R/W): FPDAT14 (LCDC)

0x0 (R/W): PB6 (GPIO) (default)

D[3:2] CFPB5[1:0]: PB5 Port Function Select Bits

0x3 (R/W): FPDAT21 (LCDC)

0x2 (R/W): Reserved

0x1 (R/W): FPDAT13 (LCDC)

0x0 (R/W): PB5 (GPIO) (default)

D[1:0] CFPB4[1:0]: PB4 Port Function Select Bits

0x3 (R/W): FPDAT20 (LCDC)

0x2 (R/W): Reserved

0x1 (R/W): FPDAT12 (LCDC)

0x0 (R/W): PB4 (GPIO) (default)

PC[3:0] Port Function Select Register (PMUX_PC_03)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PC[3:0] Port Function Select Register (PMUX_PC_03)	0x300818 (8 bits)	D7-6	CFPC3[1:0]	PC3 port function select	CFPC3[1:0]	Function	0x0	R/W	Write-protected
					0x3	reserved			
					0x2	reserved			
					0x1	PC3			
		0x0	D11						
		D5-4	CFPC2[1:0]	PC2 port function select	CFPC2[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	PC2			
		0x0	D10						
		D3-2	CFPC1[1:0]	PC1 port function select	CFPC1[1:0]	Function	0x0	R/W	
					0x3	reserved			
0x2	reserved								
0x1	PC1								
0x0	D9								
D1-0	CFPC0[1:0]	PC0 port function select	CFPC0[1:0]	Function	0x0	R/W			
			0x3	reserved					
			0x2	reserved					
			0x1	PC0					
0x0	D8								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): PC3 (GPIO)
- 0x0 (R/W): D11 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): PC2 (GPIO)
- 0x0 (R/W): D10 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): PC1 (GPIO)
- 0x0 (R/W): D9 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): PC0 (GPIO)
- 0x0 (R/W): D8 (SRAMC) (default)

PC[7:4] Port Function Select Register (PMUX_PC_47)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PC[7:4] Port Function Select Register (PMUX_PC_47)	0x300819 (8 bits)	D7-6	CFPC7[1:0]	PC7 port function select	CFPC7[1:0] Function		0x0	R/W	Write-protected
					0x3	reserved			
					0x2	reserved			
					0x1	PC7			
	0x0	D15							
	D5-4	CFPC6[1:0]	PC6 port function select	CFPC6[1:0] Function		0x0	R/W		
				0x3	reserved				
				0x2	reserved				
				0x1	PC6				
	0x0	D14							
	D3-2	CFPC5[1:0]	PC5 port function select	CFPC5[1:0] Function		0x0	R/W		
				0x3	reserved				
				0x2	reserved				
				0x1	PC5				
	0x0	D13							
	D1-0	CFPC4[1:0]	PC4 port function select	CFPC4[1:0] Function		0x0	R/W		
0x3				reserved					
0x2				reserved					
0x1				PC4					
0x0	D12								

The GPIO pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): PC7 (GPIO)
- 0x0 (R/W): D15 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): PC6 (GPIO)
- 0x0 (R/W): D14 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): PC5 (GPIO)
- 0x0 (R/W): D13 (SRAMC) (default)

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

- 0x3 (R/W): Reserved
- 0x2 (R/W): Reserved
- 0x1 (R/W): PC4 (GPIO)
- 0x0 (R/W): D12 (SRAMC) (default)

Port Noise Filter Control Register (GPIO_FILTER)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Port Noise Filter Control Register (GPIO_FILTER)	0x30083e (8 bits)	D7-1	–	reserved	–	–	–	0 when being read.
		D0	ANFEN	input port noise filter enable	1 Enable 0 Disable	0	R/W	Write-protected

D[7:1] Reserved**D0 ANFEN: Input Port Noise Filter Enable Bit**

Enables or disables the noise filters for peripheral input ports.

- 1 (R/W): Enabled
- 0 (R/W): Disabled (default)

Setting 1 to ANFEN enables the noise filters to remove noise on the signals input from the ports shown below.

USI: USI_DI, USI_CS, USI_CK
 USIL: USIL_DI, USIL_CS, USIL_CK
 FSIO: SIN0, SIN1, SCLK0, SCLK1, #SRDY0, #SRDY1
 REMC: REMC_I
 T16A5: T16A_EXCL_0, T16A_EXCL_1, T16A_ATMA_0, T16A_ATMA_1, T16A_ATMB_0,
 T16A_ATMB_1
 ADC10: #ADTRIG
 GPIO: FPT0–FPTF interrupt ports (See note below.)

When ANFEN is set to 0 (default), the input signals bypass the noise filters.

- Notes:**
- These noise filters cannot be enabled individually.
 - The noise filters are not effective if these ports are used as general-purpose input port. However, the noise filters for the general-purpose input ports that are selected as FPT interrupt ports (FPT0 to FPTF) are effective.

GPIO/PMUX Write Protect Register (GPIO_PROTECT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
GPIO/PMUX Write Protect Register (GPIO_PROTECT)	0x30083f (8 bits)	D7–0	PPROT[7:0]	GPIO/PMUX register protect flag	Writing 10010110 (0x96) removes the write protection of the GPIO registers (0x300320–0x30032c and 0x30083e) and PMUX registers (0x300800–0x300819). Writing another value set the write protection.	0x0	R/W	

D[7:0] PPROT[7:0]: GPIO/PMUX Register Protect Flag Bits

Enables or disables write protection of the GPIO registers (0x300320–0x30032c and 0x30083e) and PMUX registers (0x300800–0x300819).

0x96 (R/W): Disable write protection

Other than 0x96 (R/W): Write-protect the register (default: 0x0)

Before altering any GPIO/PMUX register within addresses 0x300320–0x30032c, 0x300800–0x300819, and 0x30083e, write data 0x96 to PPROT[7:0] to disable write protection. If PPROT[7:0] is set to other than 0x96, even if an attempt is made to alter any GPIO/PMUX register by executing a write instruction, the content of the register will not be altered even though the instruction may have been executed without a problem. Once PPROT[7:0] is set to 0x96, the GPIO/PMUX registers can be rewritten any number of times until being reset to other than 0x96. When rewriting the GPIO/PMUX registers has finished, PPROT[7:0] should be set to other than 0x96 to prevent accidental writing to the GPIO/PMUX registers.

25 A/D Converter (ADC10)

25.1 ADC10 Module Overview

The S1C33L26 incorporates an A/D converter with the following features:

- Conversion method: Successive approximation type
- Resolution: 10 bits
- Input channels: Max. 6 channels
- A/D conversion clock: Max. 2 MHz
- Sampling rate: Max. 100 ksp/s
- Analog input voltage range: V_{SS} to AV_{DD}
- Sampling & hold circuit included
- Supports two conversion modes:
 - One-time conversion mode
(for single channel or multi-channels)
 - Continuous conversion mode
(for single channel or multi-channels, terminated with software)
- Supports three conversion triggers:
 - Software trigger
 - External trigger (input from the #ADTRIG pin)
 - T8 Ch.2 underflow trigger
- The conversion results can be read as 16-bit data with the 10-bit converted data aligned to left or right.
- Two types of interrupts can be generated: Conversion completion interrupt
Conversion data overwrite error interrupt

Figure 25.1.1 shows the A/D converter configuration.

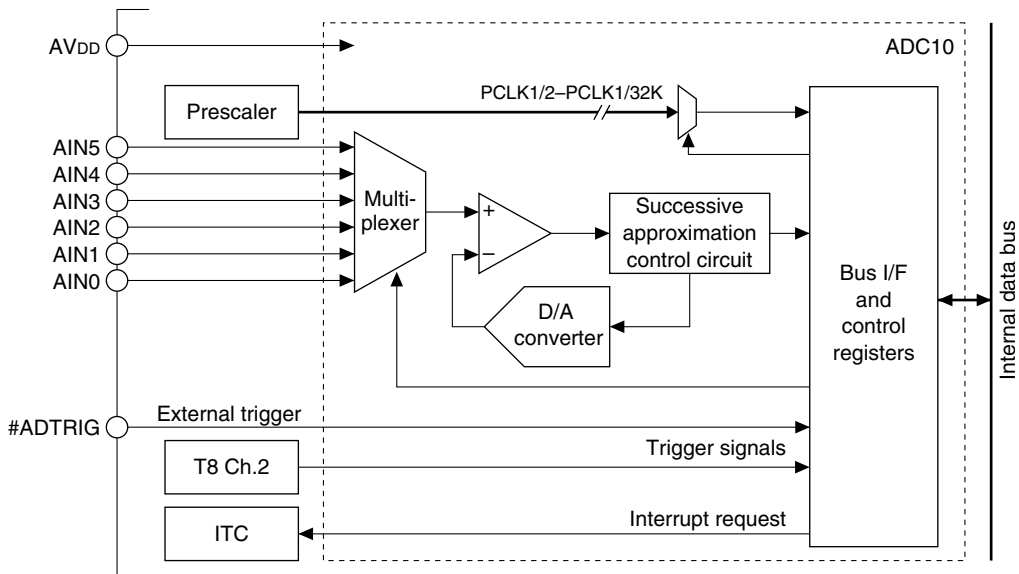


Figure 25.1.1 A/D Converter Configuration

25.2 ADC10 Input Pins

Table 25.2.1 lists the A/D converter input pins.

Table 25.2.1 List of A/D Converter Input Pins

Pin name	I/O	Qty	Function
AIN[5:0]	I	6	Analog signal input pins AIN0 (Ch.0) to AIN5 (Ch.5) Input the analog signals to be A/D converted. The analog input voltage AV_{IN} must be within the range of $V_{SS} \leq AV_{IN} \leq AV_{DD}$.
#ADTRIG	I	1	External trigger input pin Input a trigger signal to start A/D conversion from an external source.
AV _{DD}	–	1	Analog power-supply pin Always supply the HV _{DD} voltage even if the A/D converter is not used.

The A/D converter input pins (AIN[5:0], #ADTRIG) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as A/D converter input pins.

For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

Note: The ADC10 converts the digital values input to the P70, P71, P72, P73, P74, and P75 as analog inputs even if these ports are configured as general-purpose input ports by the port function select bits.

25.3 A/D Converter Settings

Make the following settings before starting A/D conversion.

- (1) Set the analog input pins. See Section 25.2.
- (2) Set the A/D conversion clock.
- (3) Select the A/D conversion start and end channels.
- (4) Select the A/D conversion mode.
- (5) Select the A/D conversion trigger source.
- (6) Set the sampling time.
- (7) Select the conversion result storing mode.
- (8) When using A/D converter interrupts, set interrupt conditions. See Section 25.5.

Note: Make sure the A/D converter is disabled (ADEN/ADC10_CTL register = 0) before changing the above settings. Changing the settings while the A/D converter is enabled may cause a malfunction.

25.3.1 A/D Conversion Clock Setting

To use the A/D converter, the clocks used in the A/D converter must be supplied by turning on the peripheral module clock (PCLK1) output from the clock management unit (CMU) and the PCLK1 division clocks output from the Prescaler (PSC Ch.0). For more information on clock control, see the “Clock Management Unit (CMU)” and “Prescaler (PSC)” chapters.

The A/D conversion clock can be selected from the 15 PCLK1 division clocks supplied by the Prescaler. Use ADDF[3:0]/ADC10_CLK register for this selection as shown in Table 25.3.1.1.

- Notes:**
- For the A/D conversion clock frequency range that can be used for this A/D converter, see “A/D Converter Characteristics” in the “Electrical Characteristics” chapter.
 - 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 may cause the A/D converter to operate erratically.

Table 25.3.1.1 A/D Conversion Clock (PCLK1 Division Ratio) Selections

ADDF[3:0]	Division Ratio
0xf	Reserved
0xe	1/32768
0xd	1/16384
0xc	1/8192
0xb	1/4096
0xa	1/2048
0x9	1/1024
0x8	1/512
0x7	1/256
0x6	1/128
0x5	1/64
0x4	1/32
0x3	1/16
0x2	1/8
0x1	1/4
0x0	1/2

(Default: 0x0)

25.3.2 Selecting A/D 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 ADCS[2:0]/ADC10_TRG register and ADCE[2:0]/ADC10_TRG register, respectively.

Table 25.3.2.1 Relationship between ADCS/ADCE and Input Channels

ADCS[2:0]/ADCE[2:0]	Channel selected
0x7–0x6	Reserved
0x5	AIN5
0x4	AIN4
0x3	AIN3
0x2	AIN2
0x1	AIN1
0x0	AIN0

(Default: 0x0)

Example: Operation of one A/D conversion

ADCS[2:0] = 0, ADCE[2:0] = 0

Converted only in AIN0

ADCS[2:0] = 0, ADCE[2:0] = 3

Converted in the following order: AIN0→AIN1→AIN2→AIN3

ADCS[2:0] = 4, ADCE[2:0] = 1

Converted in the following order: AIN4→AIN5→(AIN6)→(AIN7)→AIN0→AIN1

Note: The control circuits in the A/D converter supports up to eight channels for expansion in the future, and it performs A/D conversion if a channel (AIN6–AIN7) without an analog input is specified. In this case, the results that will be stored to ADD[15:0]/ADC10_ADD register is 0x0. To avoid A/D conversion for the channels without an input, set the ADCS[2:0] to equal or smaller than ADCE[2:0] within the available analog inputs.

25.3.3 A/D Conversion Mode Setting

The A/D converter provides two conversion modes that can be selected using ADMS/ADC10_TRG register: one-time conversion mode and continuous conversion mode.

1. One-time conversion mode (ADMS = 0)

The A/D converter performs A/D conversion for all analog inputs within the range from the start channel specified by ADCS[2:0]/ADC10_TRG register to the end channel specified by the ADCE[2:0]/ADC10_TRG register once and then stops automatically.

2. Continuous conversion mode (ADMS = 1)

The A/D converter repeatedly performs A/D conversion for the channels in the range specified by ADCS[2:0] and ADCE[2:0] until stopped with software.

At initial reset, the A/D converter is set to one-time conversion mode.

25.3.4 Trigger Selection

Select a trigger source to start A/D conversion from among the three types listed in Table 25.3.4.1 using ADTS[1:0]/ADC10_TRG register.

Table 25.3.4.1 Trigger Selection

ADTS[1:0]	Trigger source
0x3	External trigger (#ADTRIG)
0x2	Reserved
0x1	T8 Ch.2
0x0	Software trigger

(Default: 0x0)

1. External trigger (#ADTRIG)

The signal input to the #ADTRIG pin is used as a trigger. To use this trigger source, the I/O port pin must be configured for the #ADTRIG input using the port function select bit (see the “I/O Ports (GPIO)” chapter). An A/D conversion starts when a Low level of the #ADTRIG signal is detected.

Note: When using an external trigger to start A/D conversion, ensure to maintain the Low period of the trigger signal input to the #ADTRIG pin for two or more C33 PE Core operating clock cycles.

2. T8 Ch.2

The underflow signal of T8 Ch.2 is used as a trigger. Since T8 underflow cycle can be programmed with flexibility, this trigger source is effective when periodic A/D conversions are required. For more information on timer settings, see the “8-bit Timers (T8)” chapter.

3. Software trigger

Writing 1 to ADCTL/ADC10_CTL register with software serves as a trigger to start A/D conversion.

25.3.5 Sampling Time Setting

The analog signal input sampling time in this A/D converter is configured with ADST[2:0]/ADC10_TRG register. In the S1C33L26, do not alter ADST[2:0] from the default value (0x7).

Table 25.3.5.1 Sampling Time Settings

ADST[2:0]	Sampling time (in conversion clock cycles)
0x7	9 cycles
0x6	8 cycles
0x5	7 cycles
0x4	6 cycles
0x3	5 cycles
0x2	4 cycles
0x1	3 cycles
0x0	2 cycles

(Default: 0x7)

25.3.6 Setting Conversion Result Storing Mode

The A/D converter loads the 10-bit conversion results into ADD[15:0]/ADC10_ADD register (16-bit register) after an A/D conversion has completed. At this time, the 10-bit conversion results are aligned in the 16-bit register according to the conversion result storing mode set with STMD/ADC10_TRG register either as the high-order 10 bits (left justify mode) or the low-order 10 bits (right justify mode). The remaining six bits are all set to 0.

ADD bit	15	...	10	9	...	6	5	...	0		
Left justify mode (STMD = 1)	(MSB)	10-bit conversion results						(LSB)	0	...	0
Right justify mode (STMD = 0)	0	...	0	(MSB)	10-bit conversion results						(LSB)

Figure 25.3.6.1 Conversion Data Alignment

25.4 A/D Conversion Control and Operations

The A/D converter should be controlled in the sequence shown below.

1. Activate the A/D converter.
2. Start A/D conversion.
3. Read the A/D conversion results.
4. Terminate A/D conversion.

25.4.1 Activating A/D Converter

After the settings described in Section 25.3 have been completed, write 1 to ADEN/ADC10_CTL register to enable the A/D converter. The A/D converter is thereby ready to accept a trigger to start A/D conversion. To set up the A/D converter again, or when the A/D converter is not used, ADEN must be set to 0.

25.4.2 Starting A/D Conversion

The A/D converter starts A/D conversion when a trigger is input while ADEN is 1. When software trigger is selected, an A/D conversion starts by writing 1 to ADCTL/ADC10_CTL register.

The A/D converter accepts triggers from only the trigger source selected by ADTS[1:0]/ADC10_TRG register.

Once a trigger is input, the A/D converter starts sampling of the analog input signal and A/D conversion beginning with the conversion start channel selected by ADCS[2:0]/ADC10_TRG register.

The software trigger bit ADCTL functions as an A/D conversion status bit that goes 1 while A/D conversion is underway even if it has started by another trigger source. The channel in which conversion is underway can be identified by reading ADICH[2:0]/ADC10_CTL register.

25.4.3 Reading A/D Conversion Results

Upon completion of the A/D conversion in the start channel, the A/D converter loads the conversion results into ADD[15:0]/ADC10_ADD register and sets the conversion completion flag ADCF/ADC10_CTL register. If multiple channels are specified using ADCS[2:0]/ADC10_TRG register and ADCE[2:0]/ADC10_TRG register, the A/D converter continues A/D conversions in the subsequent channels.

The results of A/D conversion are stored in ADD[15:0] each time conversion in one channel is completed. At the same time, a conversion completion interrupt can be generated, enabling to read out the converted data. If no conversion completion interrupt is used, read the conversion results from ADD[15:0] after confirming that ADCF is set to 1 indicating completion of conversion. ADCF is reset to 0 when ADD[15:0] is read.

When a single channel or multiple channels are being converted continuously, the conversion results must be read out from ADD[15:0] before the following conversion has completed. If the A/D conversion currently underway is completed while ADCF is set to 1 (before reading the previous conversion results), ADD[15:0] is overwritten and the overwrite error flag ADOWE/ADC10_CTL register is set to 1. At this time, a conversion data overwrite error interrupt can be generated. After the conversion results are read from ADD[15:0], ADOWE should be read to check whether the read data is valid or not. Or enable conversion data overwrite error interrupts and perform error handling using the interrupt. Once ADOWE is set, it will not be reset until software writes 1. Since ADCF is also set simultaneously with ADOWE, read out the converted data to reset ADCF.

Note: Occurrence of an overwrite error does not stop continuous conversion.

25.4.4 Terminating A/D Conversion

One-time conversion mode (ADMS = 0)

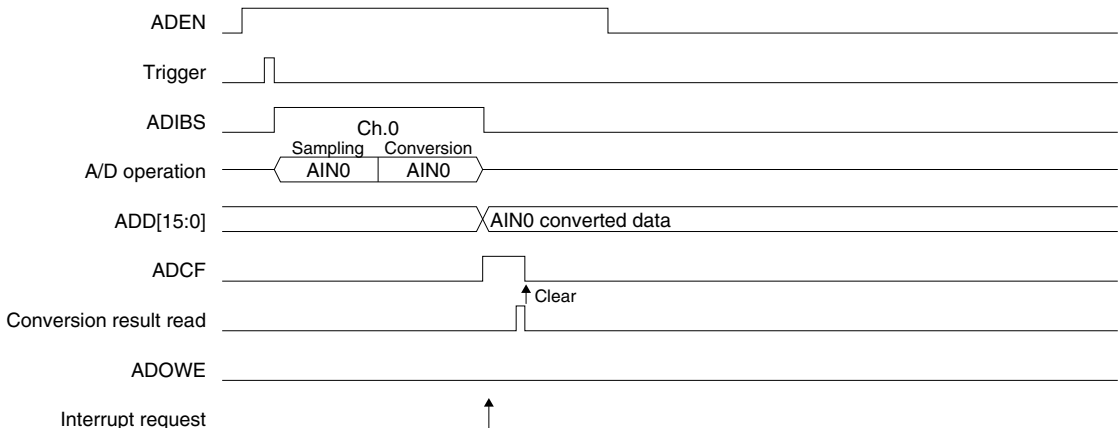
In one-time mode, the A/D converter performs A/D conversion within the channel range successively beginning with the conversion start channel specified by ADCS[2:0]/ADC10_TRG register and terminates once the conversion end channel specified by ADCE[2:0]/ADC10_TRG register has been completed. ADCTL/ADC10_CTL register is reset to 0 upon completion of the conversion sequence.

Continuous conversion mode (ADMS = 1)

In continuous conversion mode, the A/D converter repeatedly performs A/D conversion from the conversion start channel to the conversion end channel. The hardware does not stop the conversion sequence. To stop A/D conversion, write 0 to ADCTL. Since the conversion sequence is forcibly terminated, the results of the conversion then underway cannot be obtained.

25.4.5 Timing Charts

Figure 25.4.5.1 shows the operations of the A/D converter.



(1) Single channel (AIN0) one-time conversion mode (ADCS = 0, ADCE = 0, ADMS = 0)

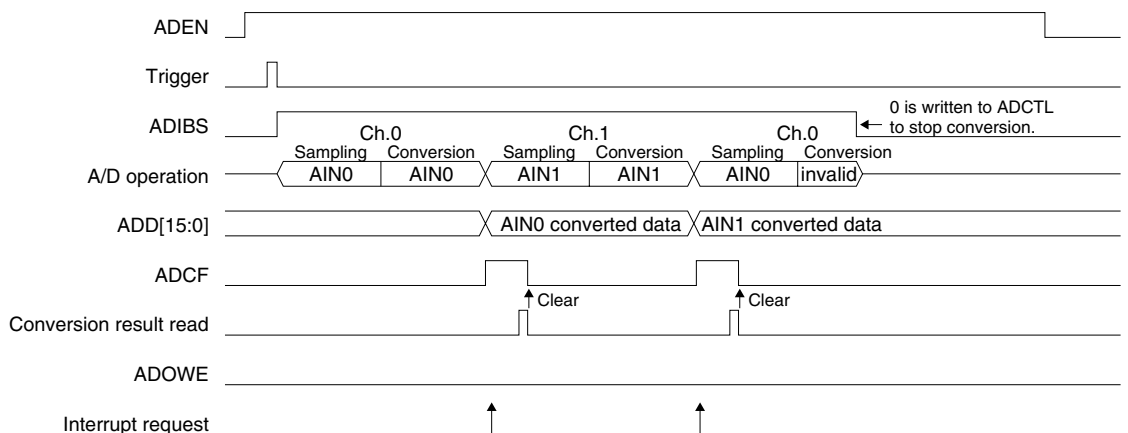
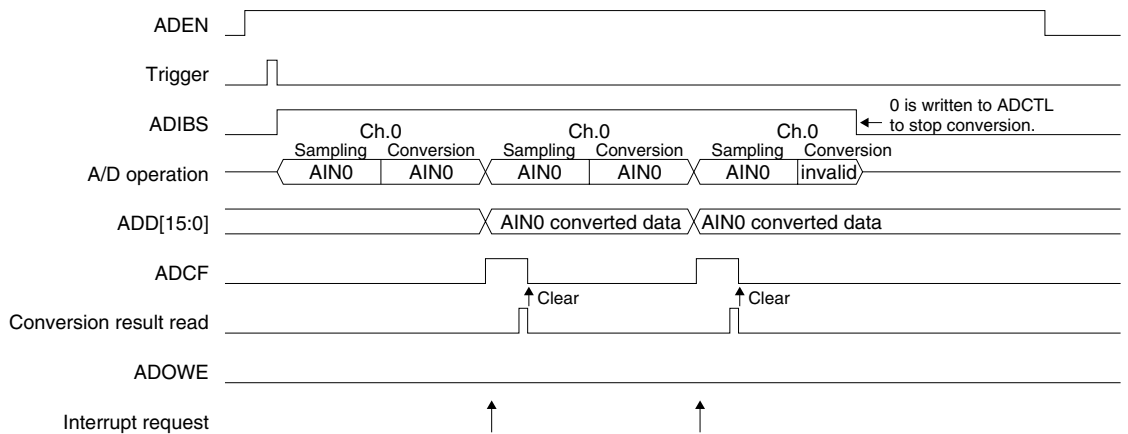
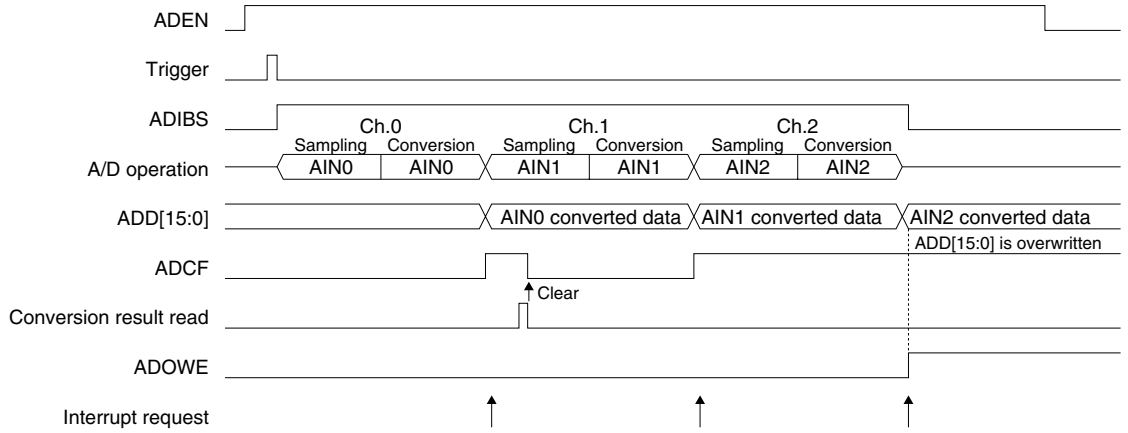


Figure 25.4.5.1 A/D Converter Operations

25.5 A/D Converter Interrupts and DMA

The A/D converter includes a function for generating the following two different types of interrupts.

- Conversion completion interrupt
- Conversion data overwrite error interrupt

The A/D converter outputs one interrupt signal shared by the two above interrupt causes to the interrupt controller (ITC). Inspect the status flag to determine the interrupt cause occurred.

Conversion completion interrupt

To use this interrupt, set ADCIE/ADC10_CTL register to 1. If ADCIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When A/D conversion in a channel has completed, the A/D converter sets ADCF/ADC10_CTL register to 1, indicating that the converted data can be read out. If conversion completion interrupts are enabled (ADCIE = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met.

You can inspect ADCF in the ADC10 interrupt handler routine to determine whether the ADC10 interrupt is attributable to a completion of conversion. If ADCF is 1, the converted data can be read out from ADD[15:0]/ADC10_ADD register by the interrupt handler routine. The interrupt cause ADCF is reset to 0 by reading ADD[15:0] and this interrupt will not be generated until the subsequent conversion has completed.

Conversion data overwrite error interrupt

To use this interrupt, set ADOIE/ADC10_CTL register to 1. If ADOIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If the following A/D conversion has completed when ADD[15:0] has not been read (ADCF = 1), the A/D converter sets ADOWE/ADC10_CTL register to 1, indicating that ADD[15:0] is overwritten. If conversion data overwrite error interrupts are enabled (ADOIE = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met.

You can inspect ADOWE in the ADC10 interrupt handler routine to determine whether the ADC10 interrupt is attributable to an overwrite error. If ADOWE is 1, perform error handling by the interrupt handler routine. The interrupt cause ADOWE is reset to 0 by writing 1.

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

DMA Transfer

The A/D converter can invoke a DMA when A/D conversion in a channel has completed. This allows continuous data transfer via the DMAC between memory and the conversion result register (ADC10_ADD). Two DMAC channels (Ch.0 and Ch.7) are available for A/D converter. For more information on DMA transfer, see the “DMA Controller (DMAC)” chapter.

25.6 Control Register Details

Table 25.6.1 List of A/D Converter Registers

Address	Register name		Function
0x301300	ADC10_ADD	A/D Conversion Result Register	A/D converted data
0x301302	ADC10_TRG	A/D Trigger/Channel Select Register	Sets start/end channels and conversion mode.
0x301304	ADC10_CTL	A/D Control/Status Register	Controls A/D converter and indicates conversion status.
0x301306	ADC10_CLK	A/D Clock Control Register	Controls A/D converter clock.

The A/D converter registers are described in detail below. These are 16-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

A/D Conversion Result Register (ADC10_ADD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Conversion Result Register (ADC10_ADD)	0x301300 (16 bits)	D15–0	ADD[15:0]	A/D converted data ADD[9:0] are effective when STMD = 0 (ADD[15:10] = 0) ADD[15:6] are effective when STMD = 1 (ADD[5:0] = 0)	0x0 to 0x3ff	0x0	R	

D[15:0] ADD[15:0]: A/D Converted Data Bits

The A/D conversion results are stored. (Default: 0x0)

The data alignment in this 16-bit register (conversion result storing mode) can be selected using the STMD/ADC10_TRG register.

	ADD bit	15	...	10	9	...	6	5	...	0	
Left justify mode (STMD = 1)	(MSB)	10-bit conversion results						(LSB)	0	...	0
Right justify mode (STMD = 0)	0	...	0	(MSB)	10-bit conversion results						(LSB)

Figure 25.6.1 Conversion Data Alignment

This register is a read-only, so writing to this register is ignored.

A/D Trigger/Channel Select Register (ADC10_TRG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
A/D Trigger/Channel Select Register (ADC10_TRG)	0x301302 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.	
		D13–11	ADCE[2:0]	End channel select	0x0 to 0x5	0x0	R/W		
		D10–8	ADCS[2:0]	Start channel select	0x0 to 0x5	0x0	R/W		
		D7	STMD	Conversion result storing mode	1 ADD[15:6] 0 ADD[9:0]	0	R/W		
		D6	ADMS	Conversion mode select	1 Continuous 0 Single	0	R/W		
		D5–4	ADTS[1:0]	Conversion trigger select	ADTS[1:0] Trigger	0x0	R/W		
						0x3 #ADTRIG pin reserved 0x2 reserved 0x1 T8 Ch.2 0x0 Software			
		D3	–	reserved	–	–	–	–	0 when being read.
		D2–0	ADST[2:0]	Sampling time setting	ADST[2:0] Sampling time	0x7 9•ADCCLK 0x6 8•ADCCLK 0x5 7•ADCCLK 0x4 6•ADCCLK 0x3 5•ADCCLK 0x2 4•ADCCLK 0x1 3•ADCCLK 0x0 2•ADCCLK	0x7	R/W	Always set to 0x7.

D[15:14] Reserved

D[13:11] ADCE[2:0]: End Channel Select Bits

Sets the conversion end channel with a channel number from 0 to 5. (Default: 0x0 = AIN0)

Analog inputs can be A/D-converted continuously from the channel set by ADCS[2:0] to the channel set by ADCE[2:0] in one A/D conversion. If only one channel is to be A/D converted, set the same channel number in both ADCS[2:0] and ADCE[2:0].

Table 25.6.2 Relationship between ADCS/ADCE and Input Channels

ADCS[2:0]/ADCE[2:0]	Channel selected
0x7–0x6	Reserved
0x5	AIN5
0x4	AIN4
0x3	AIN3
0x2	AIN2
0x1	AIN1
0x0	AIN0

(Default: 0x0)

D[10:8] ADCS[2:0]: Start Channel Select Bits

Sets the conversion start channel with a channel number from 0 to 5. (Default: 0x0 = AIN0)

D7 STMD: Conversion Result Storing Mode Bit

Selects the data alignment when the conversion results are loaded into ADD[15:0].

- 1 (R/W): Left justify mode (10-bit conversion results → ADD[15:6], ADD[5:0] = 0)
- 0 (R/W): Right justify mode (10-bit conversion results → ADD[9:0], ADD[15:10] = 0) (default)

D6 ADMS: Conversion Mode Select Bit

Selects an A/D conversion mode.

- 1 (R/W): Continuous conversion mode
- 0 (R/W): One-time conversion mode (default)

Writing 1 to ADMS sets the A/D converter to continuous conversion mode. In this mode, A/D conversions in the range of the channels selected by ADCS[2:0] and ADCE[2:0] are executed continuously until stopped with software.

When ADMS is 0, the A/D converter operates in one-time conversion mode. In this mode, A/D conversion is terminated after all inputs in the range of the channels selected by ADCS[2:0] and ADCE[2:0] have been converted once.

D[5:4] ADTS[1:0]: Conversion Trigger Select Bits

Selects a trigger source to start A/D conversion.

Table 25.6.3 Trigger Selection

ADTS[1:0]	Trigger source
0x3	External trigger (#ADTRIG)
0x2	Reserved
0x1	T8 Ch.2
0x0	Software trigger

(Default: 0x0)

When an external trigger is used, the #ADTRIG pin must be configured in advance using the port function select bit (see the “I/O Ports (GPIO)” chapter). A/D conversion is started when the #ADTRIG signal goes Low. When T8 Ch.2 is used, since its underflow signal serves as a trigger, set the underflow cycle and other conditions for the timer.

D3 Reserved

D[2:0] ADST[2:0]: Sampling Time Setting Bits

Sets the analog input sampling time.

Table 25.6.4 Sampling Time Settings

ADST[2:0]	Sampling time (in conversion clock cycles)
0x7	9 cycles
0x6	8 cycles
0x5	7 cycles
0x4	6 cycles
0x3	5 cycles
0x2	4 cycles
0x1	3 cycles
0x0	2 cycles

(Default: 0x7)

Note: Do not alter ADST[2:0] from the default value (0x7).

A/D Control/Status Register (ADC10_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
A/D Control/ Status Register (ADC10_CTL)	0x301304 (16 bits)	D15	–	reserved	–	–	–	0 when being read.	
		D14–12	ADICH[2:0]	Conversion channel indicator	0x0 to 0x5	0x0	R		
		D11	–	reserved	–	–	–	0 when being read.	
		D10	ADIBS	ADC10 status	1 Busy 0 Idle	0	0	R	
		D9	ADOWE	Overwrite error flag	1 Error 0 Normal	0	0	R/W	Reset by writing 1.
		D8	ADCF	Conversion completion flag	1 Completed 0 Run/Standby	0	0	R	Reset when ADC10_ADD is read.
		D7–6	–	reserved	–	–	–	–	0 when being read.
		D5	ADOIE	Overwrite error interrupt enable	1 Enable 0 Disable	0	0	R/W	
		D4	ADCIE	Conversion completion int. enable	1 Enable 0 Disable	0	0	R/W	
		D3–2	–	reserved	–	–	–	–	0 when being read.
		D1	ADCTL	A/D conversion control	1 Start 0 Stop	0	0	R/W	
		D0	ADEN	ADC10 enable	1 Enable 0 Disable	0	0	R/W	

D15 Reserved

D[14:12] ADICH[2:0]: Conversion Channel Indicator Bits

Indicates the channel number (0 to 5) currently being A/D-converted. (Default: 0x0 = AIN0)

When A/D conversion is performed in multiple channels, read this bit to identify the channel in which conversion is underway.

D11 Reserved

D10 ADIBS: ADC10 Status Bit

Indicates the A/D converter status.

1 (R): Being converted

0 (R): Conversion completed/standby (default)

ADIBS is set to 1 at the input trigger signal edge (at the beginning of sampling) and is reset to 0 upon completion of conversion (when ADCTL is set to 0).

D9 ADOWE: Overwrite Error Flag Bit

Indicates that the converted results in ADD[15:0]/ADC10_ADD register have been overwritten before reading.

1 (R): Overwrite error (cause of interrupt has occurred)

0 (R): Normal (cause of interrupt has not occurred) (default)

1 (W): Flag is reset

0 (W): Ignored

When a single channel or multiple channels are being converted continuously, ADD[15:0] is overwritten and ADOWE is set to 1 if the A/D conversion currently underway is completed while ADCF is set to 1 (before reading the previous conversion results). After the conversion results are read from ADD[15:0], ADOWE should be read to check whether the read data is valid or not.

ADOWE is a cause of ADC10 interrupt. When ADOWE is set to 1, a conversion data overwrite error interrupt request is output to the ITC if ADOIE has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied. ADOWE is reset by writing 1.

D8 ADCF: Conversion Completion Flag Bit

Indicates that A/D conversion has been completed.

1 (R): Conversion completed (cause of interrupt has occurred)

0 (R): Being converted/standby (cause of interrupt has not occurred) (default)

ADCF is set to 1 when A/D conversion is completed, and the converted data is loaded into ADD[15:0]/ADC10_ADD register.

ADCF is a cause of ADC10 interrupt. When ADCF is set to 1, a conversion completion interrupt request is output to the ITC if ADCIE has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied. ADCF is reset to 0 by reading ADD[15:0].

An overwrite error occurs if the next A/D conversion is completed while ADCF is set (see ADOWE above), ADCF must be reset by reading ADD[15:0] before an overwrite occurs. When an overwrite error occurs, ADCF is also set due to completion of conversion.

D[7:6] Reserved**D5 ADOIE: Overwrite Error Interrupt Enable Bit**

Enables or disables interrupts caused by occurrences of conversion data overwrite errors.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting ADOIE to 1 enables conversion data overwrite error interrupt requests to the ITC; setting to 0 disables interrupts.

D4 ADCIE: Conversion Completion Interrupt Enable Bit

Enables or disables interrupts caused by completion of conversion.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting ADCIE to 1 enables conversion data overwrite error interrupt requests to the ITC; setting to 0 disables interrupts.

D[3:2] Reserved**D1 ADCTL: A/D Conversion Control Bit**

Controls A/D conversion.

1 (W): Software trigger

0 (W): Stop A/D conversion

1 (R): Being converted

0 (R): Conversion completed/standby (default)

Write 1 to ADCTL to start A/D conversion by a software trigger. If any other trigger is used, ADCTL is automatically set to 1 by the hardware.

ADCTL remains set while A/D conversion is underway. In one-time conversion mode, upon completion of A/D conversion in the specified channels, ADCTL is reset to 0 and the A/D conversion circuit stops operating. To stop A/D conversion during operation in continuous conversion mode, reset ADCTL by writing 0.

When ADEN is 0 (A/D conversion disabled), ADCTL is fixed to 0, with no trigger accepted.

D0 ADEN: ADC10 Enable Bit

Enables or disables the A/D converter operations.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Writing 1 to ADEN enables the A/D converter, meaning it is ready to start A/D conversion (i.e., ready to accept a trigger).

When ADEN is 0, the A/D converter is disabled, meaning it is unable to accept a trigger.

Before setting the modes, start/end channels, or other A/D converter conditions, be sure to reset ADEN to 0. This helps to prevent the A/D converter from operating erratically.

- Notes:**
- To prevent interrupt recurrences, ADCF/ADC10_CTL register and ADOWE/ADC10_CTL register must be reset in the interrupt handler routine after an ADC10 interrupt has occurred.
 - To prevent unwanted interrupts, reset ADCF and ADOWE before enabling interrupts with ADCIE/ADC10_CTL register and ADOIE/ADC10_CTL register.

A/D Clock Control Register (ADC10_CLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
A/D Clock Control Register (ADC10_CLK)	0x301306 (16 bits)	D15-4	–	reserved	–	–	–	0 when being read.	
		D3-0	ADDF[3:0]	A/D converter clock division ratio select	ADDF[3:0] Division ratio	0x0	R/W	Source clock = PCLK1	
					0xf	reserved			
					0xe	1/32768			
					0xd	1/16384			
					0xc	1/8192			
					0xb	1/4096			
					0xa	1/2048			
					0x9	1/1024			
					0x8	1/512			
					0x7	1/256			
					0x6	1/128			
					0x5	1/64			
					0x4	1/32			
					0x3	1/16			
					0x2	1/8			
			0x1	1/4					
			0x0	1/2					

D[15:4] Reserved

D[3:0] ADDF[3:0]: A/D Converter Clock Division Ratio Select Bits

Selects the A/D converter clock (PCLK1 division ratio).

Table 25.6.5 A/D Conversion Clock (PCLK1 Division Ratio) Selections

ADDF[3:0]	Division Ratio
0xf	Reserved
0xe	1/32768
0xd	1/16384
0xc	1/8192
0xb	1/4096
0xa	1/2048
0x9	1/1024
0x8	1/512
0x7	1/256
0x6	1/128
0x5	1/64
0x4	1/32
0x3	1/16
0x2	1/8
0x1	1/4
0x0	1/2

(Default: 0x0)

Note: The A/D converter uses the prescaler output as the source clock, the prescaler must be run in advance.

26 LCD Controller (LCDC)

26.1 LCDC Module Overview

The S1C33L26 has a built-in LCD controller (LCDC) that supports 4/8-bit monochrome and color LCD panels, and 12/16/24-bit Generic HR-TFT panels. Also the S1C33L26 contains a 20K-byte VRAM (IVRAM) allowing a 320×240 -dot 4-color/gray scale screen (2-bpp mode) to be displayed. Furthermore, the bus controller (SRAMC, SDRAMC) allows the LCDC to access the external SDRAM/SRAM as a VRAM, thus a 320×240 -dot screen with 24-bpp color depth (16M colors) 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 interface is implemented. This method allows the LCDC to access SDRAM/SRAM (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 20K-byte internal VRAM (IVRAM) is mapped at addresses 0x90000 to 0x94fff.
- The external VRAM map is configurable.
- The frame interrupt signal is output to 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-, 16- or 24-bit Generic HR-TFT interface
- Typical resolutions
 - 320 × 240 (24-bpp mode, external VRAM is required) bpp = bits per pixel
 - 320 × 240 (2-bpp mode)
 * Note that the panel width must be a multiple of (16 ÷ bpp).

Display modes

- Due to frame rate modulation, gray scale 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 4K colors can be simultaneously displayed on a color passive LCD 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- and 16-bpp mode
- A maximum of 16M 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
 - 64K-color display in 16-bpp mode
 - 16M-color display in 24-bpp mode

Display features

- Picture-in-Picture Plus to display a variable size window overlaid over background image.
- Virtual display function to handle images with a different resolution from the LCD panel (any area in the virtual screen can be displayed on the LCD.)

Clock

- LCLK for the LCD controller is generated in the CMU by dividing the OSC3 clock by 1 to 32.
- Different clock paths are provided for the AHB bus interface (for accessing the VRAM), SAPB interface (for accessing the control registers), and LCLK, 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
- Blank display

26.2 Block Diagram

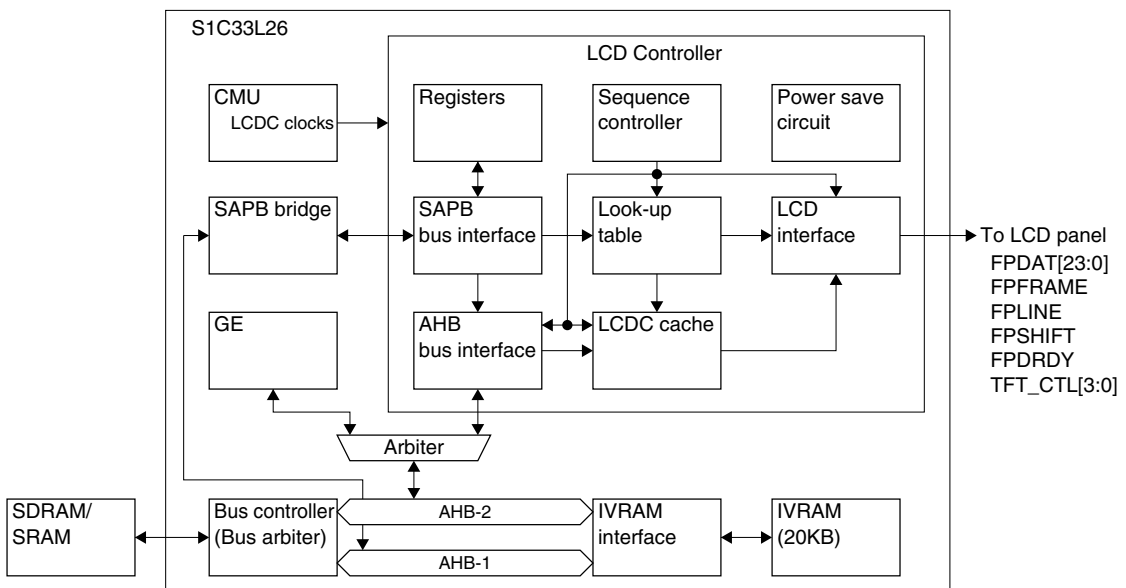


Figure 26.2.1 LCD Controller Block Diagram

SAPB bus interface

The C33 PE Core accesses the LCDC registers and monochrome 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 color look-up table. It also generates display data memory addresses for refreshing display.

Look-up table

In color mode, the LCDC uses LUTRAM (switched from DSTRAM via software) as the color look-up table (LUT). This consists of three tables (256×5 bits for red, 256×6 bits for green, and 256×5 bits for blue) and is used to set up color data to be displayed.

In monochrome mode, a 16×4 -bit table included in the LCDC is used as the look-up table to set up gray scale data to be displayed.

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.

26.3 LCDC Output Pins

Table 26.3.1 lists the output pins of the LCD controller. Table 26.3.2 shows the pin configurations classified by type of LCD panel.

Table 26.3.1 LCDC Output Pins

Pin name	I/O	Qty	Function
FPDAT[23:0]	O	24	LCD display data outputs
FPFRAME	O	1	LCD frame clock output
FPLINE	O	1	LCD line clock output
FPSHIFT	O	1	LCD shift clock output
FPDRDY	O	1	LCD DRDY/MOD signal output
TFT_CTL[3:0]	O	4	TFT interface control signal outputs

The LCDC output pins are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as LCDC output pins. For detailed information on pin function switching, see the “I/O Ports (GPIO)” chapter.

Table 26.3.2 Pin Configurations by LCD Panel Type

Pin name	Monochrome passive panel		Color passive panel			TFT panel
	4-bit data width	8-bit data width	4-bit data width	8-bit data width format 1	8-bit data width format 2	— *2
FPFRAME	FPFRAME					SPS
FPLINE	FPLINE					LP
FPSHIFT	FPSHIFT					DCLK
FPDRDY	MOD			FPSHIFT2	MOD	DEN
FPDAT0	— *1	D0	— *1	D0	D0	B3
FPDAT1	— *1	D1	— *1	D1	D1	B4
FPDAT2	— *1	D2	— *1	D2	D2	B5
FPDAT3	— *1	D3	— *1	D3	D3	B6
FPDAT4	D0	D4	D0	D4	D4	B7
FPDAT5	D1	D5	D1	D5	D5	G2
FPDAT6	D2	D6	D2	D6	D6	G3
FPDAT7	D3	D7	D3	D7	D7	G4
FPDAT8			— *1			G5
FPDAT9			— *1			G6
FPDAT10			— *1			G7
FPDAT11			— *1			R3
FPDAT12			— *1			R4
FPDAT13			— *1			R5
FPDAT14			— *1			R6
FPDAT15			— *1			R7
FPDAT16			— *1			B0
FPDAT17			— *1			B1
FPDAT18			— *1			B2
FPDAT19			— *1			G0
FPDAT20			— *1			G1
FPDAT21			— *1			R0
FPDAT22			— *1			R1
FPDAT23			— *1			R2

26 LCD CONTROLLER (LCDC)

Pin name	Monochrome passive panel		Color passive panel			TFT panel
	4-bit data width	8-bit data width	4-bit data width	8-bit data width format 1	8-bit data width format 2	— *2
TFT_CTL0			— *1			PS
TFT_CTL1			— *1			CLS
TFT_CTL2			— *1			REV
TFT_CTL3			— *1			SPL

*1 These pins can be used for other peripheral functions.

*2 Since the LCDC supports maximum 24-bit data width TFT panels, no LCD panel data width configuration is required.

When using a 12-bit data width TFT panel, connect it to the R7–R4, G7–G4, and B7–B4 pins.

When using a 16-bit data width TFT panel, connect it to the R7–R3, G7–G2, and B7–B3 pins.

When using a 18-bit data width TFT panel, connect it to the R7–R2, G7–G2, and B7–B2 pins.

Table 26.3.3 LCD Panel Configurations

LCD Panel	PANSELSEL *	COLOR *	DWD[1:0] *	BPP[2:0] *
Monochrome passive panel	0 (STN)	0 (Mono)	0 (4-bit data width) 1 (8-bit data width) 3 (8-bit data width)	0 (1 bpp) 1 (2 bpp) 2 (4 bpp)
Color passive panel	0 (STN)	1 (Color)	0 (4-bit data width) 1 (8-bit data width format 1) 3 (8-bit data width format 2)	0 (1 bpp) 1 (2 bpp) 2 (4 bpp) 3 (8 bpp) 4 (12 bpp) 5 (16 bpp)
TFT panel	1 (TFT)	1 (Color)	—	0 (1 bpp) 1 (2 bpp) 2 (4 bpp) 3 (8 bpp) 4 (12 bpp) 5 (16 bpp) 6 (24 bpp)

* The PANSELSEL, COLOR, DWD[1:0], and BPP[2:0] control bits are assigned in the LCDC_DISPMOD register.

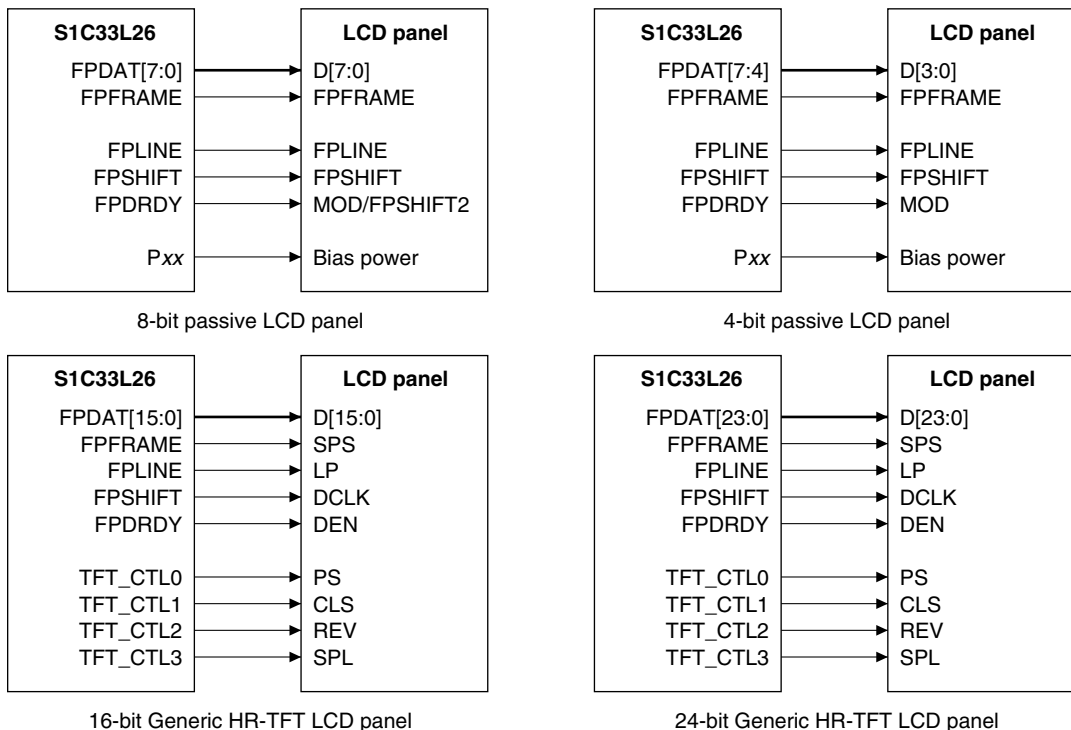


Figure 26.3.1 Typical LCD-Panel Connections

26.4 System Settings

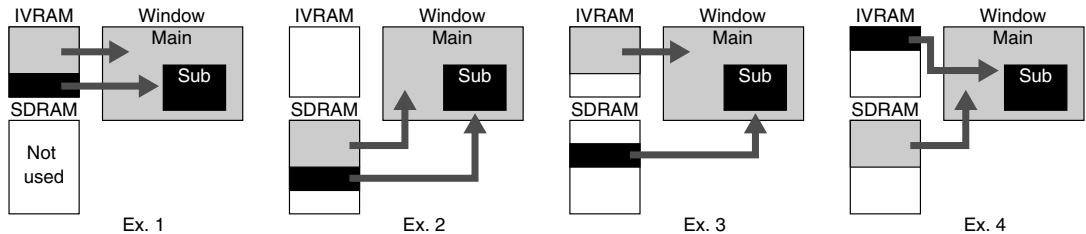
26.4.1 Configuration of Display Data Memory (VRAM)

The S1C33L26 has a built-in 20K-byte display data memory (IVRAM). This memory allows selection whether it is used as a VRAM by locating at 0x90000 to 0x94fff in Area 3 or a general-purpose RAM by locating in Area 0. Setting IVRAM_LOE/MISC_IRAM_LOE register to 0 configures the IVRAM as a general-purpose RAM in Area 0; setting to 1 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.

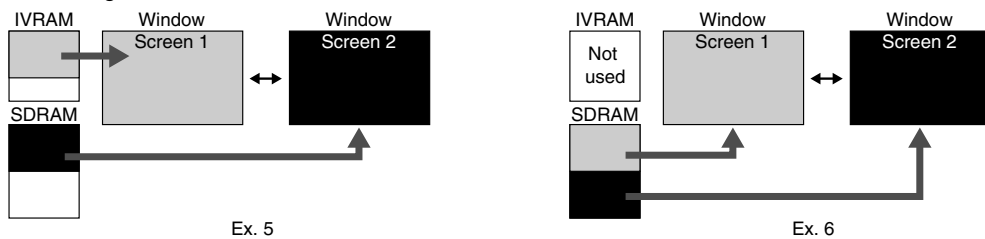
The LCDC can use an external SDRAM/SRAM as a VRAM in addition to IVRAM (the external SDRAM/SRAM can store general-purpose data as well as display data). There is no special configuration procedure for use of the external VRAM. Furthermore, both the external SDRAM/SRAM and IVRAM can be used as a VRAM 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 26.4.1.1 shows memory usage examples.

When using Picture-in-Picture Plus



When switching screens



When using virtual screen area

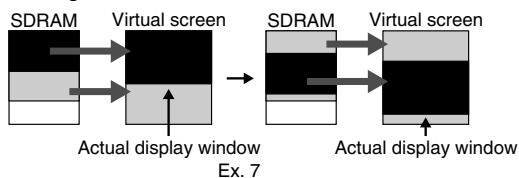


Figure 26.4.1.1 Memory Usage Examples

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, 24)

For example, a 320 × 240-pixel screen displayed in 64K colors (16-bpp mode) needs a 38,400 words (150K bytes) of memory area.

26.4.2 Setting the LCDC Clock

The LCDC operates with the LCLK, BCLK, and PCLK2 clocks supplied from the CMU. For controlling the clocks, see the “Clock Management Unit (CMU)” chapter.

LCLK

This is the LCDC operating clock (pixel clock) generated by dividing the OSC3 clock. The frequency divider generates 32 kinds of clocks from OSC3/1 to OSC3/32. Select a division ratio according to the frame rate using LCLKDIV[4:0]/CMU_LCLKDIV register.

$$\text{Frame rate} = \frac{f_{\text{LCLK}}}{\text{HT} \times \text{VT}} \text{ [Hz]}$$

f_{LCLK} : LCLK frequency

HT: Horizontal total period (horizontal panel size + horizontal non-display period) [pixels]

VT: Vertical total period (vertical panel size + vertical non-display period) [lines]

Table 26.4.2.1 LCDC Clock (OSC3 Division Ratio) Selections

LCLKDIV[4:0]	Division ratio (OSC3/n)	LCLKDIV[4:0]	Division ratio (OSC3/n)
0x1f	1/32	0xf	1/16
0x1e	1/31	0xe	1/15
0x1d	1/30	0xd	1/14
0x1c	1/29	0xc	1/13
0x1b	1/28	0xb	1/12
0x1a	1/27	0xa	1/11
0x19	1/26	0x9	1/10
0x18	1/25	0x8	1/9
0x17	1/24	0x7	1/8
0x16	1/23	0x6	1/7
0x15	1/22	0x5	1/6
0x14	1/21	0x4	1/5
0x13	1/20	0x3	1/4
0x12	1/19	0x2	1/3
0x11	1/18	0x1	1/2
0x10	1/17	0x0	1/1

(Default: 0x7)

LCLK_EN/CMU_CLKCTL register is used for clock supply control (default: off). Before using the LCDC, set LCLK_EN to 1.

Note: Disable LCLK supply (LCLK_EN = 0) when changing the clock division ratio using LCLKDIV[4:0] or before executing the slp instruction.

BCLK

This clock is required for the LCDC to access the VRAM. BCLK can be stopped in HALT mode using BCLK_EN/CMU_CLKCTL register.

PCLK2

This clock is required for accessing the LCDC registers. PCLK2 can be stopped using PCLK2_EN/CMU_CLKCTL register.

26.5 Setting the LCD Panel

26.5.1 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-, 16- or 24-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 PANELSEL/LCDC_DISPMD register to select the type of LCD panel, either STN or HR-TFT.

PANELSEL = 1: Generic HR-TFT panel selected

PANELSEL = 0: STN panel selected (default)

Selecting between color and monochrome

Use COLOR/LCDC_DISPMD register to select the type of LCD panel, either color or monochrome.

COLOR = 1: Color panel selected

COLOR = 0: Monochrome panel selected (default)

Selecting the STN panel data width

Use DWD[1:0]/LCDC_DISPMD register to select the data width and format for STN panels.

Table 26.5.1.1 Data Width Selection of STN Panels

COLOR	DWD[1:0]	LCD panel
1	0x3	Color single 8-bit passive LCD panel (format 2)
	0x2	Reserved
	0x1	Color single 8-bit passive LCD panel (format 1)
	0x0	Color single 4-bit passive LCD panel
0	0x3	Monochrome single 8-bit passive LCD panel
	0x2	Reserved
	0x1	Monochrome single 8-bit passive LCD panel
	0x0	Monochrome single 4-bit passive LCD panel

The data width for a generic HR-TFT LCD panel is determined according to the display (bpp) mode selected.

26.5.2 STN Panel Timing Parameters

The STN panel timing parameters shown in Figures below can be set using the LCDC control registers.

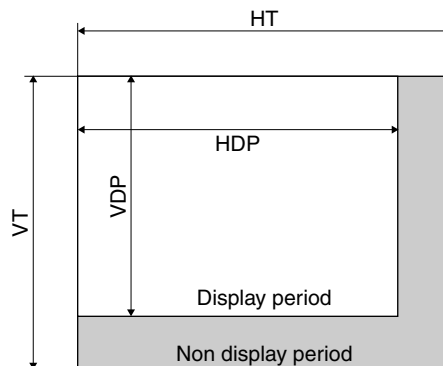
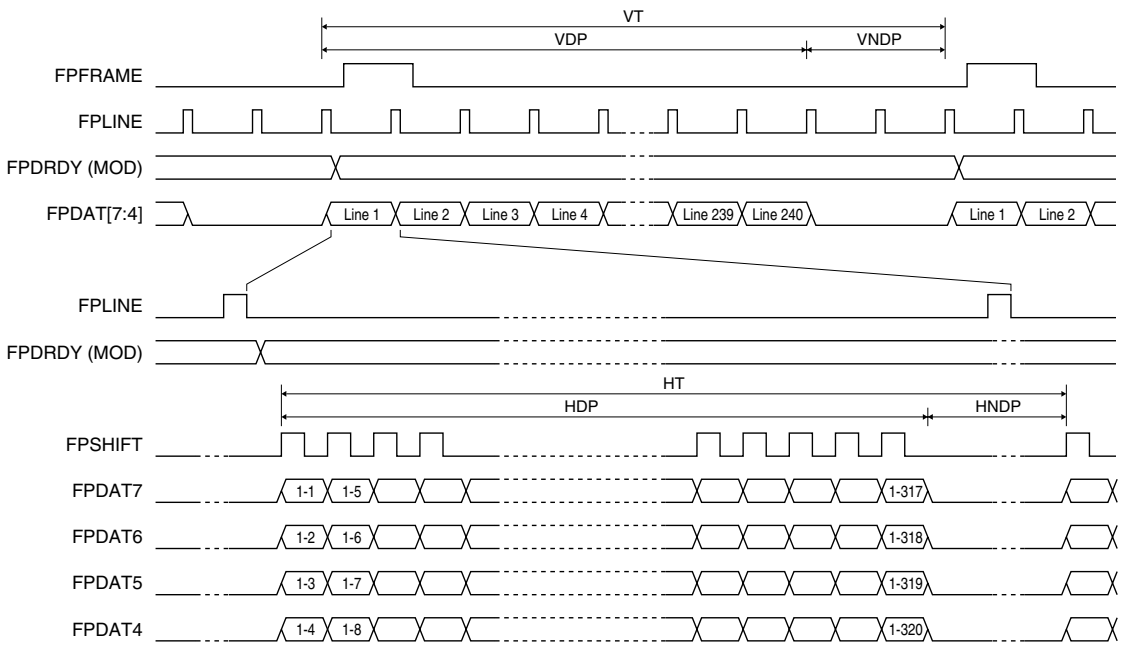


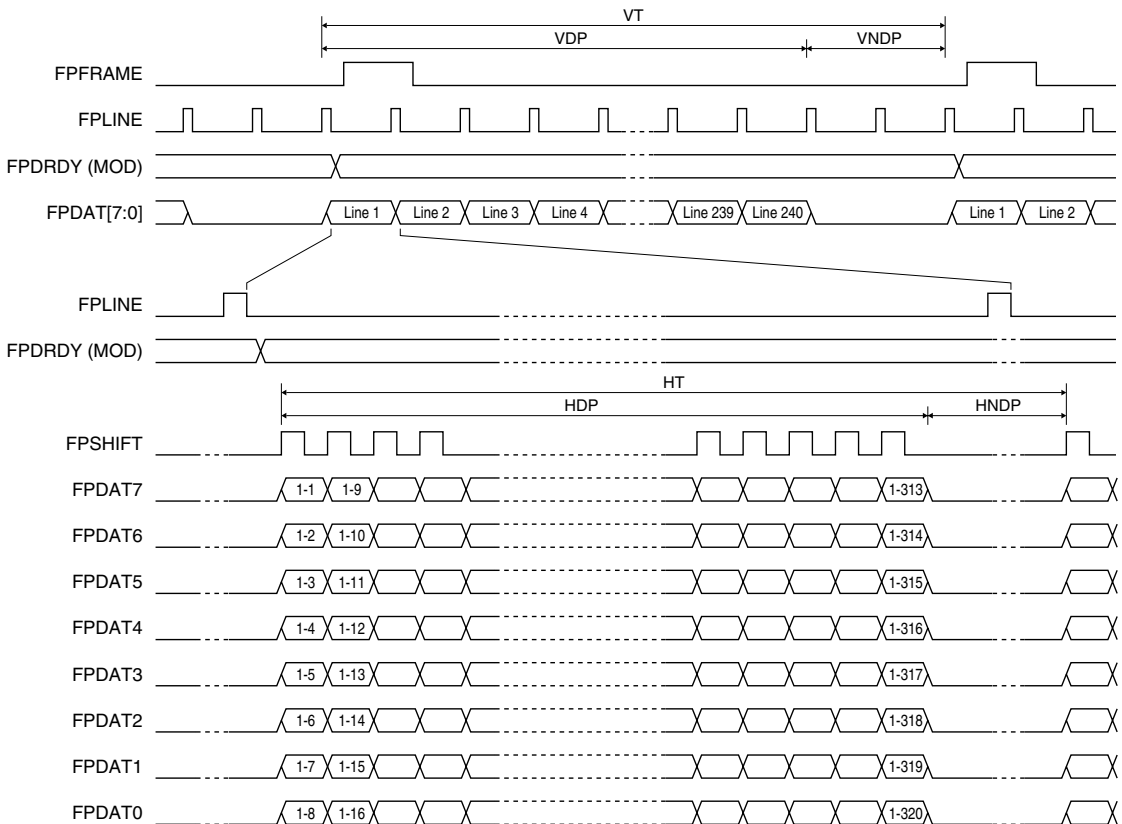
Figure 26.5.2.1 STN Panel Timing Parameters

26 LCD CONTROLLER (LCDC)



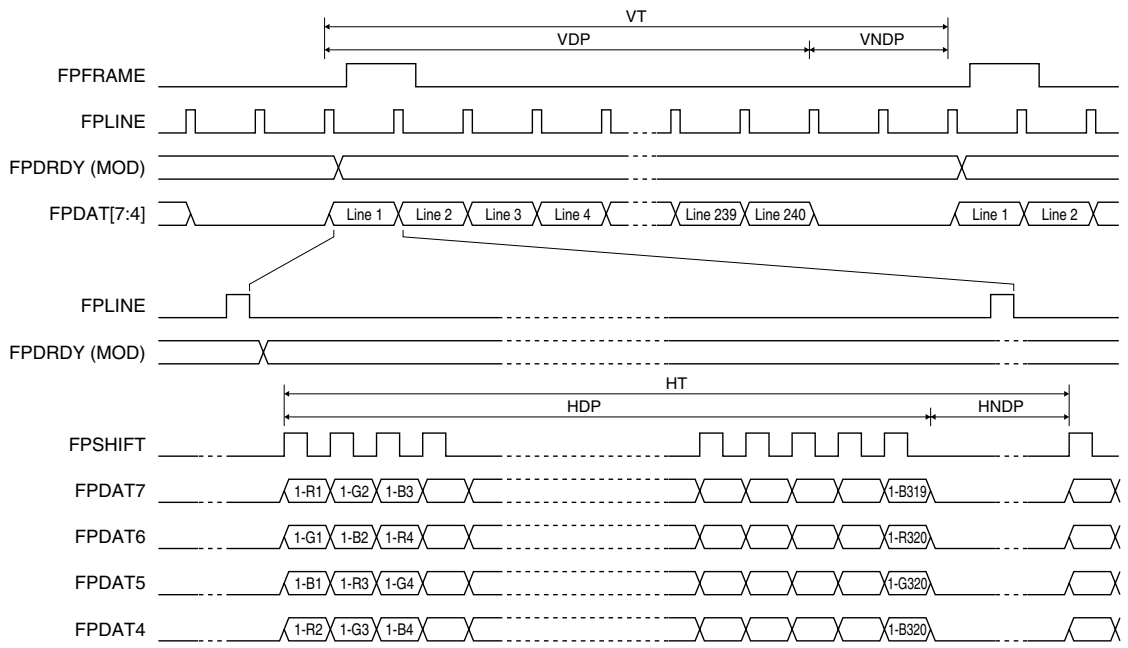
* Diagram drawn with 2 FPLINE vertical blank period
 Example timing for a 320 × 240 panel
 For this timing diagram FPSMASK is set to 1

Figure 26.5.2.2 4-bit Single Monochrome Panel Timing Chart (Example)



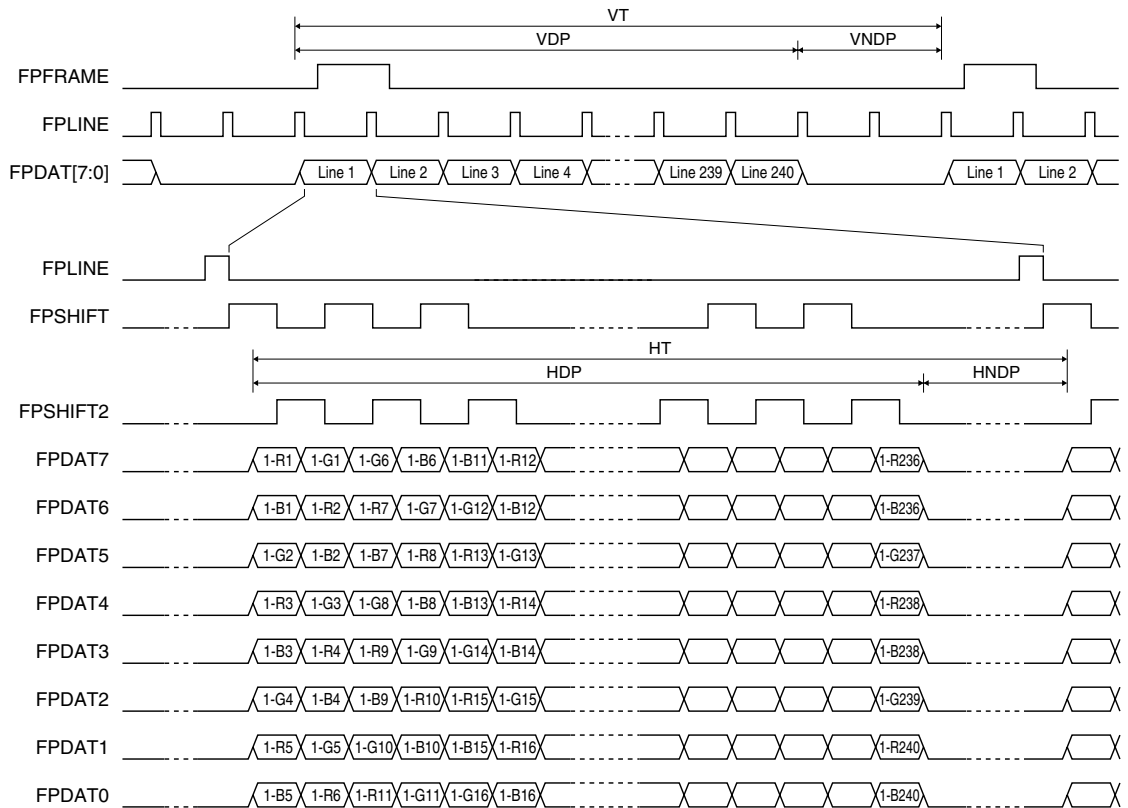
* Diagram drawn with 2 FPLINE vertical blank period
 Example timing for a 320 × 240 panel
 For this timing diagram FPSMASK is set to 1

Figure 26.5.2.3 8-bit Single Monochrome Panel Timing Chart (Example)



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

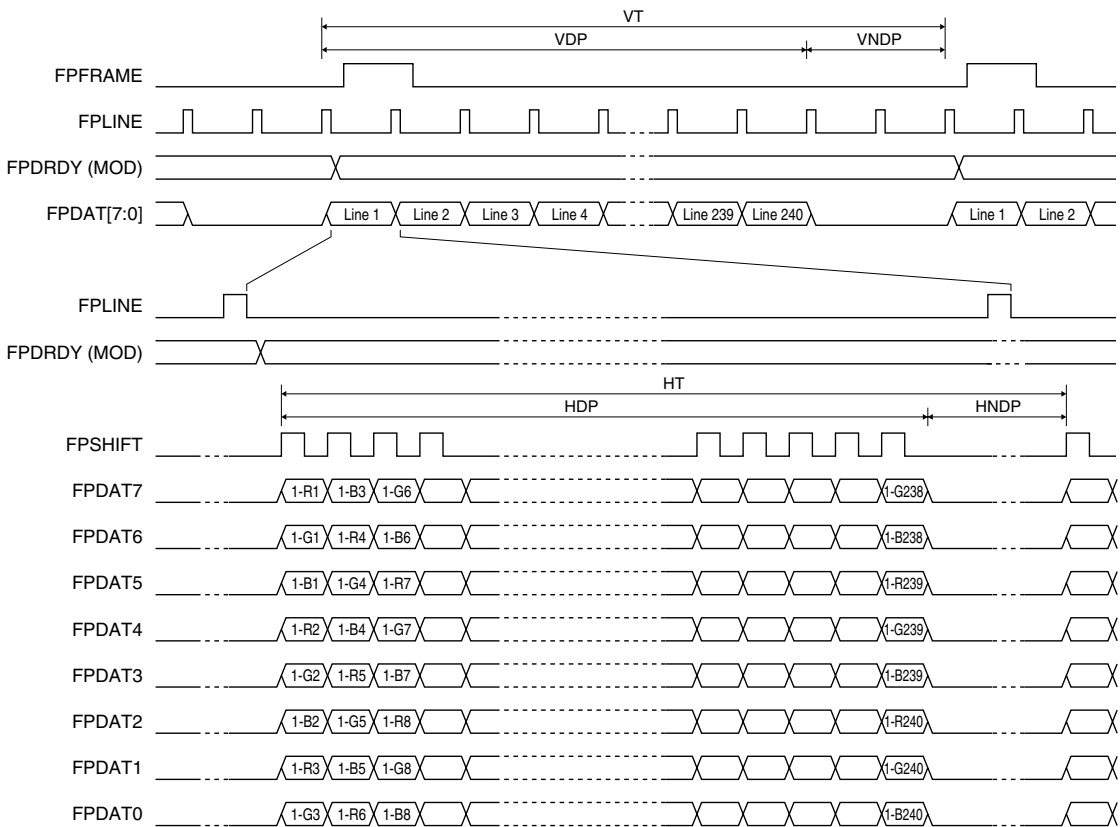
Figure 26.5.2.4 4-bit Single Color Panel Timing Chart (Example)



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

Figure 26.5.2.5 8-bit Single Color Panel (Format 1) Timing Chart (Example)

26 LCD CONTROLLER (LCDC)



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

Figure 26.5.2.6 8-bit Single Color Panel (Format 2) Timing Chart (Example)

HT: Horizontal total period

Use HTCNT[6:0]/LCDC_HDISP register to set the horizontal total period.

$$HT = (HTCNT[6:0] + 1) \times 8 [Ts] \quad Ts: \text{Pixel clock (LCLK) period}$$

HTCNT[6:0] must be programmed such that the following condition is met:

$$HTCNT[6:0] \geq HDPCNT[6:0] + 3$$

Note: HT should be determined so that the horizontal non-display period (HNDP = HT - HDP) will be longer than the time required when the LCDC accesses eight words in the VRAM.

HDP: Horizontal display period

Use HDPCNT[6:0]/LCDC_HDISP register to set the horizontal display period (= horizontal panel resolution).

$$HDP = (HDPCNT[6:0] + 1) \times 8 [Ts]$$

HDPCNT[6:0] 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]/LCDC_VDISP register to set the vertical total period.

$$VT = VTCNT[9:0] + 1 [\text{lines}]$$

VDP: Vertical display period

Use VDPCNT[9:0]/LCDC_VDISP register to set the vertical display period (= vertical panel resolution).

$$VDP = VDPCNT[9:0] + 1 [\text{lines}]$$

VDPCNT[9:0] 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/LCDC_DISPMD register = 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 Figures 26.5.2.2 to 26.5.2.6.

When monochrome passive panel is selected (COLOR = 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 FPSHIFT_MSK/LCDC_DISPMD register to 1.

Note: When using an STN panel, the registers for setting the HR-TFT timing parameters must be set to 0x0.

26.5.3 HR-TFT Panel Timing Parameters

The HR-TFT panel timing parameters shown in Figures below can be set using the LCDC control registers.

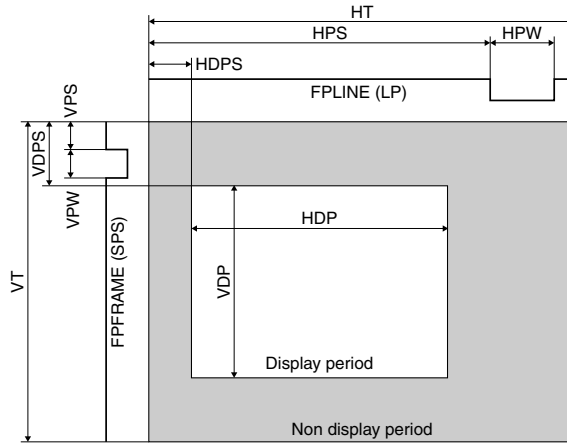
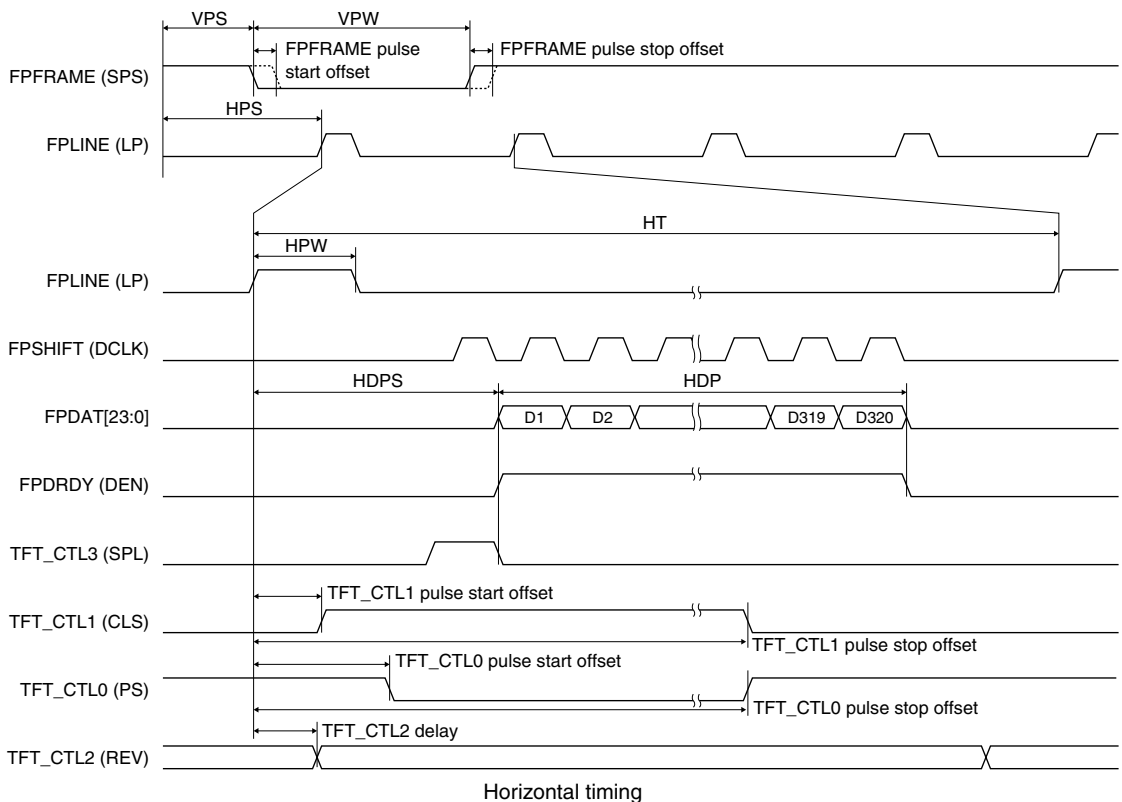


Figure 26.5.3.1 HR-TFT Panel Timing Parameters



Horizontal timing

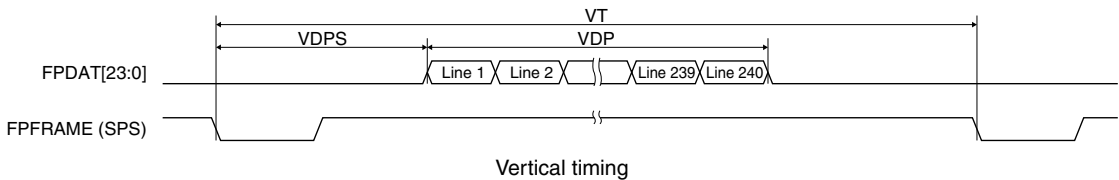


Figure 26.5.3.2 HR-TFT Panel Timing Chart

HT: Horizontal total period

Use HTCNT[6:0]/LCDC_HDISP register to set the horizontal total period.

$$HT = (HTCNT[6:0] + 1) \times 8 [Ts] \quad (Ts: \text{Pixel clock period})$$

HTCNT[6:0] must be programmed such that the following conditions are met:

$$HTCNT[6:0] \geq HDPCNT[6:0] + 3$$

$$HT > HDP + HDPS$$

Note: HT should be determined so that the horizontal non-display period (HNDP = HT - HDP) will be longer than the time required when the LCDC accesses eight words in the VRAM.

HDP: Horizontal display period

Use HDPCNT[6:0]/LCDC_HDISP register to set the horizontal display period (= horizontal panel resolution).

$$HDP = (HDPCNT[6:0] + 1) \times 8 [Ts]$$

HDPCNT[6:0] 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]/LCDC_HDISP register to set the horizontal display period start position for the HR-TFT panel.

$$HDPS = HDPCNT[9:0] + 1 [Ts]$$

HDPCNT[9:0] must be programmed such that the following condition is met:

$$HT > HDP + HDPS$$

HPS: Horizontal sync pulse start position

Use FPLINE_ST[9:0]/LCDC_FPLINE register to set the horizontal sync pulse (FPLINE or LP) start position for the HR-TFT panel.

$$HPS = FPLINE_ST[9:0] + 1 [Ts]$$

HPW: Horizontal sync pulse width

Use FPLINE_WD[6:0]/LCDC_FPLINE register to set the horizontal sync pulse width for the HR-TFT panel.

$$HPW = FPLINE_WD[6:0] + 1 [Ts]$$

Horizontal sync pulse polarity

Use FPLINE_POL/LCDC_FPLINE register to set the horizontal sync pulse polarity for the HR-TFT panel.

FPLINE_POL = 1: Active high

FPLINE_POL = 0: Active low (default)

VT: Vertical total period

Use VTCNT[9:0]/LCDC_VDISP register to set the vertical total period.

$$VT = VTCNT[9:0] + 1 [\text{lines}]$$

VTCNT[9:0] must be programmed such that the following condition is met:

$$VT > VDP + VDPS$$

VDP: Vertical display period

Use VDPSCNT[9:0]/LCDC_VDISP register to set the vertical display period (= vertical panel resolution).

$$\text{VDP} = \text{VDPSCNT}[9:0] + 1 \text{ [lines]}$$

VDPSCNT[9:0] must be programmed such that the following condition is met:

$$\text{VT} \geq \text{VDP} + 1$$

VDPS: Vertical display period start position

Use VDPSCNT[9:0]/LCDC_VDPS register to set the vertical display period start position for the HR-TFT panel.

$$\text{VDPS} = \text{VDPSCNT}[9:0] \text{ [lines]}$$

VDPSCNT[9:0] must be programmed such that the following condition is met:

$$\text{VT} > \text{VDP} + \text{VDPS}$$

VPS: Vertical sync pulse start position

Use FPFRAME_ST[9:0]/LCDC_FPFR register to set the vertical sync pulse (FPFRAME or SPS) start position for the HR-TFT panel.

$$\text{VPS} = \text{FPFRAME_ST}[9:0] \text{ [lines]} = \text{FPFRAME_ST}[9:0] \times \text{HT} \text{ [Ts]}$$

VPW: Vertical sync pulse width

Use FPFRAME_WD[6:0]/LCDC_FPFR register to set the vertical sync pulse width for the HR-TFT panel.

$$\text{VPW} = \text{FPFRAME_WD}[6:0] + 1 \text{ [lines]} = (\text{FPFRAME_WD}[6:0] + 1) \times \text{HT} \text{ [Ts]}$$

Vertical sync pulse polarity

Use FPFRAME_POL/LCDC_FPFR register to set the vertical sync pulse polarity for the HR-TFT panel.

FPFRAME_POL = 1: Active High

FPFRAME_POL = 0: Active low (default)

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.

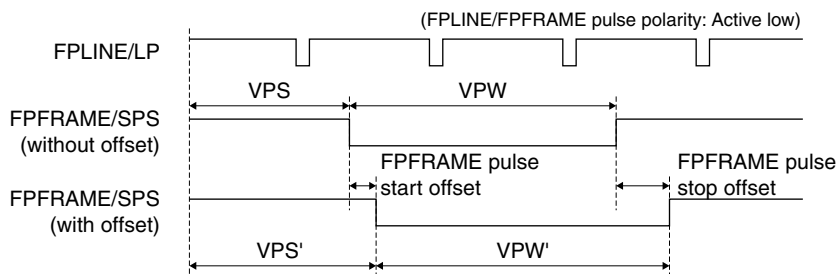


Figure 26.5.3.3 Vertical Sync Pulse Offset

Use FPFRAME_STOFS[9:0]/LCDC_FPFROFS register and FPFRAME_STPOFS[9:0]/LCDC_FPFROFS register to adjust the vertical sync pulse start and stop positions.

$$\text{VPS}' = \text{FPFRAME_ST}[9:0] \times \text{HT} + \text{FPFRAME_STOFS}[9:0] \text{ [Ts]}$$

$$\text{VPW}' = (\text{FPFRAME_WD}[6:0] + 1) \times \text{HT} - \text{FPFRAME_STOFS}[9:0] + \text{FPFRAME_STPOFS}[9:0] \text{ [Ts]}$$

FPSHIFT (DCLK) signal

The FPSHIFT (DCLK) signal polarity for HR-TFT panels can be selected using FPSHIFT_POL/LCDC_TFTSO register.

When HR-TFT panel is selected (PANELSEL/LCDC_DISPMD register = 1), the FPSHIFT (DCLK) 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 FPSHIFT_MSK/LCDC_DISPMD register to 1.

26 LCD CONTROLLER (LCDC)

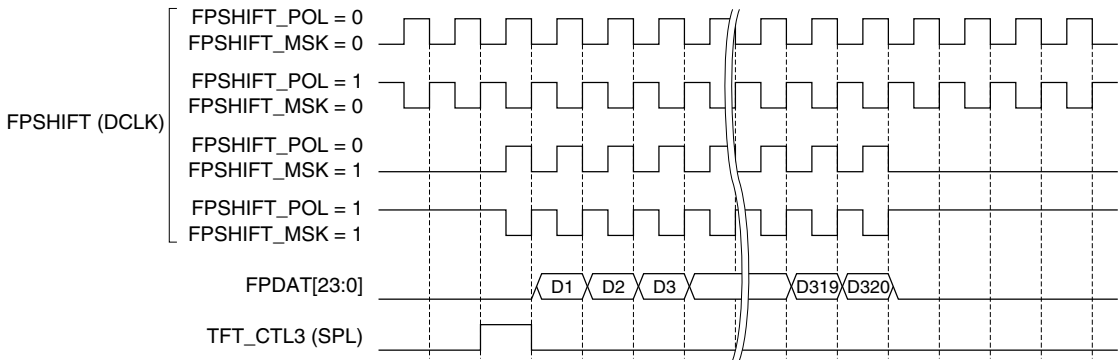


Figure 26.5.3.4 FPSHIFT (DCLK) Variations

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]/LCDC_TFT_CTL1 register to set the pulse start position and CTL1STP[9:0]/LCDC_TFT_CTL1 register to set the pulse stop position. These values should be specified an offset from the FPLINE pulse start position.

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/LCDC_TFTSO register and CTLCNT_RUN/LCDC_TFTSO register must be set to 1.

When CTL1CTL 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 CTL01SWAP/LCDC_TFTSO register.

TFT_CTL1 pin: CLS output (CTL01SWAP = 0), PS output (CTL01SWAP = 1)

TFT_CTL0 pin: PS output (CTL01SWAP = 0), CLS output (CTL01SWAP = 1)

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]/LCDC_TFT_CTL0 register to set the pulse start position and CTL0STP[9:0]/LCDC_TFT_CTL0 register to set the pulse stop position. These values should be specified an offset from the FPLINE pulse start position.

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, CTLCNT_RUN must be set to 1.

The TFT_CTL1 and TFT_CTL0 signals can be swapped using CTL01SWAP.

TFT_CTL2 (REV) delay

Use CTL2DLY[9:0]/LCDC_TFT_CTL2 register to set the TFT_CTL2 toggle edge delay time from the FPLINE pulse start edge. To program the TFT_CTL2 delay time, CTLCNT_RUN must be set to 1.

26.5.4 Display Modes

By setting the LCDC_DISPMOD register, a display mode (color/mono, use of LUT) and a bpp mode (color depth or gray levels) can be selected according to the LCD panel used as shown in Table 26.5.4.1.

Table 26.5.4.1 LCD Panel Type and Display Mode

Panel	LCDC_DISPMOD register			Display mode		Available bpp mode						
	PANSELSEL	COLOR	LUTPASS	Color/Mono	LUT	1	2	4	8	12	16	24
TFT panel	1	1	1	Color	Bypass	✓	✓	✓	✓	✓	✓	✓
	1	1	0	Color	LUTRAM *	✓	✓	✓	✓	✓	✓	✓
CSTN panel	0	1	1	Color	Bypass	✓	✓	✓	✓	✓	✓	✓
	0	1	0	Color	LUTRAM *	✓	✓	✓	✓	✓	✓	✓
MSTN panel	0	0	1	Monochrome	Bypass	✓	✓	✓	✓	✓	✓	✓
	0	0	0	Monochrome	MLUT	✓	✓	✓	✓	✓	✓	✓

* To use LUTRAM, DSTRAM_CFG/MISC_RAM_LOC register must be set to 1. However, set it to 0 when rewriting LUTRAM.

Table 26.5.4.2 lists the number of displayable colors/gray levels in different display modes and bpp modes.

Table 26.5.4.2 Displayable Colors/Gray Levels

bpp mode	Color mode		Monochrome mode	
	LUT bypassed	LUTRAM used	LUT bypassed	MLUT used
1 bpp	Black & White	2 colors (entries) out of 65,536 colors *3	Black & White	2 gray levels (entries) out of 16 gray levels
2 bpp	4 gray levels	4 colors (entries) out of 65,536 colors *3	4 gray levels	4 gray levels (entries) out of 16 gray levels
4 bpp	16 gray levels	16 colors (entries) out of 65,536 colors *3	16 gray levels	16 gray levels (entries) out of 16 gray levels
8 bpp	256 colors (R: 3 bits, G: 3 bits, B: 2 bits)	256 colors (entries) out of 65,536 colors *3		
12 bpp	4,096 colors (R: 4 bits, G: 4 bits, B: 4 bits)			
16 bpp	65,536 colors *1 (R: 5 bits, G: 6 bits, B: 5 bits)			
24 bpp	16,777,216 colors *2 (R: 8 bits, G: 8 bits, B: 8 bits)			

*1: Limited to 4,096 colors in CSTN panels.

*2: Cannot be used in CSTN panels.

*3: The bit configuration of LUTRAM is R: 5 bits, G: 6 bits, and B: 5 bits.

The bpp mode is set using BPP[2:0]/LCDC_DISPMOD register.

Table 26.5.4.3 Bpp Mode Settings

BPP[2:0]	bpp mode
0x7	Reserved
0x6	24 bpp
0x5	16 bpp
0x4	12 bpp
0x3	8 bpp
0x2	4 bpp
0x1	2 bpp
0x0	1 bpp

(Default: 0x0)

To display the number of colors shown above, VRAM data must be configured to the appropriate format with or without the look-up table.

For detailed information on the VRAM data format and look-up tables, see Sections 26.5.5 to 26.5.7.

26.5.5 VRAM Data Format

This section describes the VRAM data format in each bpp mode.

Pixel data is placed in the VRAM beginning with the pixel at the upper-left corner of the screen. The subsequent pixels in the first line follow the first pixel and then the second to last line data follow. Each byte begins with the MSB and byte data are aligned in little-endian format. Color pixel data begins with Red bits, then Green and Blue bits follow.

1-bpp mode (black & white, 2 colors, or 2 gray levels)

In 1-bpp mode, each bit in the VRAM corresponds to one pixel. The bit data represents the brightness (0 or 1) of the pixel when the look-up table is bypassed or an LUT entry number (0 or 1) when the look-up table is used.

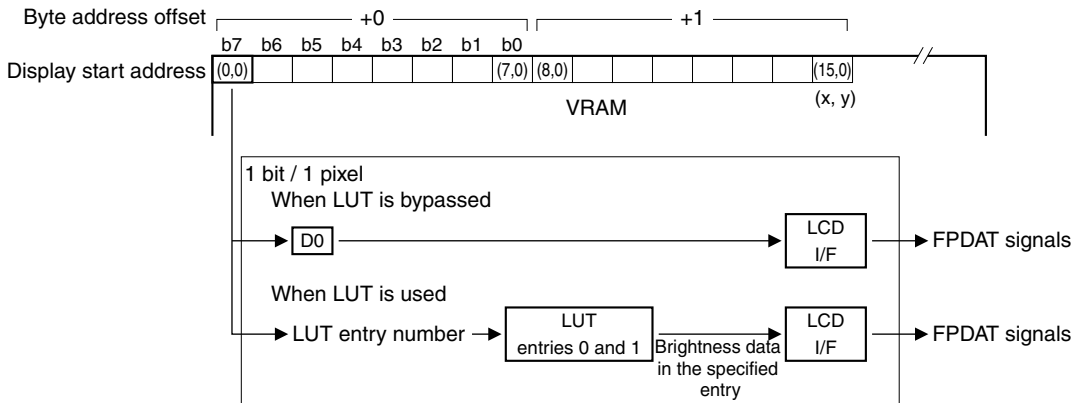
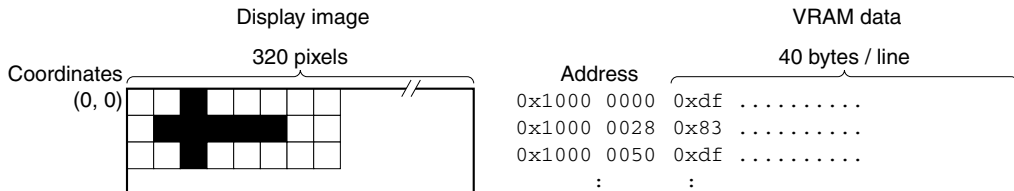


Figure 26.5.5.1 VRAM Data Format in 1-bpp Mode

Example

- VRAM start address: 0x10000000
- Screen width: 320 pixels
- LUT: Bypassed
- LCD characteristics: Data = 0 → Low LCD brightness



Note) Display may be inverted depending on the LCD panel used.

Figure 26.5.5.2 Example of VRAM Data in 1-bpp Mode

2-bpp mode (4 colors or 4 gray levels)

In 2-bpp mode, each 2-bit data in the VRAM corresponds to one pixel. The 2-bit data represents the brightness (0 to 3) of the pixel when the look-up table is bypassed or an LUT entry number (0 to 3) when the look-up table is used.

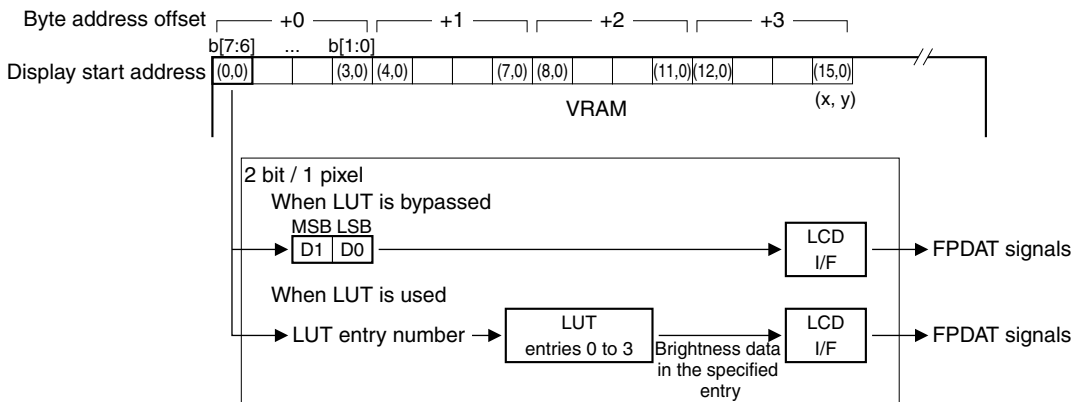
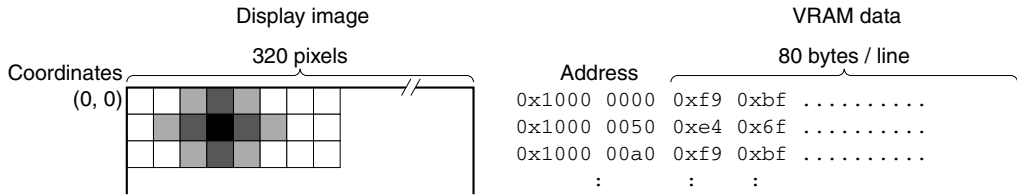


Figure 26.5.5.3 VRAM Data Format in 2-bpp Mode

Example

VRAM start address: 0x10000000
 Screen width: 320 pixels
 LUT: Bypassed
 LCD characteristics: Data = 0 → Low LCD brightness



Note) Display may be inverted depending on the LCD panel used.
 Figure 26.5.5.4 Example of VRAM Data in 2-bpp Mode

4-bpp mode (16 colors or 16 gray levels)

In 4-bpp mode, each 4-bit data in the VRAM corresponds to one pixel. The 4-bit data represents the brightness (0 to 15) of the pixel when the look-up table is bypassed or an LUT entry number (0 to 15) when the look-up table is used.

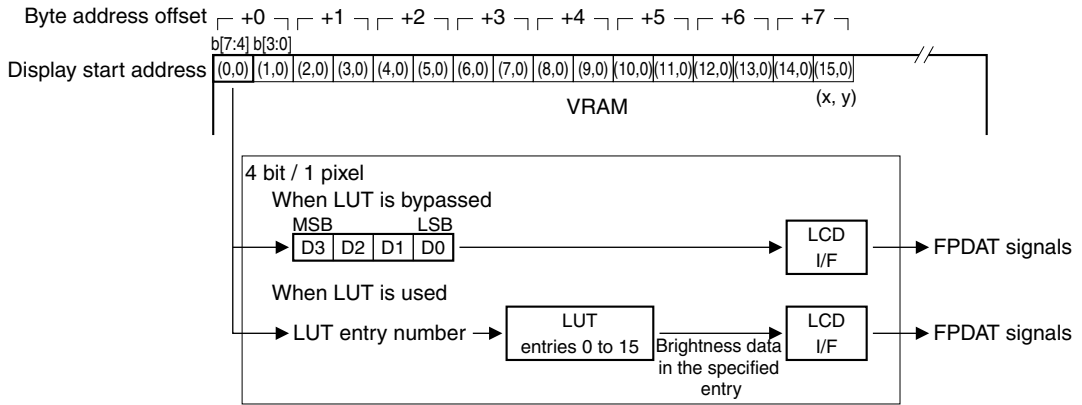
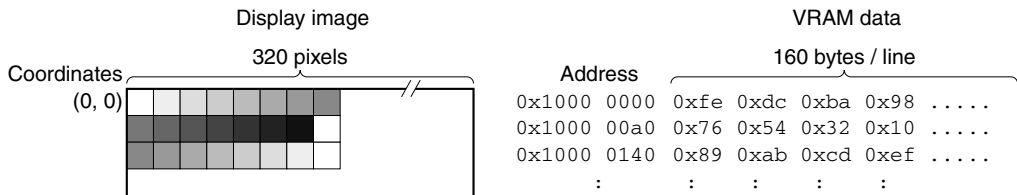


Figure 26.5.5.5 VRAM Data Format in 4-bpp Mode

Example

VRAM start address: 0x10000000
 Screen width: 320 pixels
 LUT: Bypassed
 LCD characteristics: Data = 0 → Low LCD brightness



Note) Display may be inverted depending on the LCD panel used.
 Figure 26.5.5.6 Example of VRAM Data in 4-bpp Mode

8-bpp mode (256 colors)

In 8-bpp mode, each byte in the VRAM corresponds to one pixel. The bit[7:5] (3 bits), bit[4:2] (3 bits), and bit[1:0] (2 bits) in each byte represent the Red, Green, and Blue intensities, respectively, of the pixel when the look-up table is bypassed or an LUT entry number (0 to 255) when the look-up table is used.

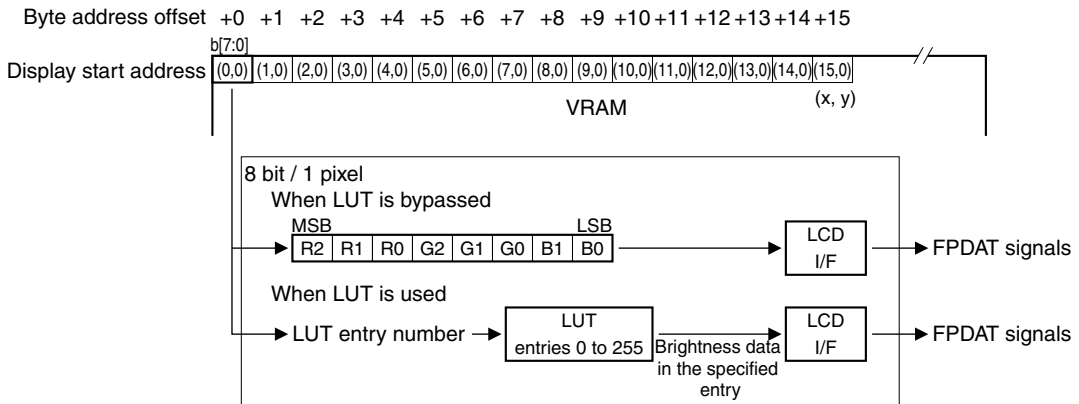
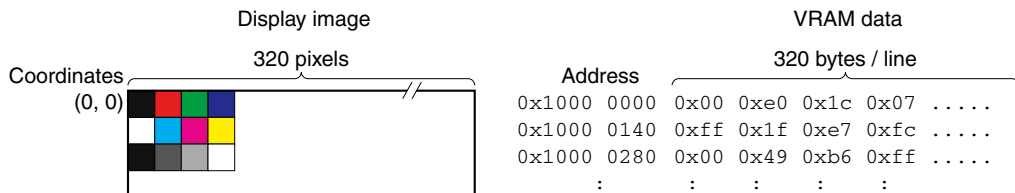


Figure 26.5.5.7 VRAM Data Format in 8-bpp Mode

Example

VRAM start address: 0x10000000
 Screen width: 320 pixels
 LUT: Bypassed
 LCD characteristics: Data = 0 → Low LCD brightness



Note) Display may be inverted depending on the LCD panel used.

Figure 26.5.5.8 Example of VRAM Data in 8-bpp Mode

12-bpp mode (4K colors)

In 12-bpp mode, each 12-bit data in the VRAM corresponds to one pixel. Display data must be manipulated in 3-byte units that consist of an even and odd X coordinate pixel pair. The bit[11:8] (4 bits), bit[7:4] (4 bits), and bit[3:0] (4 bits) in each 12-bit data represent the Red, Green, and Blue intensities, respectively, of the pixel. This mode does not support the look-up table.

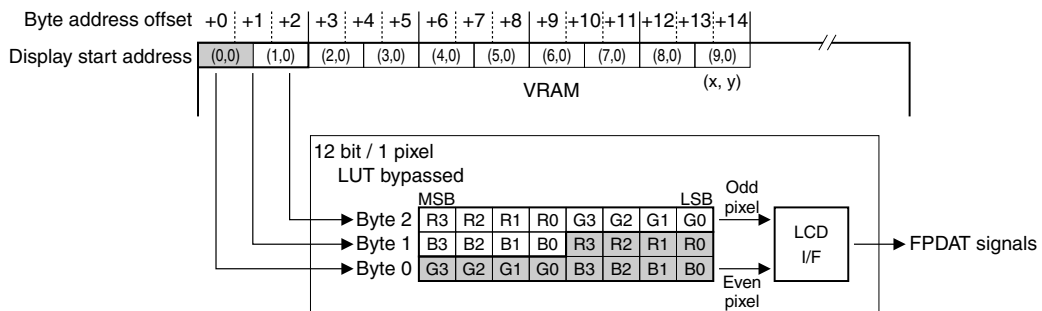
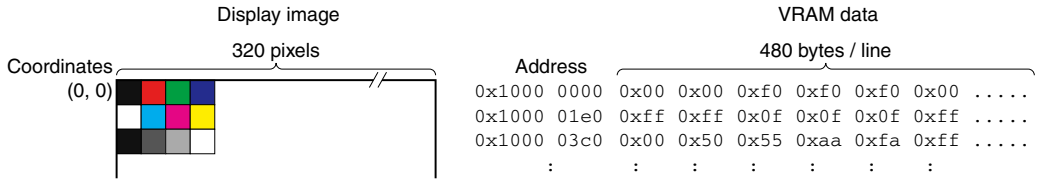


Figure 26.5.5.9 VRAM Data Format in 12-bpp Mode

Example

VRAM start address: 0x10000000
 Screen width: 320 pixels
 LUT: Bypassed
 LCD characteristics: Data = 0 → Low LCD brightness



Note) Display may be inverted depending on the LCD panel used.

Figure 26.5.5.10 Example of VRAM Data in 12-bpp Mode

16-bpp mode (64K colors for TFT, 4K colors for CSTN)

In 16-bpp mode, each 2-byte data in the VRAM corresponds to one pixel. The bit[15:11] (5 bits), bit[10:5] (6 bits), and bit[4:0] (5 bits) in each 16-bit data represent the Red, Green, and Blue intensities, respectively, of the pixel. This mode does not support the look-up table.

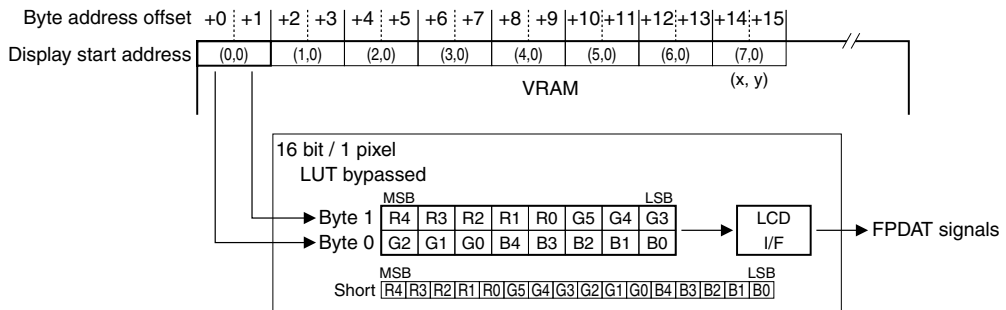
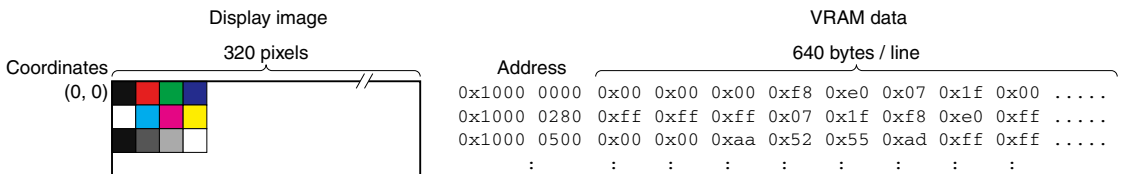


Figure 26.5.5.11 VRAM Data Format in 16-bpp Mode

Example

VRAM start address: 0x10000000
 Screen width: 320 pixels
 LUT: Bypassed
 LCD characteristics: Data = 0 → Low LCD brightness



Note) Display may be inverted depending on the LCD panel used.

Figure 26.5.5.12 Example of VRAM Data in 16-bpp Mode

Note: The LCDC supports up to 4K colors for CSTN panels even if 16-bpp mode is selected.

24-bpp mode (16M colors for TFT)

In 24-bpp mode, each 3-byte data in the VRAM corresponds to one pixel. The bit[23:16] (8 bits), bit[15:8] (8 bits), and bit[7:0] (8 bits) in each 24-bit data represent the Red, Green, and Blue intensities, respectively, of the pixel. This mode does not support the look-up table.

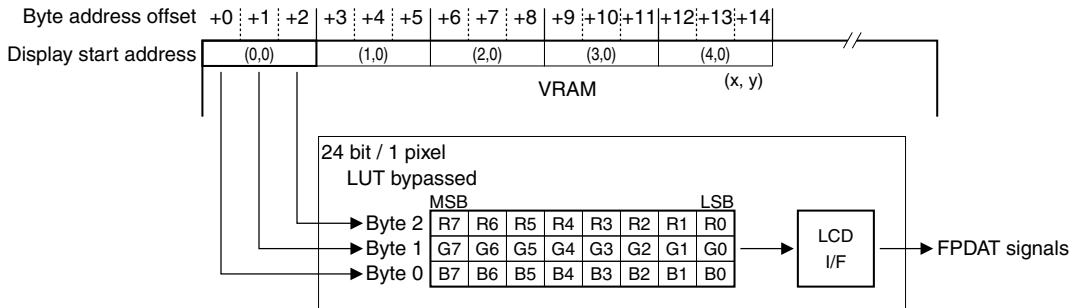
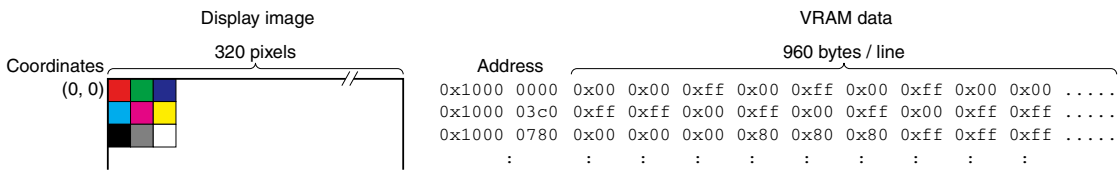


Figure 26.5.5.13 VRAM Data Format in 24-bpp Mode

Example

- VRAM start address: 0x10000000
- Screen width: 320 pixels
- LUT: Bypassed
- LCD characteristics: Data = 0 → Low LCD brightness



Note) Display may be inverted depending on the LCD panel used.

Figure 26.5.5.14 Example of VRAM Data in 24-bpp Mode

Note: When using the GE, 24-bpp mode cannot be set, as the GE does not support 24-bpp data.

26.5.6 LUT Bypass Mode

In LUT bypass mode (LUTPASS/LCDC_DISPMD register = 1), VRAM data are converted directly into the FPDAT signals. This mode always displays gray scale images in 1/2/4-bpp mode or color images in 8/12/16/24-bpp mode regardless of the COLOR/LCDC_DISPMD register setting.

LUT bypass mode for TFT panel

When a TFT panel is used, LUT bypass mode is effective if the conditions shown below are all met.

- The LUT bypass function is enabled (LUTPASS/LCDC_DISPMD register = 1).
- Color mode is selected (COLOR/LCDC_DISPMD register = 1).

In LUT bypass mode, the FPDAT signals are generated directly from pixel data in the VRAM.

The following shows the relationship between pixel data and FPDAT signals in each bpp mode:

Note: The signal levels described in this section assume that SWINV/LCDC_DISPMD register is set to 0. They will be inverted if SWINV is set to 1 (software inverse video enabled).

1-bpp mode (TFT panel, LUT bypassed)

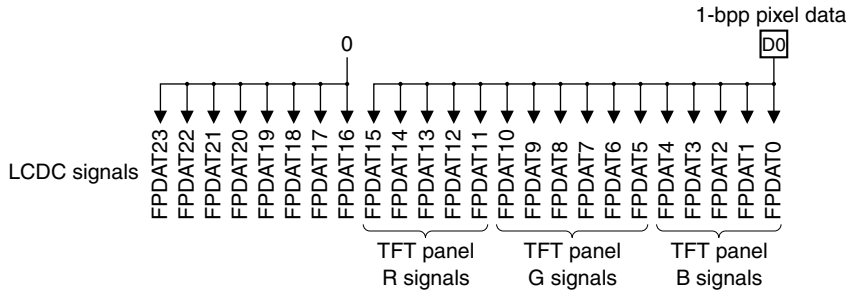


Figure 26.5.6.1 FPDAT Signals in LUT Bypass Mode (TFT panel, 1-bpp mode)

Table 26.5.6.1 Relationship between 1-bpp Pixel Data and FPDAT Signals

Pixel data	FPDAT[15:0] signals	FPDAT[23:16] signals
1	High (1)	Low (0)
0	Low (0)	Low (0)

2-bpp mode (TFT panel, LUT bypassed)

To ensure a uniform brightness, D1 and D0 are connected to the TFT panel RGB signals repeatedly.

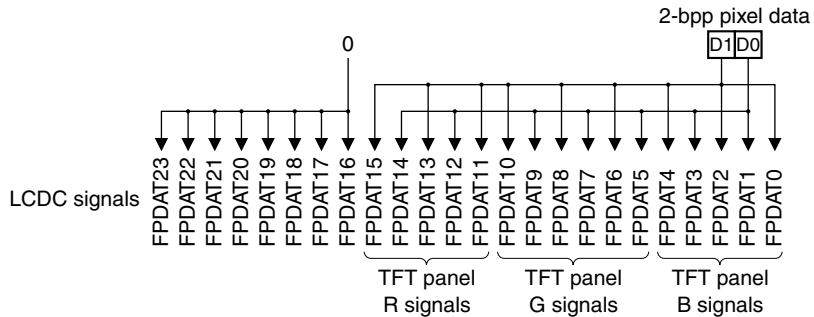


Figure 26.5.6.2 FPDAT Signals in LUT Bypass Mode (TFT panel, 2-bpp mode)

Table 26.5.6.2 Relationship between 2-bpp Pixel Data and FPDAT Signals

Pixel data	FPDAT0/2/4/6/8/10/11/13/15 signals	FPDAT1/3/5/7/9/12/14 signals	FPDAT[23:16] signals
0x3	High (1)	High (1)	Low (0)
0x2	High (1)	Low (0)	Low (0)
0x1	Low (0)	High (1)	Low (0)
0x0	Low (0)	Low (0)	Low (0)

4-bpp mode (TFT panel, LUT bypassed)

To ensure a uniform brightness, D3 to D0 are connected to the TFT panel RGB signals repeatedly.

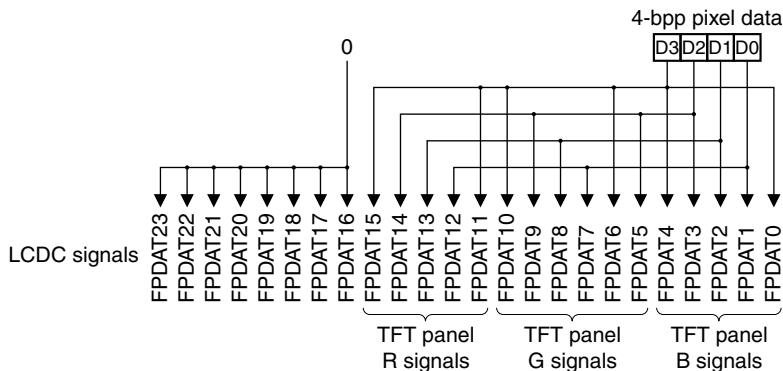


Figure 26.5.6.3 FPDAT Signals in LUT Bypass Mode (TFT panel, 4-bpp mode)

Table 26.5.6.3 Relationship between 4-bpp Pixel Data and FPDAT Signals

Pixel data	FPDAT0/4/6/10 /11/15 signals	FPDAT3/5/9/14 signals	FPDAT2/8/13 signals	FPDAT1/7/12 signals	FPDAT[23:16] signals
0xf	High (1)	High (1)	High (1)	High (1)	Low (0)
0xe	High (1)	High (1)	High (1)	Low (0)	Low (0)
0xd	High (1)	High (1)	Low (0)	High (1)	Low (0)
0xc	High (1)	High (1)	Low (0)	Low (0)	Low (0)
0xb	High (1)	Low (0)	High (1)	High (1)	Low (0)
0xa	High (1)	Low (0)	High (1)	Low (0)	Low (0)
0x9	High (1)	Low (0)	Low (0)	High (1)	Low (0)
0x8	High (1)	Low (0)	Low (0)	Low (0)	Low (0)
0x7	Low (0)	High (1)	High (1)	High (1)	Low (0)
0x6	Low (0)	High (1)	High (1)	Low (0)	Low (0)
0x5	Low (0)	High (1)	Low (0)	High (1)	Low (0)
0x4	Low (0)	High (1)	Low (0)	Low (0)	Low (0)
0x3	Low (0)	Low (0)	High (1)	High (1)	Low (0)
0x2	Low (0)	Low (0)	High (1)	Low (0)	Low (0)
0x1	Low (0)	Low (0)	Low (0)	High (1)	Low (0)
0x0	Low (0)	Low (0)	Low (0)	Low (0)	Low (0)

8-bpp mode (TFT panel, LUT bypassed)

To ensure a uniform brightness, R2–R0, G2–G0, and B1–B0 are connected to the TFT panel RGB signals repeatedly.

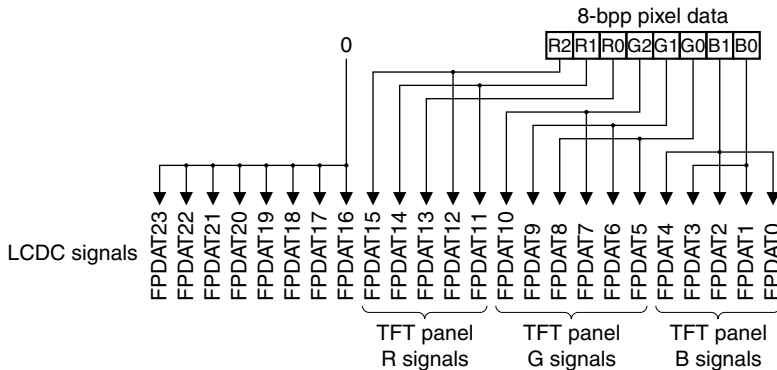


Figure 26.5.6.4 FPDAT Signals in LUT Bypass Mode (TFT panel, 8-bpp mode)

When a pixel data is 0x59 (R = 0x2, G = 0x6, B = 0x1), for example, the FPDAT signals will be configured as follows:

- FPDAT15 = Low (0)
 - FPDAT14 = High (1)
 - FPDAT13 = Low (0)
 - FPDAT12 = Low (0)
 - FPDAT11 = High (1)
 - FPDAT10 = High (1)
 - FPDAT9 = High (1)
 - FPDAT8 = Low (0)
 - FPDAT7 = High (1)
 - FPDAT6 = High (1)
 - FPDAT5 = Low (0)
 - FPDAT4 = Low (0)
 - FPDAT3 = High (1)
 - FPDAT2 = Low (0)
 - FPDAT1 = High (1)
 - FPDAT0 = Low (0)
- FPDAT[23:16] = Low (0)

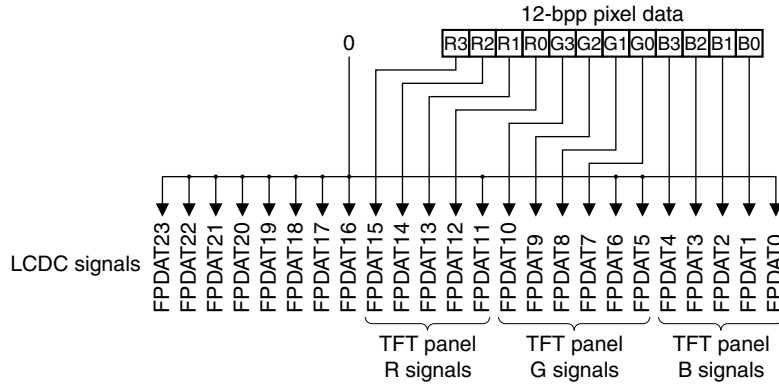
12-bpp mode (TFT panel, LUT bypassed)

Figure 26.5.6.5 FPDAT Signals in LUT Bypass Mode (TFT panel, 12-bpp mode)

When a pixel data is 0xc51 (R = 0xb, G = 0x5, B = 0x1), for example, the FPDAT signals will be configured as follows:

FPDAT15 = High (1)	FPDAT10 = Low (0)	FPDAT4 = Low (0)
FPDAT14 = High (1)	FPDAT9 = High (1)	FPDAT3 = Low (0)
FPDAT13 = Low (0)	FPDAT8 = Low (0)	FPDAT2 = Low (0)
FPDAT12 = Low (0)	FPDAT7 = High (1)	FPDAT1 = High (1)
FPDAT11 = Low (0)	FPDAT6 = Low (0)	FPDAT0 = Low (0)
	FPDAT5 = Low (0)	

FPDAT[23:16] = Low (0)

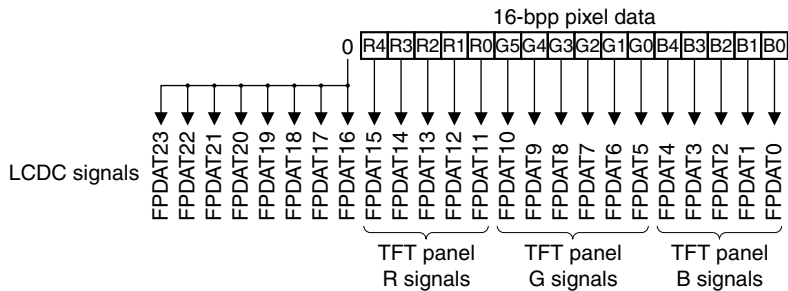
16-bpp mode (TFT panel, LUT bypassed)

Figure 26.5.6.6 FPDAT Signals in LUT Bypass Mode (TFT panel, 16-bpp mode)

When a pixel data is 0x5b37 (R = 0xb, G = 0x19, B = 0x17), for example, the FPDAT signals will be configured as follows:

FPDAT15 = Low (0)	FPDAT10 = Low (0)	FPDAT4 = High (1)
FPDAT14 = High (1)	FPDAT9 = High (1)	FPDAT3 = Low (0)
FPDAT13 = Low (0)	FPDAT8 = High (1)	FPDAT2 = High (1)
FPDAT12 = High (1)	FPDAT7 = Low (0)	FPDAT1 = High (1)
FPDAT11 = High (1)	FPDAT6 = Low (0)	FPDAT0 = High (1)
	FPDAT5 = High (1)	

FPDAT[23:16] = Low (0)

24-bpp mode (TFT panel, LUT bypassed)

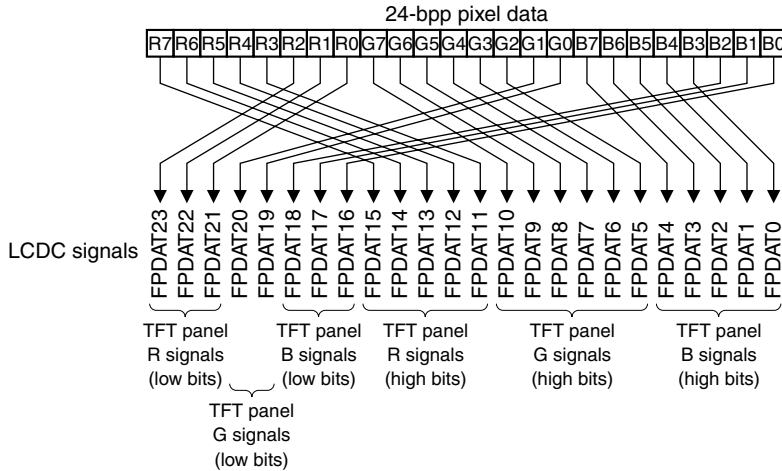


Figure 26.5.6.7 FPDAT Signals in LUT Bypass Mode (TFT panel, 24-bpp mode)

When a pixel data is 0x123456 (R = 0x12, G = 0x34, B = 0x56), for example, the FPDAT signals will be configured as follows:

- | | | |
|--------------------|-------------------|--------------------|
| FPDAT15 = Low (0) | FPDAT10 = Low (0) | FPDAT4 = Low (0) |
| FPDAT14 = Low (0) | FPDAT9 = Low (0) | FPDAT3 = High (1) |
| FPDAT13 = Low (0) | FPDAT8 = High (1) | FPDAT2 = Low (0) |
| FPDAT12 = High (1) | FPDAT7 = High (1) | FPDAT1 = High (1) |
| FPDAT11 = Low (0) | FPDAT6 = Low (0) | FPDAT0 = Low (0) |
| FPDAT23 = Low (0) | FPDAT5 = High (1) | FPDAT18 = High (1) |
| FPDAT22 = High (1) | FPDAT20 = Low (0) | FPDAT17 = High (1) |
| FPDAT21 = Low (0) | FPDAT19 = Low (0) | FPDAT16 = Low (0) |

LUT bypass mode for CSTN panel

When a CSTN panel is used, LUT bypass mode is effective if the conditions shown below are all met.

- The LUT bypass function is enabled (LUTPASS/LCDC_DISPMOD register = 1).
- Color mode is selected (COLOR/LCDC_DISPMOD register = 1).
- 1, 2, 4, 8, 12, or 16-bpp mode is selected (BPP[2:0]/LCDC_DISPMOD register = 0x0 to 0x5).

In LUT bypass mode, pixel data in the VRAM specify a halftone to generate the FPDAT signals. The following shows the relationship between pixel data and FPDAT signals in each bpp mode:

Note: CSTN panels turn the pixels on and off periodically to generate halftones. The FPDAT signals do not represent colors directly in contrast to TFT panels.

1-bpp mode (CSTN panel, LUT bypassed)

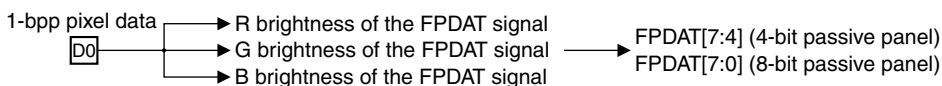


Figure 26.5.6.8 FPDAT Signals in LUT Bypass Mode (CSTN panel, 1-bpp mode)

2-bpp mode (CSTN panel, LUT bypassed)

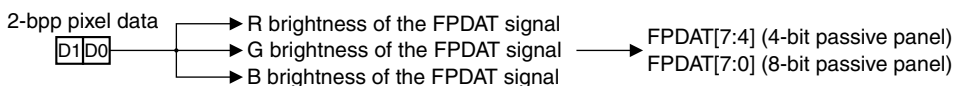


Figure 26.5.6.9 FPDAT Signals in LUT Bypass Mode (CSTN panel, 2-bpp mode)

4-bpp mode (CSTN panel, LUT bypassed)

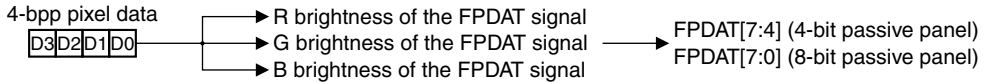


Figure 26.5.6.10 FPDAT Signals in LUT Bypass Mode (CSTN panel, 4-bpp mode)

8-bpp mode (CSTN panel, LUT bypassed)

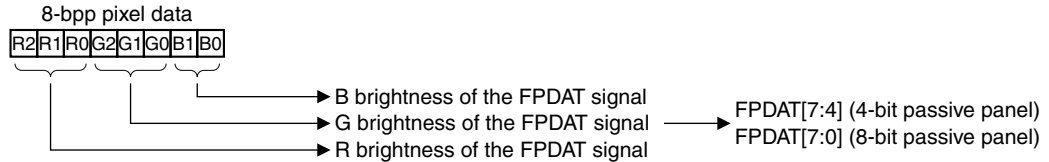


Figure 26.5.6.11 FPDAT Signals in LUT Bypass Mode (CSTN panel, 8-bpp mode)

12-bpp mode (CSTN panel, LUT bypassed)

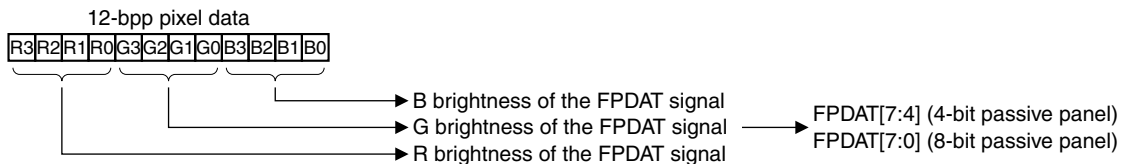


Figure 26.5.6.12 FPDAT Signals in LUT Bypass Mode (CSTN panel, 12-bpp mode)

16-bpp mode (CSTN panel, LUT bypassed)

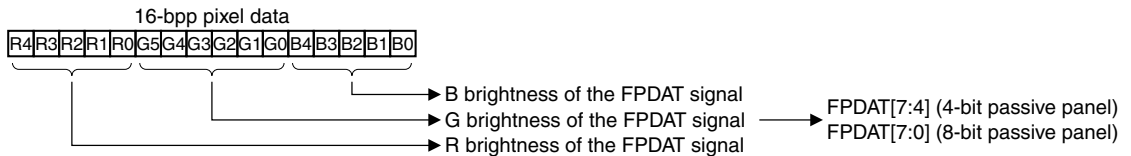


Figure 26.5.6.13 FPDAT Signals in LUT Bypass Mode (CSTN panel, 16-bpp mode)

LUT bypass mode for MSTN panel

When an MSTN panel is used, LUT bypass mode is effective if the conditions shown below are all met.

- The LUT bypass function is enabled (LUTPASS/LCDC_DISPMD register = 1).
- Monochrome mode is selected (COLOR/LCDC_DISPMD register = 0).
- 1, 2, or 4-bpp mode is selected (BPP[2:0]/LCDC_DISPMD register = 0x0 to 0x2).

In LUT bypass mode, pixel data in the VRAM specify a gray level to generate the FPDAT signals.

The following shows the relationship between pixel data and FPDAT signals in each bpp mode:

Note: An FPDAT signal is assigned to a pixel and 4 or 8 pixel data are sent to the MSTN panel at the same time. MSTN panels turn the pixels on and off periodically to generate gray levels. The FPDAT signals do not represent gray levels directly in contrast to TFT panels.

1-bpp mode (MSTN panel, LUT bypassed)



Figure 26.5.6.14 FPDAT Signals in LUT Bypass Mode (MSTN panel, 1-bpp mode)

2-bpp mode (MSTN panel, LUT bypassed)



Figure 26.5.6.15 FPDAT Signals in LUT Bypass Mode (MSTN panel, 2-bpp mode)

4-bpp mode (MSTN panel, LUT bypassed)



Figure 26.5.6.16 FPDAT Signals in LUT Bypass Mode (MSTN panel, 4-bpp mode)

26.5.7 Look-up Tables

When the look-up table is enabled, a pixel data in the VRAM is used as an index to the look-up table, so that FPDAT signals are generated based on the color or gray level stored in the LUT entry indicated by the VRAM data, before being output to the LCD panel.

The S1C33L26 includes two different look-up tables, LUTRAM for color mode and MLUT for monochrome mode.

Color look-up table (LUTRAM)

The LCDC uses LUTRAM to convert VRAM data into color data if the conditions shown below are all met.

- The LUT bypass function is disabled (LUTPASS/LCDC_DISPMD register = 0).
- Color mode is selected (COLOR/LCDC_DISPMD register = 1).
- DSTRAM (Area 3) is configured as LUTRAM (DSTRAM_CFG/MISC_IRAM_LOC register = 1).
- 1, 2, 4, or 8-bpp mode is selected (BPP[2:0]/LCDC_DISPMD register = 0x0 to 0x3).

Note: Ensure sufficient memory bandwidth for LCDC processing when using the color look-up table. An external SDRAM is recommended for the VRAM.

The color look-up table consists of 16 bits × 256 entries as shown in the figure below.

Each entry consists of 5-bit Red data (D[15:11]), 6-bit Green data (D[10:5]), and 5-bit Blue data (D[4:0]). This allows the software to set a color out of 65,536 colors in each entry. The entries to be used depend on the bpp mode selected.

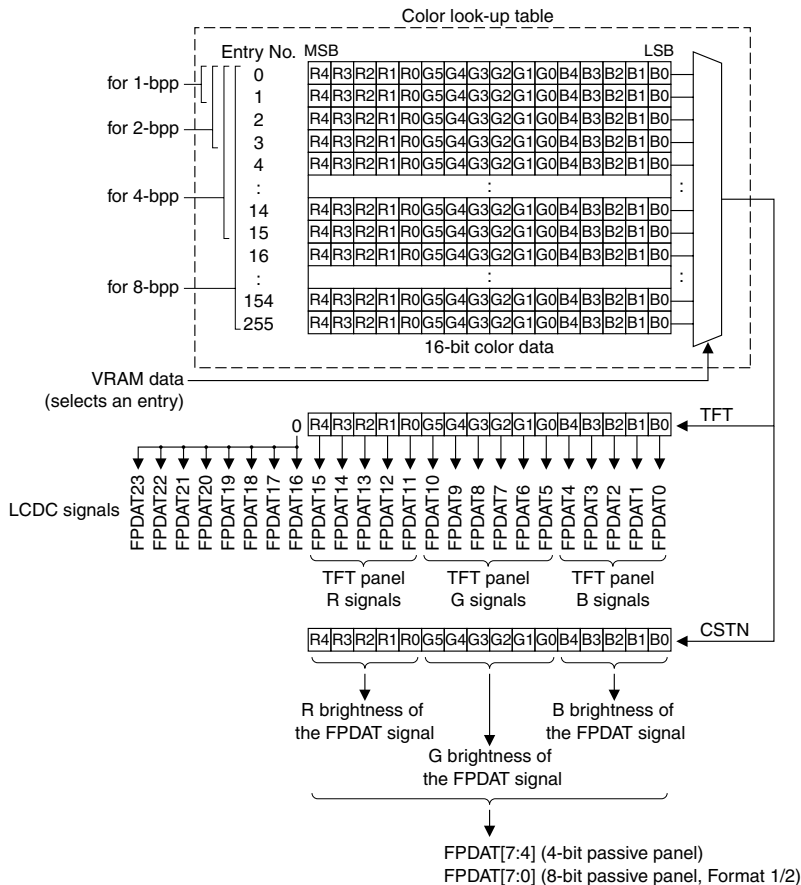


Figure 26.5.7.1 Color Look-up Table Configuration

When an entry data is 0x5b37 (R = 0xb, G = 0x19, B = 0x17), for example, the FPDAT signals (for TFT) will be configured as follows:

FPDAT15 = Low (0)	FPDAT10 = Low (0)	FPDAT4 = High (1)
FPDAT14 = High (1)	FPDAT9 = High (1)	FPDAT3 = Low (0)
FPDAT13 = Low (0)	FPDAT8 = High (1)	FPDAT2 = High (1)
FPDAT12 = High (1)	FPDAT7 = Low (0)	FPDAT1 = High (1)
FPDAT11 = High (1)	FPDAT6 = Low (0)	FPDAT0 = High (1)
	FPDAT5 = High (1)	

FPDAT[23:16] = Low (0) (Note) Display may be inverted depending on the LCD panel used.

Note: CSTN panels turn the pixels on and off periodically to generate halftones. The FPDAT signals do not represent colors directly in contrast to TFT panels.

Setting data to LUTRAM

LUTRAM (DSTRAM) cannot be accessed from the CPU, when it is configured as the color look-up table (DSTRAM_CFG = 1). Therefore, prepare the look-up table data in another memory (IVRAM or an external memory) and use the LCDC reload function to set the prepared data to the LUTRAM. For more information on the LCDC reload function, see Section 26.9.

Monochrome look-up table (MLUT)

The LCDC uses MLUT to convert VRAM data into gray scale data if the conditions shown below are all met.

- The LUT bypass function is disabled (LUTPASS/LCDC_DISPMD register = 0).
- Monochrome mode is selected (COLOR/LCDC_DISPMD register = 0).
- 1, 2, or 4-bpp mode is selected (BPP[2:0]/LCDC_DISPMD register = 0x0 to 0x2).

The monochrome look-up table consists of 4 bits × 16 entries as shown in the figure below. The software can set a gray level out of 16 levels in each entry. The entries to be used depend on the bpp mode selected.

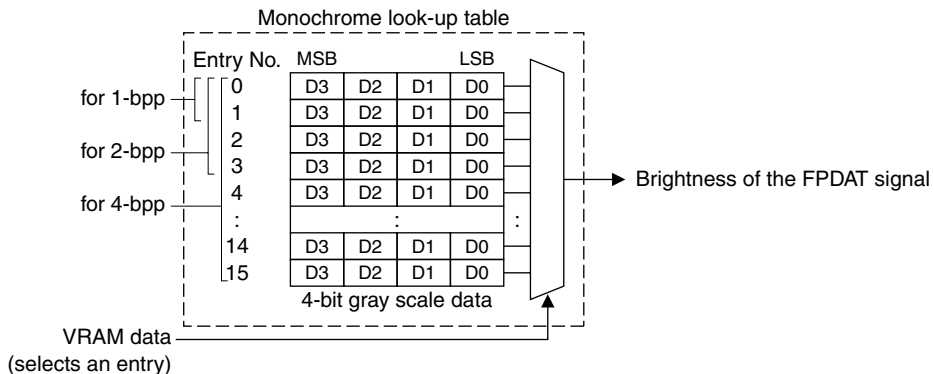


Figure 26.5.7.2 Monochrome Look-up Table Configuration

Note: An FPDAT signal is assigned to a pixel and 4 or 8 pixel data are sent to the MSTN panel at the same time. MSTN panels turn the pixels on and off periodically to generate gray levels. The FPDAT signals do not represent gray levels directly in contrast to TFT panels.

Setting data to the monochrome look-up table

Use the monochrome look-up table data registers for writing and reading 4-bit gray scale data to/from the look-up table. The monochrome look-up table data registers are mapped to addresses 0x302090 and 0x302094.

Table 26.5.7.1 Monochrome Look-up Table Registers

LUT entry	LUT bit/register
0	MLUT0[3:0]/LCDC_MLUT0
1	MLUT1[3:0]/LCDC_MLUT0
2	MLUT2[3:0]/LCDC_MLUT0
3	MLUT3[3:0]/LCDC_MLUT0
4	MLUT4[3:0]/LCDC_MLUT0
5	MLUT5[3:0]/LCDC_MLUT0
6	MLUT6[3:0]/LCDC_MLUT0
7	MLUT7[3:0]/LCDC_MLUT0
8	MLUT8[3:0]/LCDC_MLUT1
9	MLUT9[3:0]/LCDC_MLUT1
10	MLUT10[3:0]/LCDC_MLUT1
11	MLUT11[3:0]/LCDC_MLUT1
12	MLUT12[3:0]/LCDC_MLUT1
13	MLUT13[3:0]/LCDC_MLUT1
14	MLUT14[3:0]/LCDC_MLUT1
15	MLUT15[3:0]/LCDC_MLUT1

In addition settings shown above, the LCDC reload function can be used to replace the monochrome look-up table data by the values prepared in the memory. For more information on the LCDC reload function, see Section 26.9.

26.5.8 Frame Rates

The frame rate is calculated from the LCD panel's horizontal and vertical total periods, and pixel clock frequency, as shown below.

$$\text{Frame rate} = \frac{f_{\text{CLK}}}{\text{HT} \times \text{VT}}$$

f_{CLK} : Pixel clock frequency

$f_{\text{CLK}} = \text{OSC3}/1 \text{ to } \text{OSC3}/32 \text{ (Hz)}$ see Section 26.4.2, "Setting the LCDC Clock."

HT: Horizontal total period

$\text{HT} = (\text{HTCNT}[6:0] + 1) \times 8 \text{ (Ts)}$ where Ts = pixel clock period

VT: Vertical total period

$\text{VT} = \text{VTCNT}[9:0] + 1 \text{ (lines)}$

26.5.9 Other Settings

MOD rate

The period during which the MOD signal is switched can be set using the MOD[5:0]/LCDC_MODR register.

MOD = 0x0: MOD signal switched at a period of the FPPFRAME signal (default)

MOD = other than 0x0: Switched at a period of MOD + 1 FPLINE pulses

Repeating of the FRM pattern

This setup item is provided for EL panels. Whether the frame-rate modulation pattern is to be repeated every 0x40000 frames (counted by the internal frame counter) can be set using FRMRPT/LCDC_DISPMD register.

FRMRPT = 1: FRM pattern repeated (for EL panel)

FRMRPT = 0: FRM pattern not repeated (default)

26.6 Display Control

26.6.1 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. When the LCD panel is placed in power-save mode, all LCD signal output pins are fixed at low.

To change the LCD controller from power-save mode back into normal mode, set the PSAVE[1:0]/LCDC_PSAVE register to 0x3. 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] to 0x0. 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.

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 “26.4 System Settings”).
2. Set the LCD-panel parameters, display mode, and look-up tables (refer to “26.5 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 “26.6.2 Main Window Display Start Address and Virtual Screen Settings”).
6. Place the LCD controller in normal mode (PSAVE = 0x3).
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 = 0x0).
4. The LCD controller pulls LCD signals down to low.

26.6.2 Main Window Display Start Address and Virtual Screen Settings

Main window display start address

The display memory address from which to start display for the main window can be changed as desired using the LCDC_MAINADR register. The start address set in this register 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 VRAM must be specified to this register.

Main screen address offset for virtual screen

The S1C33L26 LCDC supports a virtual screen feature to use the VRAM with a different size from that of the LCD panel.

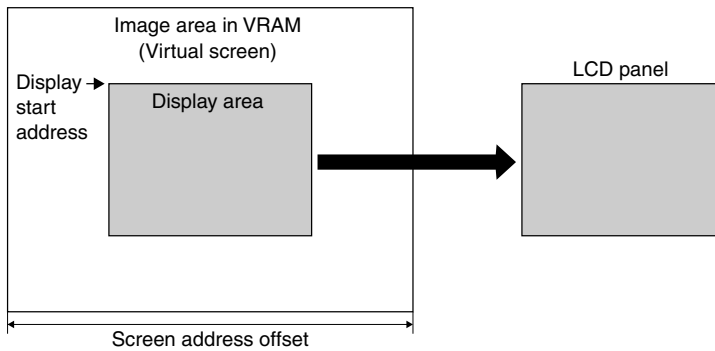


Figure 26.6.2.1 Virtual Screen Feature

The screen address offset is the number of words corresponding to the image width (virtual screen width). The LCDC manipulates display data in units of words. Therefore, the image width (number of pixels) must be a multiple of (32 bits ÷ bpp).

The screen address offset is calculated as follows:

$$\text{Main screen address offset} = \text{Virtual screen width in pixels} \times \text{bpp} / 32 \text{ [words]}$$

Set this value to MW_OFS[11:0]/LCDC_MAINOFS register.

Example: LCD panel size = 320 × 240 pixels, 8-bpp mode, image (virtual screen) size = 640 × 480 pixels

$$\text{Display line width} = 320 \times 8 / 32 = 80 \text{ (=0x50) [words]} \quad (= 320 \text{ [bytes]})$$

$$\text{MW_OFS[11:0]} = 640 \times 8 / 32 = 160 \text{ (=0xa0) [words]} \quad (= 640 \text{ [bytes]})$$

If the image data start address is 0x10000000 and the image area ([X1, Y1]–[X2, Y2] = [160, 120]–[479, 359]) is displayed in the main window, the main window start address is 0x10012ca0.

$$(X1 \times 8 / 32)_{\text{[word]}} + \text{MW_OFS[11:0]}_{\text{[word]}} \times Y1 = 0x4b28 \text{ [words]} = 0x12ca0 \text{ [bytes]}$$

$$\text{Main window start address} = 0x10000000 + 0x12ca0 = 0x10012ca0$$

The LCDC determines the addresses of the first and end pixels in each line as follows:

$$\text{First pixel address in } N\text{th line} = \text{Display start address} + (N - 1) \times \text{MW_OFS[11:0]}_{\text{[word]}}$$

$$\text{End pixel address in } N\text{th line} = \text{Display start address} + (N - 1) \times \text{MW_OFS[11:0]}_{\text{[word]}} + \text{Display line width}_{\text{[word]}} - 1$$

$$\text{1st line start address} = 0x10012ca0$$

$$\text{1st line end address} = 0x10012ca0 + (0x50 - 1) \times 4 = 0x10012dc4$$

$$\text{2nd line start address} = 0x10012ca0 + 1 \times 0xa0 \times 4 = 0x10012f20$$

$$\text{2nd line end address} = 0x10012f20 + (0x50 - 1) \times 4 = 0x10013044$$

:

$$\text{240th line start address} = 0x10012ca0 + 239 \times 0xa0 \times 4 = 0x10038220$$

$$\text{240th line end address} = 0x10038220 + (0x50 - 1) \times 4 = 0x10038344$$

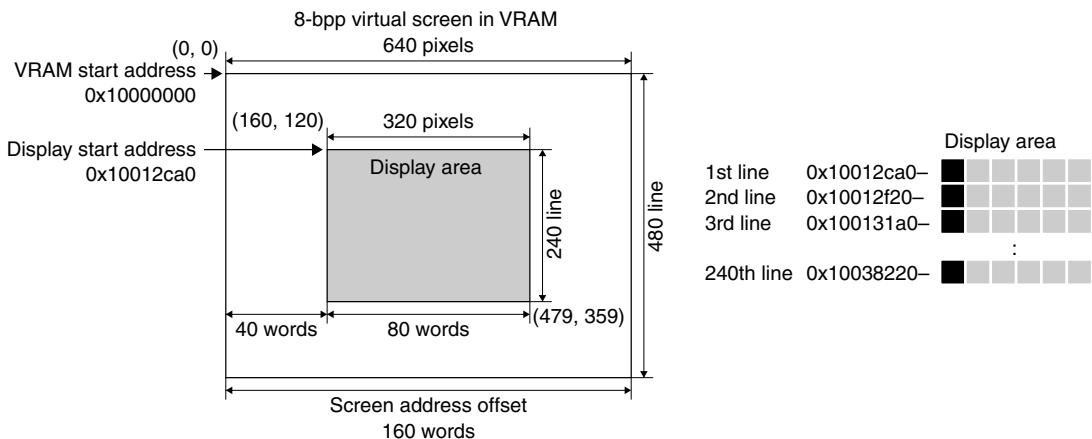


Figure 26.6.2.2 Example of Virtual Screen Configuration

26.6.3 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/LCDC_PSAVE register is provided and is set to 1 if the display is in a vertical non-display period.

To eliminate screen flicker, display data, LUT data and the display buffer should be changed in a vertical non-display period by using this interrupt or VNDPF.

For more information on the LCDC interrupt, see Section 26.7, “LCDC Interrupt.”

26.6.4 Inverting and Blanking the Display

The display can be blanked (the entire screen turned black or white) without rewriting the contents of the VRAM. Setting BLANK/LCDC_DISPMOD register 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/LCDC_DISPMOD register described below.

The display can be inverted simply by manipulating a control bit. Setting SWINV to 1 inverts the display, and setting it to 0 returns the display to normal. This is accomplished by inverting the display data output from the LCDC, rather than by inverting the pixel data in the display memory.

The screen can be made to blink using these operations. Make sure switching takes place within the vertical non-display period (VNDPF = 1).

26.6.5 Picture-in-Picture Plus and Sub-Window

Picture-in-Picture Plus enables a sub-window to be overlaid on 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 sub-window also supports the virtual screen feature the same as the main window.

The following diagram shows the sub-window configuration parameters.

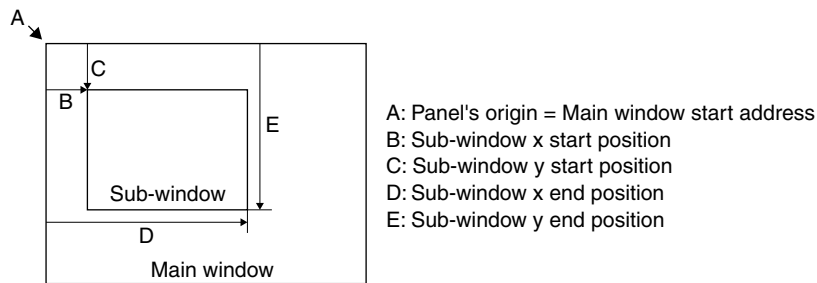


Figure 26.6.5.1 Sub-window Configuration Parameters

Display memory for the sub-window

The display data for the sub-window can be stored in IVRAM or the external VRAM. 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 LCDC_SUBADR register in the same manner as the main window. The start address set in the LCDC_SUBADR register 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 VRAM 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 PIP_XSTART[9:0]/LCDC_SUBSP register for the X coordinate and PIP_YSTART[9:0]/LCDC_SUBSP register for the Y coordinate. Use PIP_XEND[9:0]/LCDC_SUBEP register for specifying the X coordinate of the end position and PIP_YEND[9:0]/LCDC_SUBEP register for the Y coordinate.

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 $(32 \text{ bits} \div \text{bpp})$ 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
- 24-bpp mode: 3-word = 4-pixel units (because the value must be an integer)

For example, to specify the sub-window horizontal start position as 80 pixels in 8-bpp mode, set PIP_XSTART[9:0] 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 PIP_YSTART[9:0] to 60.

Sub-screen address offset for virtual screen

The virtual screen feature can also be used for the sub-window. Specify the screen address offset using SW_OFS[11:0]/LCDC_SUBOFS register.

$$\text{Sub-screen address offset} = \text{Virtual screen width in pixels} \times \text{bpp} / 32 \text{ [words]}$$

See “Main screen address offset for virtual screen” in Section 26.6.2 for more information on the virtual screen and the configurations.

Sub-window display control

The Picture-in-Picture Plus function is enabled and the sub-window is displayed by setting PIP_EN/LCDC_SUBSP register to 1. This bit must be set after the sub-window configuration parameters are set up.

At initial reset, PIP_EN 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, from 0x10000000)
- Virtual main screen size: 800 × 600 pixels, located at 0x10000000
- Virtual sub-screen size: 640 × 480 pixels, located at 0x10100000 (larger screen than the main screen, such as 1024 × 768 pixels can also be configured.)
- Sub-window size: 160 × 120 pixels
- Sub-window start position: X = 80 pixels, Y = 60 pixels

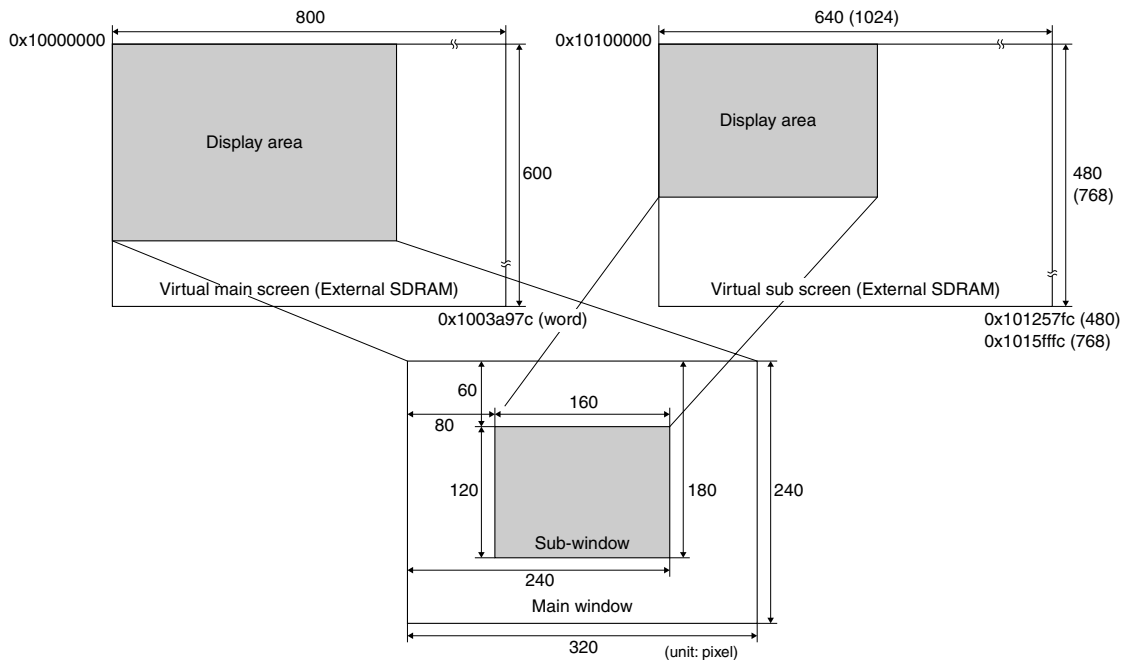


Figure 26.6.5.2 Sub-Window Configuration Example

1. Main window start address

LCDC_MAINADR register = 0x10000000 to 0x1003a97c (word boundary address)

This register is used to change the main window location in the virtual main screen VRAM.

If LCDC_MAINADR = 0x1003a97c, the main window on the LCD panel can display only the first 8 pixels. The other pixels are not displayed properly, as the virtual main screen VRAM ends at that word.

2. Main screen address offset

MW_OFS[11:0] = 800 pixels \times 4 bpp \div 32 bits = 100 words

LCDC_MAINOFS register = 100 = 0x64

3. Sub-window start address

LCDC_SUBADR register = 0x10100000 to 0x101257fc (for 640 \times 480 pixels, word boundary address)

LCDC_SUBADR register = 0x10100000 to 0x1015fffc (for 1024 \times 768 pixels, word boundary address)

This register is used to change the sub-window location in the virtual sub-screen VRAM.

If LCDC_SUBADR = 0x101257fc or 0x1015fffc, the sub-window on the LCD panel can display only the first 8 pixels. The other pixels are not displayed properly, as the virtual sub-screen VRAM ends at that word.

4. Sub-screen address offset

SW_OFS[11:0] = 640 pixels \times 4 bpp \div 32 bits = 80 words (for 640 \times 480 pixels)

LCDC_SUBOFS register = 80 = 0x50

SW_OFS[11:0] = 1024 pixels \times 4 bpp \div 32 bits = 128 words (for 1024 \times 768 pixels)

LCDC_SUBOFS register = 128 = 0x80

5. Sub-window end position

PIP_XEND[9:0] = (80 + 160) pixels \times 4 bpp \div 32 bits - 1 = 29 words = 0x1d

PIP_YEND[9:0] = 60 + 120 lines - 1 = 179 lines = 0xb3

LCDC_SUBEP register = 0x00b3001d

26 LCD CONTROLLER (LCDC)

6. Sub-window start position

$PIP_XSTART[9:0] = 80 \text{ pixels} \times 4 \text{ bpp} \div 32 \text{ bits} = 10 \text{ words} = 0x0a$
 $PIP_YSTART[9:0] = 60 \text{ lines} = 0x3c$
 $LCDC_SUBSP \text{ register} = 0x803c000a$

The MSB in this register is the sub-window enable bit. Setting the MSB enables the sub-window to be displayed.

7. Note on using 12 bpp or 24 bpp mode

If the LCDC is used in 12 bpp or 24 bpp mode, the addresses must be aligned with a 3-word boundary.

$LCDC_MAINADR \text{ register} = 0x10000000 + 3n \text{ (words)}$
 $LCDC_SUBADR \text{ register} = 0x10100000 + 3n \text{ (words)}$
 $PIP_XSTART[9:0] = 3n \text{ (words)}$
 $PIP_XEND[9:0] = 3n + 2 \text{ (words)}$

26.7 LCDC Interrupt

The LCDC module can generate frame interrupts.

Frame interrupt

To use this interrupt, set FRINTEN/LCDC_INT register to 1. If FRINTEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When a vertical non-display period begins after a frame refresh cycle (vertical display period) has finished, VNDPF/LCDC_PSAVE register is set to 1. At the same time, FRINTF/LCDC_PSAVE register is set to 1 and the LCDC outputs an interrupt signal to the interrupt controller (ITC) if frame interrupts are enabled (FRINTEN = 1). An interrupt occurs if other interrupt conditions are met.

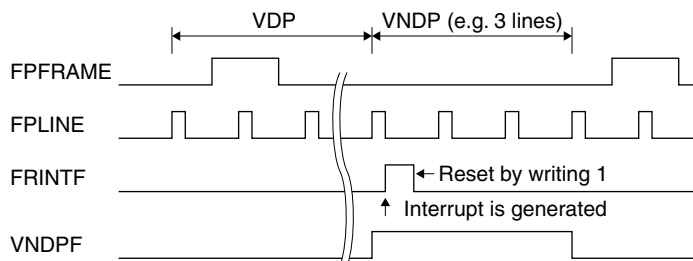


Figure 26.7.1 Frame Interrupt Timing

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

26.8 Power Save

The LCD controller has a power-save modes. Use PSAVE[1:0]/LCDC_PSAVE register to set the power-save mode.

Table 26.8.1 Power-Save Mode Settings

PSAVE[1:0]	Mode
0x3	Normal operation
0x2	Reserved
0x1	Reserved
0x0	Power-save mode

(Default: 0x0)

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[1:0] to 0x0.

The LCD controller is taken out of power-save mode by setting PSAVE[1:0] to 0x3.

Comparison of power-save and normal modes

The differences between power-save and normal modes are summarized in Table 26.8.2.

Table 26.8.2 Differences between Power-Save and Normal Modes

Item	Power-save mode	Normal
Accessing I/O registers	Enabled	Enabled
Accessing look-up table	Enabled	Enabled
Accessing VRAM	Enabled	Enabled
Display (STN panels)	Inactive	Active
Display (HR-TFT panels)	Inactive	Active
LCDC display-data-fetch operation	Inactive	Active
FPDAT[23:0] signals (STN, HR-TFT panels)	Low	Active
FPSHIFT signal (STN panels)	Low	Active
FPLINE, FPFRRAME, FPRDY signals (STN panels)	Low	Active
FPSHIFT signal (HR-TFT panels) when FPSHIFT_POL/LCDC_TFTSO register = 0	High	Active
FPSHIFT signal (HR-TFT panels) when FPSHIFT_POL/LCDC_TFTSO register = 1	Low	Active
FPFRAME signal (HR-TFT panels)	High/Low	Active
FPLINE signal (HR-TFT panels)	High/Low	Active
TFT_CTL0 signal (HR-TFT panels)	High/Low	Active
TFT_CTL1 signal (HR-TFT panels)	High/Low	Active
TFT_CTL2 signal (HR-TFT panels)	High/Low	Active
TFT_CTL3 signal (HR-TFT panels)	Low	Active

26.9 Reload Functions

The LCDC supports two reload functions, Control table reload function and LUT reload function, that reset the LCDC control registers and look-up tables using the reload data prepared in the memory.

Control table reload function

The control table reload function is used to back up and restore LCDC control register settings.

The table below shows the contents of the reload table used in the control table reload function.

Table 26.9.1 Reload Table Contents (LCDC Registers)

Address	Control register
Base + 0x00	LCDC Display Mode Register (LCDC_DISPMDR), 0x302060
Base + 0x04	Main Window Display Start Address Register (LCDC_MAINADR), 0x302070
Base + 0x08	Main Screen Address Offset Register (LCDC_MAINOFS), 0x302074
Base + 0x0c	Sub-window Display Start Address Register (LCDC_SUBADR), 0x302080
Base + 0x10	Sub-screen Address Offset Register (LCDC_SUBOFS), 0x302084
Base + 0x14	Sub-window Start Position Register (LCDC_SUBSP), 0x302088
Base + 0x18	Sub-window End Position Register (LCDC_SUBEP), 0x30208c

Base: Reload table start address

The reload table can be located in IVRAM or an external memory, and the start address (Base) can be specified using RTBL_BADR[31:10]/LCDC_RLDADR register. The low-order 10 bits of the LCDC_RLDADR register is fixed at 0x0, so the reload table always begins from a 1K-byte boundary address. Two or more reload tables can be prepared and switched by changing RTBL_BADR[31:10].

When a memory space is allocated to a reload table, the contents shown above must be programmed in the application program.

The reload table bit configuration is the same as that of the LCDC control registers.

Writing 1 to CTABRLD/LCDC_RLDCTL register resets the LCDC control registers with the reload table data. This reload operation should be performed during a vertical non-display period. CTABRLD retains 1 during reloading and it reverts to 0 when the reloading is completed.

LUT reload function

The LUT reload function is used to replace the look-up table settings. This function is effective when the look-up table function is enabled (LUTPASS/LCDC_DISPMDR = 0).

The table below shows the contents of the reload table used in the LUT reload function.

Table 26.9.2 Reload Table Contents (LUT data)

Address	Monochrome LUT data (when COLOR = 0)	Color LUT data (when COLOR = 1)
Base + 0x100	MLUT0[3:0]–MLUT7[3:0]	RLUT0[5:1], GLUT0[5:0], BLUT0[5:1]
Base + 0x102	(same as the LCDC_MLUT0 register)	RLUT1[5:1], GLUT1[5:0], BLUT1[5:1]
Base + 0x104	MLUT8[3:0]–MLUT15[3:0]	RLUT2[5:1], GLUT2[5:0], BLUT2[5:1]
Base + 0x106	(same as the LCDC_MLUT1 register)	RLUT3[5:1], GLUT3[5:0], BLUT3[5:1]
:	:	:
Base + 0x2fc	X	RLUT254[5:1], GLUT254[5:0], BLUT254[5:1]
Base + 0x2fe		RLUT255[5:1], GLUT255[5:0], BLUT255[5:1]

Base: Reload table start address

“Base” in the table refers the reload table start address specified by RTBL_BADR[31:10] (same as the control table reload function).

The monochrome LUT data configuration is the same as the LCDC_MLUT0 and LCDC_MLUT1 registers.

In color mode, each 16-bit data in the reload table consists of R, G, and B entry data as shown below.

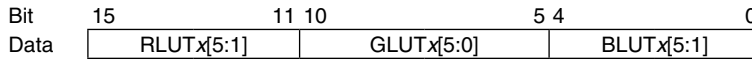


Figure 26.9.1 Color Look-Up Table Entry Data Format

Writing 1 to LUTRLD/LCDC_RLDCTL register resets the look-up tables with the reload table data. This reload operation should be performed during a vertical non-display period. LUTRLD retains 1 during reloading and it reverts to 0 when the reloading is completed.

If LUTRLD and CTABRLD are both set to 1 at the same time, the LCDC replace the control register data first, then LUT data.

In color mode, DSTRAM must be switched to LUTRAM before setting the look-up tables using the LUT reload function.

26.10 Control Register Details

Figure 26.10.1 List of LCDC Registers

Address	Register name	Function
0x302000	LCDC_INT	LCDC Interrupt Enable Register
0x302004	LCDC_PSAVE	Status and Power Save Configuration Register
0x302010	LCDC_HDISP	Horizontal Display Register
0x302014	LCDC_VDISP	Vertical Display Register
0x302018	LCDC_MODR	MOD Rate Register
0x302020	LCDC_HDPS	Horizontal Display Start Position Register
0x302024	LCDC_VDPS	Vertical Display Start Position Register
0x302028	LCDC_FPLINE	FPLINE Pulse Setting Register
0x30202c	LCDC_FPFR	FPFRAME Pulse Setting Register
0x302030	LCDC_FPFROFS	FPFRAME Pulse Offset Register
0x302040	LCDC_TFTSO	TFT Special Output Register
0x302044	LCDC_TFT_CTL1	TFT_CTL1 Pulse Register
0x302048	LCDC_TFT_CTL0	TFT_CTL0 Pulse Register
0x30204c	LCDC_TFT_CTL2	TFT_CTL2 Register
0x302050	LCDC_RLDCTL	LCDC Reload Control Register
0x302054	LCDC_RLDADR	LCDC Reload Table Base Address Register
0x302060	LCDC_DISPMOD	LCDC Display Mode Register
0x302070	LCDC_MAINADR	Main Window Display Start Address Register
0x302074	LCDC_MAINOFS	Main Screen Address Offset Register
0x302080	LCDC_SUBADR	Sub-window Display Start Address Register
0x302084	LCDC_SUBOFS	Sub-screen Address Offset Register
0x302088	LCDC_SUBSP	Sub-window Start Position Register
0x30208c	LCDC_SUBEP	Sub-window End Position Register
0x302090	LCDC_MLUT0	Monochrome Look-up Table Register 0
0x302094	LCDC_MLUT1	Monochrome Look-up Table Register 1

The LCDC registers are described in detail below. These are 32-bit registers.

Note: When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.

LCDC Interrupt Enable Register (LCDC_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC Interrupt Enable Register (LCDC_INT)	0x302000 (32 bits)	D31-1	–	reserved	–	–	–	0 when being read.
		D0	FRINTEN	Frame interrupt enable	1 Enable 0 Disable	0	R/W	

D[31:1] Reserved

D0 FRINTEN: Frame Interrupt Enable Bit

Enables or disables LCDC frame interrupts.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When using the frame interrupt, set FRINTEN to 1. The frame interrupt requests to the ITC is enabled.

When this bit is set to 0, the frame interrupt will not be generated.

Status and Power Save Configuration Register (LCDC_PSAVE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Status and Power Save Configuration Register (LCDC_PSAVE)	0x302004 (32 bits)	D31	FRINTF	Frame interrupt flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.	
		D30-8	–	reserved	–	–	–	0 when being read.	
		D7	VNDPF	Vertical display status flag	1 VNDP 0 VDP	1	R		
		D6-2	–	reserved	–	–	–	–	0 when being read.
		D1-0	PSAVE[1:0]	Power save mode select	PSAVE[1:0] Mode	0x3 Normal 0x2 reserved 0x1 reserved 0x0 Power save	0x0	R/W	

D31 FRINTF: Frame Interrupt Flag Bit

Indicates the frame interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

FRINTF is set to 1 when a vertical non-display period begins. If FRINTEN/LCDC_INT register has been set to 1, a frame interrupt request is sent to the ITC at the same time.

D[30:8] Reserved

D7 VNDPF: Vertical Display Status Flag Bit

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.

D[6:2] Reserved

D[1:0] PSAVE[1:0]: Power Save Mode Select Bits

Selects the power-save mode.

Table 26.10.2 Power-Save Mode Settings

PSAVE[1:0]	Mode
0x3	Normal operation
0x2	Reserved
0x1	Reserved
0x0	Power-save mode

(Default: 0x0)

The LCD controller is placed in power-save mode by setting PSAVE[1:0] to 0x0. 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[1:0] to 0x3.

Horizontal Display Register (LCDC_HDISP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Horizontal Display Register (LCDC_HDISP)	0x302010 (32 bits)	D31–23	–	reserved	–	–	–	0 when being read.
		D22–16	HTCNT[6:0]	Horizontal total period (HT) setup HT = HDP + HNDP HT > HDPS + HDP (for HR-TFT)	HT = (HTCNT + 1) × 8 [Ts] HNDP = (HTCNT - HDPCNT) × 8 [Ts]	0x0	R/W	
		D15–7	–	reserved	–	–	–	0 when being read.
		D6–0	HDPCNT [6:0]	Horizontal display period (HDP) setup	HDP = (HDPCNT + 1) × 8 [Ts]	0x0	R/W	

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]:

$$HTCNT[6:0] \geq HDPCNT[6:0] + 3$$

$$HT > HDP + HDPS$$

Note: HT should be determined so that the horizontal non-display period (HNDP = HT - HDP) will be longer than the time required when the LCDC accesses eight words in the VRAM.

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]:

$$HDP \geq 16 \quad (HDPCNT[6:0] \geq 1)$$

Example: when 320 × 240 LCD (STN/TFT) panel is used

$$HDP = 320$$

$$HDPCNT[6:0] = 320/8 - 1 = 39 (= 0x27)$$

Vertical Display Register (LCDC_VDISP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vertical Display Register (LCDC_VDISP)	0x302014 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	VTCNT[9:0]	Vertical total period (VT) setup VT = VDP + VNDP VT > VDPS + VDP (for HR-TFT)	VT = VTCNT + 1 [lines] VNDP = VTCNT - VDPCNT [lines]	0x0	R/W	
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	VDPCNT [9:0]	Vertical display period (VDP) setup	VDP = VDPCNT + 1 [lines]	0x0	R/W	

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] VDPCNT[9:0]: Vertical Display Period (VDP) Setup Bits

Sets the vertical display period (VDP, panel vertical resolution) in line units. (Default: 0x0)

$$VDP = VDPCNT[9:0] + 1 \text{ [lines]}$$

The following condition must be satisfied when setting VDPCNT[9:0]:

$$VT \geq VDP + 1$$

Example: when 320 × 240 LCD (STN/TFT) panel is used

$$VDP = 240$$

$$VDPCNT[9:0] = 240 - 1 = 239 (= 0xef)$$

MOD Rate Register (LCDC_MODR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
MOD Rate Register (LCDC_MODR)	0x302018 (32 bits)	D31–6	–	reserved	–	–	–	0 when being read.
		D5–0	MOD[5:0]	LCD MOD rate setup	0x0 to 0x3f	0x0	R/W	

D[31:6] Reserved

D[5:0] MOD[5:0]: 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 FPPFRAME signal. If another period is desired, set the FPLINE pulse-count value.

Horizontal Display Start Position Register (LCDC_HDPS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Horizontal Display Start Position Register (LCDC_HDPS)	0x302020 (32 bits)	D31–10	–	reserved	–	–	–	0 when being read.
		D9–0	HDPSCNT[9:0]	Horizontal display period start position for TFT HT > HDP + HDPS + 1 (HR-TFT) HT > HDP + HDPS (other TFT)	HDPS = HDPSCNT [Ts]	0x0	R/W	0x0 must be set for STN panels.

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 Bits

Sets the horizontal display period start position (HDPS) for HR-TFT panels in pixel clock units. (Default: 0x0)

$$HDPS = HDPSCNT[9:0] + 1 \text{ [Ts]} \quad (\text{Ts: pixel clock period})$$

The following condition must be satisfied when setting HDPSCNT[9:0]:

$$HT > HDP + HDPS$$

Vertical Display Start Position Register (LCDC_VDPS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vertical Display Start Position Register (LCDC_VDPS)	0x302024 (32 bits)	D31–10	–	reserved	–	–	–	0 when being read.
		D9–0	VDPSCNT[9:0]	Vertical display period start position for TFT VT > VDP + VDPS	VDPS = VDPSCNT [lines]	0x0	R/W	0x0 must be set for STN panels.

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 Bits

Sets the vertical display period start position (VDPS) for HR-TFT panels in line units. (Default: 0x0)

$$VDPS = VDPSCNT[9:0] \text{ [lines]}$$

The following condition must be satisfied when setting VDPSCNT[9:0]:

$$VT > VDP + VDPS$$

FPLINE Pulse Setting Register (LCDC_FPLINE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPLINE Pulse Setting Register (LCDC_FPLINE)	0x302028 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	FPLINE_ST[9:0]	FPLINE pulse start position setup	Start position = FPLINE_ST + 1 [Ts]	0x0	R/W	*1: For TFT 0x0 must be set for STN panels.
		D15–8	–	reserved	–	–	–	0 when being read.
		D7	FPLINE_POL	FPLINE pulse polarity setup	1 Active high 0 Active low	0	R/W	(*1)
	D6–0	FPLINE_WD[6:0]	FPLINE pulse width setup	Pulse width = FPLINE_WD + 1 [Ts]	0x0	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[31:26] Reserved**D[25:16] FPLINE_ST[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)

$$HPS = FPLINE_ST[9:0] + 1 \text{ [Ts]} \quad (\text{Ts: pixel clock period})$$

D[15:8] Reserved**D7 FPLINE_POL: 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] FPLINE_WD[6:0]: FPLINE Pulse Width Setup Bits

Sets the horizontal sync pulse width (HPW) for HR-TFT panels in pixel clock units. (Default: 0x0)

$$HPW = FPLINE_WD[6:0] + 1 \text{ [Ts]} \quad (\text{Ts: pixel clock period})$$

FPFRAME Pulse Setting Register (LCDC_FPFR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPFRAME Pulse Setting Register (LCDC_FPFR)	0x30202c (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	FPFRAME_ST[9:0]	FPFRAME pulse start position setup	Start position = FPFRAME_ST × HT [Ts]	0x0	R/W	*1: For TFT 0x0 must be set for STN panels.
		D15–8	–	reserved	–	–	–	0 when being read.
		D7	FPFRAME_POL	FPFRAME pulse polarity setup	1 Active high 0 Active low	0	R/W	(*1)
	D6–0	FPFRAME_WD[6:0]	FPFRAME pulse width setup	Pulse width = (FPFRAME_WD+1) × HT [Ts]	0x0	R/W	(*1)	

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] FPFRAME_ST[9:0]: FPFRAME Pulse Start Position Setup Bits**

Sets the vertical sync pulse (FPFRAME or SPS) start position (VPS) for HR-TFT panels. (Default: 0x0)

$$VPS = FPFRAME_ST[9:0] \text{ [lines]} = FPFRAME_ST[9:0] \times HT \text{ [Ts]} \quad (\text{Ts: pixel clock period})$$

D[15:8] Reserved

D7 FPFFRAME_POL: 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[6:0] FPFFRAME_WD[6:0]: FPFFRAME Pulse Width Setup Bits

Sets the vertical sync pulse width (VPW) for HR-TFT panels. (Default: 0x0)

$$VPW = FPFFRAME_WD[6:0] + 1 \text{ [lines]} = (FPFFRAME_WD[6:0] + 1) \times HT \text{ [Ts]}$$

(Ts: pixel clock period)

FPFFRAME Pulse Offset Register (LCDC_FPFROFS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPFFRAME Pulse Offset Register (LCDC_FPFROFS)	0x302030 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	FPFFRAME_STPOFS[9:0]	FPFFRAME pulse stop offset	Stop offset = FPFFRAME_STPOFS [Ts]	0x0	R/W	*1: For TFT 0x0 must be set for STN panels.
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	FPFFRAME_STOFS[9:0]	FPFFRAME pulse start offset	Start offset = FPFFRAME_STOFS [Ts]	0x0	R/W	(*1)

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] FPFFRAME_STPOFS[9:0]: FPFFRAME 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)

$$VPW' = (FPFFRAME_WD[6:0] + 1) \times HT - FPFFRAME_STOFS[9:0] + FPFFRAME_STPOFS[9:0] \text{ [Ts]}$$

(Ts: pixel clock period)

D[15:10] Reserved**D[9:0] FPFFRAME_STOFS[9:0]: FPFFRAME Pulse Start Offset Bits**

Adjusts the vertical sync pulse start position, which has been set in line units, in pixel clock units. (Default: 0x0)

$$VPS' = FPFFRAME_ST[9:0] \times HT + FPFFRAME_STOFS[9:0] \text{ [Ts]}$$

(Ts: pixel clock period)

TFT Special Output Register (LCDC_TFTSO)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TFT Special Output Register (LCDC_TFTSO)	0x302040 (32 bits)	D31–4	–	reserved	–	–	–	0 when being read.
		D3	CTL1CTL	TFT_CTL1 control	1 Program 0 Toggle/line	0	R/W	For TFT
		D2	CTLCNT_RUN	TFT_CTL0–2 control counter run/stop	1 Run 0 Stop	0	R/W	0x0 must be set for STN panels.
		D1	FPSHIFT_POL	FPSHIFT polarity	1 Falling 0 Rising	0	R/W	
		D0	CTL01SWAP	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 LCDC_TFT_CTL1 register 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 CTLCNT_RUN: TFT_CTL0–2 Control Counter Run/Stop Bit

Starts and stops the TFT_CTL0–2 control counters.

1 (R/W): Run

0 (R/W): Stop (default)

The LCDC has a built-in counters to control the TFT_CTL0–2 signal output timings. Setting CTLCNT_RUN to 1 starts the counters to generate the TFT_CTL0 (PS), TFT_CTL1 (CLS), and TFT_CTL2 (REV) signals programmed by the application. Be sure to set this bit to 1 when using the TFT_CTL0–2 signals.

When CTLCNT_RUN = 0, the counters stop counting. When the TFT_CTL0–2 signals are not used, set this bit to 0 to reduce power consumption.

D1 FPSHIFT_POL: FPSHIFT Polarity Bit

Selects the polarity of the FPSHIFT (DCLK) signal for HR-TFT panels.

1 (R/W): Falling edge

0 (R/W): Rising edge (default)

When FPSHIFT_POL is set to 1, the FPDAT[23:0] output signal toggles at the rising edge (sampled at the falling edge) of the FPSHIFT (DCLK) signal. When FPSHIFT_POL is set to 0, the FPDAT[23:0] output signal toggles at the falling edge (sampled at the rising edge) of the FPSHIFT (DCLK) signal.

D0 CTL01SWAP: 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)

TFT_CTL1 Pulse Register (LCDC_TFT_CTL1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TFT_CTL1 Pulse Register (LCDC_TFT_CTL1)	0x302044 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	CTL1STP [9:0]	TFT_CTL1 pulse stop offset TFT_CTL1 pulse width = (CTL1STP - CTL1ST + 1) Ts	Stop offset = CTL1STP + 1 [Ts]	0x0	R/W	*2: For TFT This register is enabled when CTLCNT_RUN = 1.
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	CTL1ST [9:0]	TFT_CTL1 pulse start offset	Start offset = CTL1ST [Ts]	0x0	R/W	(*2)

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 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 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/LCDC_TFTSO register and CTLCNT_RUN/LCDC_TFTSO register to 1.

TFT_CTL0 Pulse Register (LCDC_TFT_CTL0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TFT_CTL0 Pulse Register (LCDC_TFT_ CTL0)	0x302048 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	CTL0STP [9:0]	TFT_CTL0 pulse stop offset TFT_CTL0 pulse width = (CTL0STP - CTL0ST + 1) Ts	Stop offset = CTL0STP + 1 [Ts]	0x0	R/W	*2: For TFT This register is enabled when CTLCNT_RUN = 1.
		D15–10 D9–0	– CTL0ST [9:0]	reserved TFT_CTL0 pulse start offset	– Start offset = CTL0ST [Ts]	– 0x0	– R/W	0 when being read. (*2)

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 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 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 CTLCNT_RUN/LCDC_TFTSO register to 1.

TFT_CTL2 Register (LCDC_TFT_CTL2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TFT_CTL2 Register (LCDC_TFT_ CTL2)	0x30204c (32 bits)	D31–10	–	reserved	–	–	–	0 when being read.
		D9–0	CTL2DLY [9:0]	TFT_CTL2 delay setup	Delay = CTL2DLY [Ts]	0x0	R/W	*2: For TFT This register is enabled when CTLCNT_RUN = 1.

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 CTLCNT_RUN/LCDC_TFTSO register to 1.

LCDC Reload Control Register (LCDC_RLDCTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC Reload Control Register (LCDC_ RLDCTL)	0x302050 (32 bits)	D31–2	–	reserved	–	–	–	0 when being read.
		D1	LUTRLD	LUT reload trigger	1 Trigger	0 Ignored	0	W
					1 Reloading	0 Finished		R
		D0	CTABRLD	Control table reload trigger	1 Trigger	0 Ignored	0	W
1 Reloading	0 Finished				R			

D[31:2] Reserved

D1 LUTRLD: LUT Reload Trigger Bit

Replaces the look-up table values by the reload table data.

1 (W): Trigger to reload

0 (W): Ignored

1 (R): Reloading

0 (R): Reloading has finished (default)

26 LCD CONTROLLER (LCDC)

Writing 1 to LUTRLD resets the look-up table entries with the reload table data. This reload operation should be performed during a vertical non-display period. LUTRLD retains 1 during reloading and it reverts to 0 when the reloading is completed.

The LUT reload function is effective when the look-up table function is enabled (LUTPASS/LCDC_DISPMD register = 0).

In color mode, DSTRAM must be switched to LUTRAM before setting the look-up tables using the LUT reload function.

If LUTRLD and CTABRLD are both set to 1 at the same time, the LCDC replace the control register data first, then LUT data.

See Section 26.9 for the reload table contents.

D0 CTABRLD: Control Table Reload Trigger Bit

Replaces the LCDC control registers by the reload table data.

1 (W): Trigger to reload

0 (W): Ignored

1 (R): Reloading

0 (R): Reloading has finished (default)

Writing 1 to CTABRLD resets the control registers with the reload table data. This reload operation should be performed during a vertical non-display period. CTABRLD retains 1 during reloading and it reverts to 0 when the reloading is completed.

See Section 26.9 for the reload table contents.

LCDC Reload Table Base Address Register (LCDC_RLDADR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC Reload Table Base Address Register (LCDC_RLDADR)	0x302054 (32 bits)	D31–10	RTBL_ BADR[31:10]	Reload table base address (1KB boundary address, A[9:0] = 0x0)	Areas 3*–5, 7–10, 13–16, and 19–22	0x0	R/W	* DSTRAM cannot be used.
		D9–0	–	reserved	–	–	–	0 when being read.

D[31:10] RTBL_BADR[31:10]: Reload Table Base Address Bits

Specifies the reload table base address. (Default: 0x0)

A 1K-byte boundary address in IVRAM or an external memory must be specified.

Note: DSTRAM in Area 3 cannot be used to locate the reload table. For more information, see “Bus Masters and Accessible Memories” in the “Memory Map” chapter.

D[9:0] Reserved

LCDC Display Mode Register (LCDC_DISPMD)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
LCDC Display Mode Register (LCDC_DISPMD)	0x302060 (32 bits)	D31	PANSELSEL	Panel type select	1	TFT	0 STN	0	R/W		
		D30	COLOR	Color/mono select	1	Color	0 Mono	0	R/W		
		D29	FPSHIFT_MSK	FPSHIFT mask enable	1	Enable	0 Disable	0	R/W		
		D28	–	reserved				–	–	0 when being read.	
		D27–26	DWD[1:0]	LCD panel data width select		DWD[1:0]	Data width		0x0	R/W	
						0x3	8 bits (fmt2)				
						0x2	reserved				
						0x1	8 bits (fmt1)				
						0x0	4 bits				
		D25	SWINV	Software video invert	1	Invert	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	Repeat	0 Not repeat	0	R/W		
D6–5	–	reserved				–	–	0 when being read.			
D4	LUTPASS	LUT bypass mode select	1	Bypass	0 Use	1	R/W				
D3	–	reserved				–	–	0 when being read.			
D2–0	BPP[2:0]	Bit-per-pixel select		BPP[2:0]	bpp		0x0	R/W			
				0x7	reserved						
				0x6	24 bpp						
				0x5	16 bpp						
				0x4	12 bpp						
				0x3	8 bpp						
				0x2	4 bpp						
				0x1	2 bpp						
				0x0	1 bpp						

D31 **PANSELSEL: Panel Type Select Bit**

Selects the type of connected LCD panel (STN or TFT).

1 (R/W): TFT panel

0 (R/W): STN panel (default)

When TFT panel is selected, COLOR and DWD[1:0] settings are disabled.

D30 **COLOR: Color/Mono Select Bit**

Selects the type of connected LCD panel (color or monochrome).

1 (R/W): Color panel

0 (R/W): Monochrome panel (default)

D29 **FPSHIFT_MSK: FPSHIFT Mask Enable Bit**

Enables the FPSHIFT mask (effective only for STN monochrome LCD panels and HR-TFT panels).

1 (R/W): Enabled

0 (R/W): Disabled (default)

When FPSHIFT_MSK is set to 1, the FPSHIFT signal is masked and is not output during the non-display period. When FPSHIFT_MSK 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 STN LCD panel's data width and format.

Table 26.10.3 Data Width Selection of STN Panels

COLOR	DWD[1:0]	Data width
1	0x3	Color single 8-bit passive LCD panel (format 2)
	0x2	Reserved
	0x1	Color single 8-bit passive LCD panel (format 1)
	0x0	Color single 4-bit passive LCD panel
0	0x3	Monochrome single 8-bit passive LCD panel
	0x2	Reserved
	0x1	Monochrome single 8-bit passive LCD panel
	0x0	Monochrome single 4-bit passive LCD panel

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 the LCDC output, and does not affect the display memory.

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.

D[6:5] Reserved**D4 LUTPASS: LUT Bypass Mode Select Bit**

Selects whether the look-up table is bypassed.

1 (R/W): Bypassed (default)

0 (R/W): Used

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 color mode, DSTRAM must be switched to LUTRAM before the color look-up tables can be used.

In 12-, 16-, and 24-bpp color mode, this bit must be set to 1 as the look-up table cannot be used.

D3 Reserved**D[2:0] BPP[2:0]: Bit-Per-Pixel Select Bits**

Selects the bpp mode.

Table 26.10.4 Specification of Bpp Mode

BPP[2:0]	bpp mode
0x7	Reserved
0x6	24 bpp
0x5	16 bpp
0x4	12 bpp
0x3	8 bpp
0x2	4 bpp
0x1	2 bpp
0x0	1 bpp

(Default: 0x0)

Main Window Display Start Address Register (LCDC_MAINADR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Main Window Display Start Address Register (LCDC_MAINADR)	0x302070 (32 bits)	D31–0	MW_START [31:0]	Main window start address MW_START31 = MSB MW_START0 = LSB	0x0 to 0xfffffc (Areas 3–5, 7–10, 13–16, and 19–22)	0x0	R/W	

D[31:0] MW_START[31:0]: Main Window Start Address Bits

Sets the main window display start address. (Default: 0x0)

Note that a word boundary address (A[1:0] = 0b00) in the IVRAM or external VRAM must be specified to this register.

Main Screen Address Offset Register (LCDC_MAINOFS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Main Screen Address Offset Register (LCDC_MAINOFS)	0x302074 (32 bits)	D31–12	–	reserved	–	–	–	0 when being read.
		D11–0	MW_OFS [11:0]	Main screen address offset	Main screen width (pixels) × bpp/32	0x0	R/W	

D[31:12] Reserved

D[11:0] MW_OFS[11:0]: Main Screen Address Offset Bits

Sets the main virtual screen width in words. (Default: 0x0)

The set value is calculated as follows:

$$\text{MW_OFS}[11:0] = \text{virtual screen width in pixels} \times \text{bpp} / 32$$

See “Main screen address offset for virtual screen” in Section 26.6.2 for more information on the virtual screen and the configurations.

Sub-window Display Start Address Register (LCDC_SUBADR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Sub-window Display Start Address Register (LCDC_SUBADR)	0x302080 (32 bits)	D31–0	SW_START [31:0]	Sub-window start address SW_START31 = MSB SW_START0 = LSB	0x0 to 0xfffffc (Areas 3–5, 7–10, 13–16, and 19–22)	0x0	R/W	

D[31:0] SW_START[31:0]: Sub-Window Start Address Bits

Sets the sub-window display start address. (Default: 0x0)

Note that a word boundary address (A[1:0] = 0b00) in the IVRAM or external VRAM must be specified to this register.

Sub-Screen Address Offset Register (LCDC_SUBOFS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Sub-screen Address Offset Register (LCDC_SUBOFS)	0x302084 (32 bits)	D31–12	–	reserved	–	–	–	0 when being read.
		D11–0	SW_OFS [11:0]	Sub-screen address offset	Sub-screen width (pixels) × bpp/32	0x0	R/W	

D[31:12] Reserved

D[11:0] SW_OFS[11:0]: Sub-Screen Address Offset Bits

Sets the sub-virtual screen width in words. (Default: 0x0)

The set value is calculated as follows:

$$\text{SW_OFS}[11:0] = \text{virtual screen width in pixels} \times \text{bpp} / 32$$

See “Main screen address offset for virtual screen” in Section 26.6.2 for more information on the virtual screen and the configurations.

Sub-Window Start Position Register (LCDC_SUBSP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Sub-window Start Position Register (LCDC_SUBSP)	0x302088 (32 bits)	D31	PIP_EN	PIP enable	1 Enable 0 Disable	0	R/W	
		D30–26	–	reserved	–	–	–	0 when being read.
		D25–16	PIP_YSTART[9:0]	Sub-window vertical (Y) start position	Y start position = PIP_YSTART (lines) from the origin	0x0	R/W	*3: This register is enabled when PIP_EN = 1.
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	PIP_XSTART[9:0]	Sub-window horizontal (X) start position	X start position = PIP_XSTART (pixels) from the origin (word units)	0x0	R/W	(*3)

D31 PIP_EN: PIP Enable Bit

Enables the Picture-in-Picture Plus function to display the sub-window in the main window.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Configure the sub-window using the registers at 0x302080 to 0x30208c before setting PIP_EN to 1.

D[30:26] Reserved

D[25:16] PIP_YSTART[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.

$$\text{PIP_YSTART}[9:0] = Y_{\text{START}} [\text{lines}]$$

For example, to specify the sub-window vertical start position as 60 lines, set PIP_YSTART[9:0] to 60.

D[15:10] Reserved

D[9:0] PIP_XSTART[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.

$$\text{PIP_XSTART}[9:0] = X_{\text{START}} \text{ pixels} \times \text{bpp} \div 32 [\text{words}]$$

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

24-bpp mode: 3-word = 4-pixel units (because the value must be an integer)

For example, to specify the sub-window horizontal start position as 80 pixels in 8-bpp mode, set PIP_XSTART[9:0] to 20.

Sub-Window End Position Register (LCDC_SUBEP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Sub-window End Position Register (LCDC_SUBEP)	0x30208c (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	PIP_YEND[9:0]	Sub-window vertical (Y) end position	Y end position = PIP_YEND (lines) from the origin	0x0	R/W	*3: This register is enabled when PIP_EN = 1.
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	PIP_XEND[9:0]	Sub-window horizontal (X) end position	X end position = PIP_XEND (pixels) from the origin (word units)	0x0	R/W	(*3)

D[31:26] Reserved

D[25:16] PIP_YEND[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.

$$\text{PIP_YEND}[9:0] = Y_{\text{END}} - 1 \text{ [lines]}$$

D[15:10] Reserved**D[9:0] PIP_XEND[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.

$$\text{PIP_XEND}[9:0] = X_{\text{END}} \text{ pixels} \times \text{bpp} \div 32 - 1 \text{ [words]}$$

Monochrome Look-up Table Registers 0 and 1 (LCDC_MLUT0/1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Monochrome Look-up Table Register 0 (LCDC_MLUT0)	0x302090 (32 bits)	D31–28	MLUT7[3:0]	Monochrome LUT entry 7 data	0x0 to 0xf	0x0	R/W	
		D27–24	MLUT6[3:0]	Monochrome LUT entry 6 data	0x0 to 0xf	0x0	R/W	
		D23–20	MLUT5[3:0]	Monochrome LUT entry 5 data	0x0 to 0xf	0x0	R/W	
		D19–16	MLUT4[3:0]	Monochrome LUT entry 4 data	0x0 to 0xf	0x0	R/W	
		D15–12	MLUT3[3:0]	Monochrome LUT entry 3 data	0x0 to 0xf	0x0	R/W	
		D11–8	MLUT2[3:0]	Monochrome LUT entry 2 data	0x0 to 0xf	0x0	R/W	
		D7–4	MLUT1[3:0]	Monochrome LUT entry 1 data	0x0 to 0xf	0x0	R/W	
		D3–0	MLUT0[3:0]	Monochrome LUT entry 0 data	0x0 to 0xf	0x0	R/W	
Monochrome Look-up Table Register 1 (LCDC_MLUT1)	0x302094 (32 bits)	D31–28	MLUT15[3:0]	Monochrome LUT entry 15 data	0x0 to 0xf	0x0	R/W	
		D27–24	MLUT14[3:0]	Monochrome LUT entry 14 data	0x0 to 0xf	0x0	R/W	
		D23–20	MLUT13[3:0]	Monochrome LUT entry 13 data	0x0 to 0xf	0x0	R/W	
		D19–16	MLUT12[3:0]	Monochrome LUT entry 12 data	0x0 to 0xf	0x0	R/W	
		D15–12	MLUT11[3:0]	Monochrome LUT entry 11 data	0x0 to 0xf	0x0	R/W	
		D11–8	MLUT10[3:0]	Monochrome LUT entry 10 data	0x0 to 0xf	0x0	R/W	
		D7–4	MLUT9[3:0]	Monochrome LUT entry 9 data	0x0 to 0xf	0x0	R/W	
		D3–0	MLUT8[3:0]	Monochrome LUT entry 8 data	0x0 to 0xf	0x0	R/W	

These registers are used to set data to the monochrome look-up table entries. The entries used depend on the bpp mode.

1-bpp mode: Entries 0 and 1 are used.

2-bpp mode: Entries 0 to 3 are used.

4-bpp mode: All entries are used.

D[31:28], D[27:24], D[23:20], D[19:16], D[15:12], D[11:8], D[7:4], D[3:0]**MLUT n [3:0]: Monochrome LUT Entry n Data Bits**

Sets the 4-bit data for the monochrome look-up table entry n . (Default: 0x0)

27 Graphics Engine (GE)

27.1 GE Module Overview

The S1C33L26 includes a graphics engine (GE) for drawing basic objects (e.g., straight lines, rectangles, circles) and bitmap texts with a specified font on a VRAM. The GE provides drawing commands in which drawing effects such as clipping, line width setting, octant selection for circle drawing, block copy, XOR/mesh/transparency, color replacement/color inversion with palettes, and resize/tiling/rotation in 90-degree angle (texts/compressed image) can be specified as well as shape and color. The GE executes the commands programmed in the memory sequentially to draw pictures on the VRAM. The generated image is displayed on the LCD panel via the LCDC. The following shows the main features of the GE:

- Drawing functions
 - Basic objects: points, straight lines (vertical, horizontal, and sloped lines), rectangles, and circles
 - Filled objects: triangles, rectangles, quadrilaterals, and circles
 - Texts with a font specified
 - Compressed image (original run-length encoding)
- Drawing effects
 - Clipping
 - Line width setting
 - Drawing color setting
 - Octant selection for circle drawing
 - Block copy
 - XOR/mesh/transparency
 - Color replacement/color inversion with palettes
 - Resize/tiling/rotation in 90-degree angle (texts/compressed image)
- Command set
 - Drawing functions can be specified by a 32-bit command + one to six 32-bits arguments.
 - Number of commands: 18 commands in all
 - The image to be displayed can be programmed by writing commands sequentially in the memory.
 - The command execution start address can be specified using a register.
 - The drawing sequence stops when a STOP command appears.
- Other
 - Supports a maximum $4,096 \times 4,096$ pixels of work area (drawing area).
 - Supports 1 bpp (2 colors/2 gray scale levels) to 16 bpp (65,536 colors) modes.
(Note: GE cannot be used when the LCDC is set to 12/24 bpp mode.)
 - Color depth conversion for image data using color conversion table (CCT)
 - VRAM write direction can be specified in 90-degree angle (supporting LCD panel rotation).
 - Can generate end-of-execution, drawing error, and calculation error interrupts.

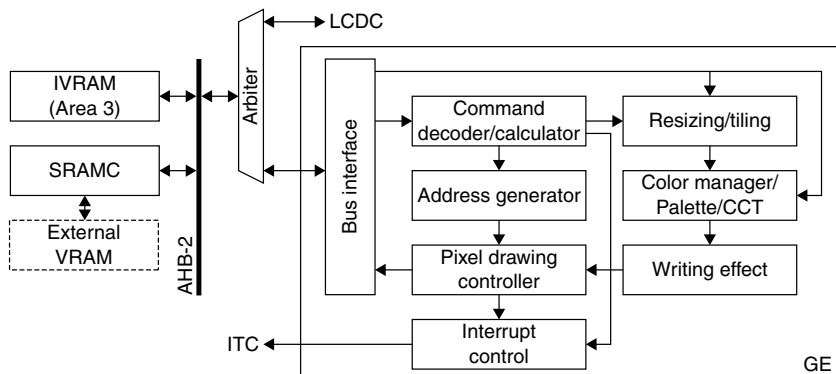


Figure 27.1.1 GE Module Configuration

27.2 Operating Clock

The GE module operates with the GCLK clock supplied from the CMU. For details on controlling the clocks, see the “Clock Management Unit (CMU)” chapter.

27.3 Drawing Functions

This section describes basic functions to draw objects and drawing effects individually. For details on the commands to execute the functions, see Section 27.5.

27.3.1 Drawing Area

Coordinate system and work area

The GE module supports a maximum 4,096 pixels × 4,096 pixels of drawing area of which the upper left corner is the origin point (X, Y = 0, 0) of the coordinate system. Note, however, that the physical area for drawing depends on the VRAM capacity.

Effective range of the coordinate system: (X, Y) = (0, 0) to (4095, 4095)

A work area to which objects/images can actually be drawn (VRAM is used normally) must be configured within this range in advance.

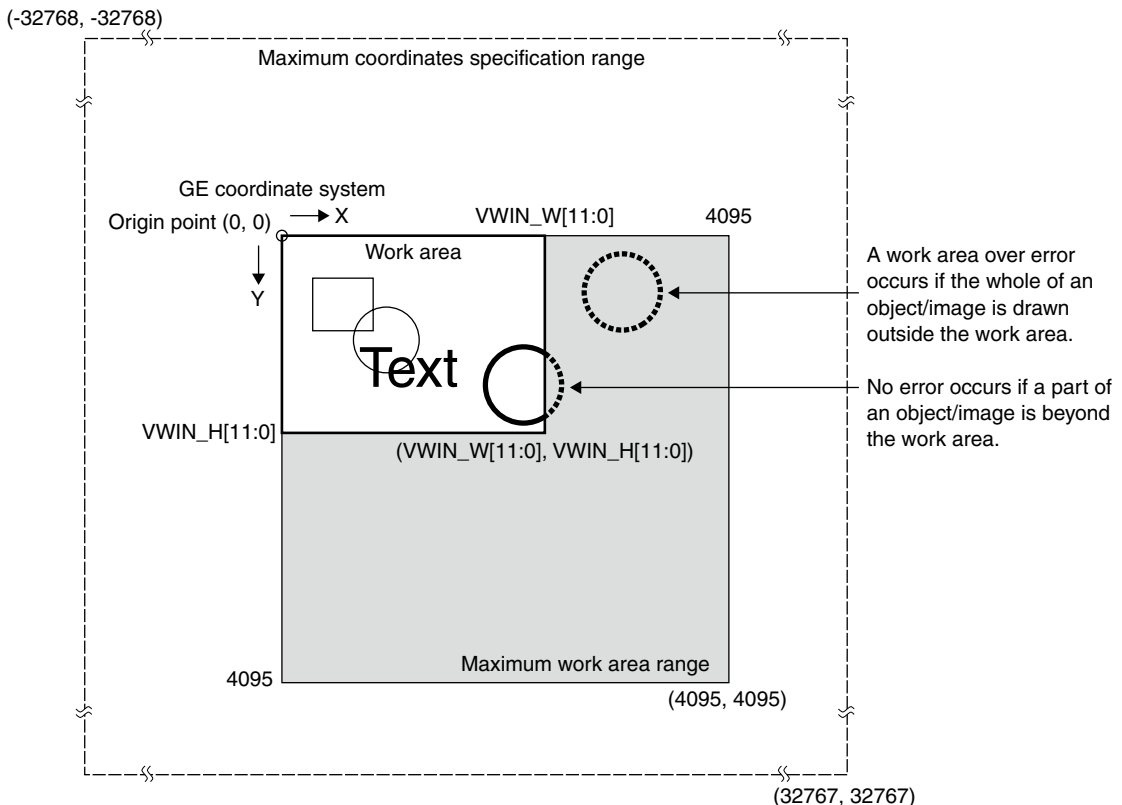


Figure 27.3.1.1 Work Area

The work area is configured with coordinate values specified using VWIN_W[11:0]/GE_WK_SIZE register for width and VWIN_H[11:0]/GE_WK_SIZE register for height.

Work area width: VWIN_W[11:0] + 1 (pixels)

Work area height: VWIN_H[11:0] + 1 (pixels)

Effective coordinate values: (0, 0) to (VWIN_W[11:0], VWIN_H[11:0])

The work area cannot be configured beyond 4,096 pixels × 4,096 pixels even if an external large capacity VRAM is connected.

If the whole of an object/image drawn by a command is located outside the work area (coordinates are specified in signed 16-bit values), a work area over error occurs (an interrupt can be generated). If only a part of an object/image is beyond the work area, the drawing within the work area is performed without an error.

Note: If the X or Y coordinate in the object, which is generated by a drawing command, exceeds the range from -32,768 to 32,767, the GE module may not draw the object properly.

The table below lists VRAM capacities required for different work area size and bpp mode (color depth).

Table 27.3.1.1 Work Area Size and VRAM Capacity

Color depth	VRAM capacity
16 bpp (65,536 colors)	Number of all pixels ÷ 2 (words)
8 bpp (256 colors)	Number of all pixels ÷ 4 (words)
4 bpp (16 colors/16-level gray scale)	Number of all pixels ÷ 8 (words)
2 bpp (4 colors/4-level gray scale)	Number of all pixels ÷ 16 (words)
1 bpp (2 colors/2-level gray scale)	Number of all pixels ÷ 32 (words)

(Number of all pixels = Horizontal pixel count × Vertical pixel count of work area)

Note: The work area width (data size) must be specified in word (32 bits) increments. If an incorrect value is specified, the GE corrects it to the word (32 bits) boundary value larger and closest to the specified value.

The data size depends on the bpp mode. Furthermore, when the work area is rotated, its width and height must be reset according to the rotation angle. Therefore, VWIN_W[11:0] and VWIN_H[11:0] should be set after setting the bpp mode and the angle of work area rotation. See Table 27.3.1.4 for VWIN_W[11:0] and VWIN_H[11:0] settings for different bpp modes and rotation angles.

Settings for VRAM address calculation

In order for the GE module to write objects/images specified with coordinate values to the appropriate VRAM location, set the information required for converting coordinates within the work area into VRAM addresses as below.

Work area start address

Set the VRAM address corresponding to the origin point (0, 0) of the work area to VWIN_ADDR[31:2]/GE_WK_ADDR register.

Note: Set a 1K-byte boundary address for the work area start address.

Color depth

The correspondence between coordinates and VRAM addresses/bits depends on the color depth (bits per pixel). Set the bpp mode using DISP_BPP[2:0]/GE_DISP_CFG register.

Table 27.3.1.2 Color Depth Settings

DISP_BPP[2:0]	Color depth
0x7 to 0x5	Reserved
0x4	16 bpp (65,536 colors)
0x3	8 bpp (256 colors)
0x2	4 bpp (16 colors/16-level gray scale)
0x1	2 bpp (4 colors/4-level gray scale)
0x0	1 bpp (2 colors/2-level gray scale)

(Default: 0x0)

Relationship between LCD display and work area

The embedded LCDC module supports a virtual screen function to use an image area larger than the LCD panel size. Any location in the virtual screen area can be displayed on the LCD panel. This allows multiple-screens to be configured in the VRAM and panning/scrolling large images.

Normally the work area configured in the GE corresponds to the LCDC virtual screen area. Also a memory area other than the LCDC VRAM can be allocated for the work area, since the GE supports block transfer between the work area and other memory areas (including a built-in RAM LCD driver).

For correspondence between GE and LCDC settings, see Section 27.7.

VRAM rotation

Some applications require rotating display image according to LCD panel rotation. In order to use the same drawing commands regardless of whether the LCD panel is rotated or not, GE is able to write object/image data as if the VRAM is rotated. The rotation angle can be selected from 0°, 90°, 180°, and 270° as shown in the figures below. Note, however, that the display width and height must be configured according to the LCD panel orientation as well as the rotation angle.

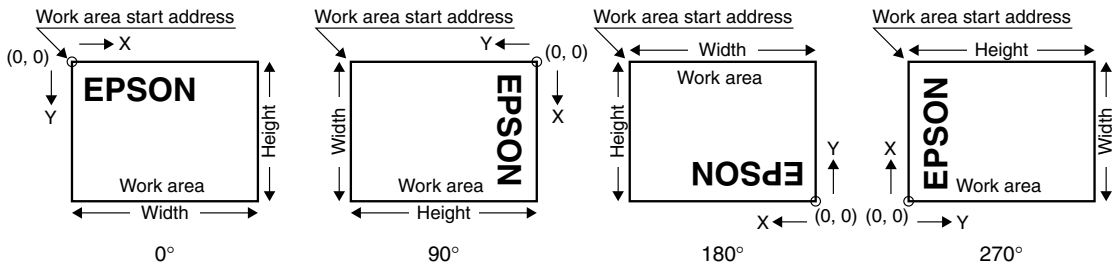


Figure 27.3.1.2 VRAM Rotation

The rotation angle is selected using VWIN_ROT[1:0]/GE_ROTATE register.

Table 27.3.1.3 Angle of VRAM Rotation

VWIN_ROT[1:0]	Rotation angle
0x3	270°
0x2	180°
0x1	90°
0x0	0°

(Default: 0x0)

Changing this setting does not affect the contents of the work area. To rotate the image according to the LCD panel, configure the work area and execute the drawing commands again.

Table 27.3.1.4 lists the VWIN_W[11:0] and VWIN_H[11:0] settings by angle of VRAM rotation and bpp mode.

Table 27.3.1.4 Work Area Size Specifications

bpp mode	VWIN_W[11:0]	VWIN_H[11:0]
1 bpp	$\text{int}((\text{Width} + 31) / 32) \times 32 - 1$	Height - 1
2 bpp	$\text{int}((\text{Width} + 15) / 16) \times 16 - 1$	Height - 1
4 bpp	$\text{int}((\text{Width} + 7) / 8) \times 8 - 1$	Height - 1
8 bpp	$\text{int}((\text{Width} + 3) / 4) \times 4 - 1$	Height - 1
16 bpp	$\text{int}((\text{Width} + 1) / 2) \times 2 - 1$	Height - 1

The Width and Height mean the maximum logical coordinate values in Figure 27.3.1.2.

Clipping area

Each drawing command can enable/disable the clipping function as a drawing effect. When a command with clipping enabled is executed, the object/image specified by the command is drawn only within the clipping area that has been configured in advance. Drawing to outside the clipping area is masked even if it is within the work area.

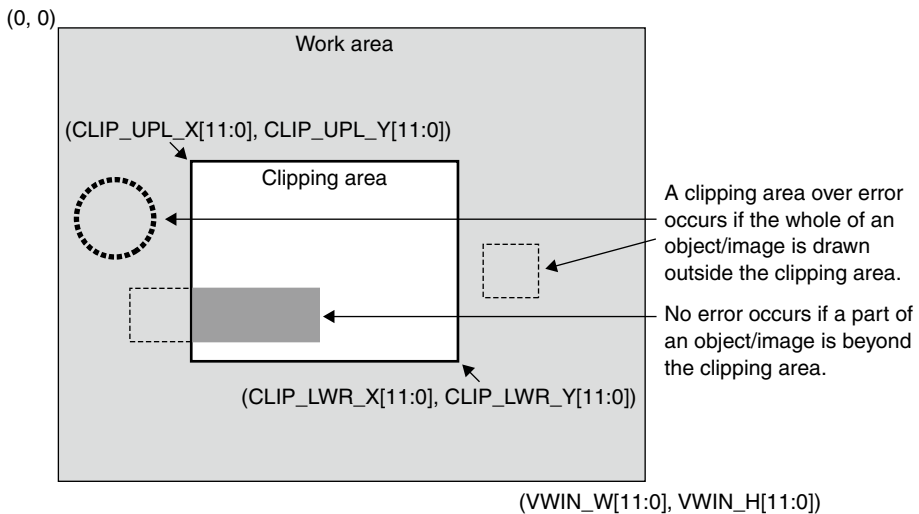


Figure 27.3.1.3 Clipping Area

To configure a clipping area, write the coordinate values of the upper left and lower right corners to the registers shown below.

X coordinate of the upper left corner (0 to 4,095): CLIP_UPL_X[11:0]/GE_CLIP_ST register

Y coordinate of the upper left corner (0 to 4,095): CLIP_UPL_Y[11:0]/GE_CLIP_ST register

X coordinate of the lower right corner (0 to 4,095): CLIP_LWR_X[11:0]/GE_CLIP_END register

Y coordinate of the lower right corner (0 to 4,095): CLIP_LWR_Y[11:0]/GE_CLIP_END register

Configuring a clipping area using these registers does not enable the clipping function. Enable it in the drawing commands to perform clipping.

If the whole of an object/image drawn by a command is located outside the clipping area when the clipping function is enabled, a clipping area over error occurs (an interrupt can be generated). If only a part of an object/image is beyond the clipping area, the drawing within the clipping area is performed without an error.

- Notes:**
- When configuring a clipping area, the X and Y coordinates of the upper left corner must be set first. Otherwise, proper operations cannot be guaranteed.
 - Do not configure a clipping area that exceeds the work area, as proper drawing results cannot be guaranteed.
 - When the GE_CONFIG command is used to set the GE_CLIP_ST register, the setting value is not written to the register until the GE_CONFIG command for setting the GE_CLIP_END register is executed.

27.3.2 Drawing Basic Objects

This section introduces the objects that can be drawn using the drawing commands. The drawing commands also allow drawing effect specifications such as clipping and transparency (see Section 27.3.5).

Dot

Dot drawing / DOT command (0x10)

Fills the pixel at the X and Y coordinates specified with the specified color.

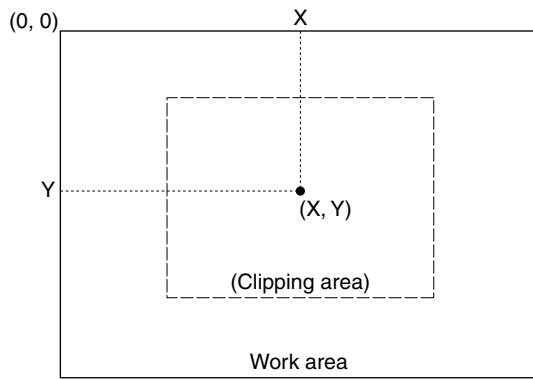


Figure 27.3.2.1 Dot Drawing

Straight line

Straight line drawing / LINE command (0x11)

Draws a line between the two points specified in X and Y coordinates with the specified color. The line width can also be specified in number of pixels. Not only horizontal and vertical lines but also oblique lines can be drawn according to the specified coordinates.

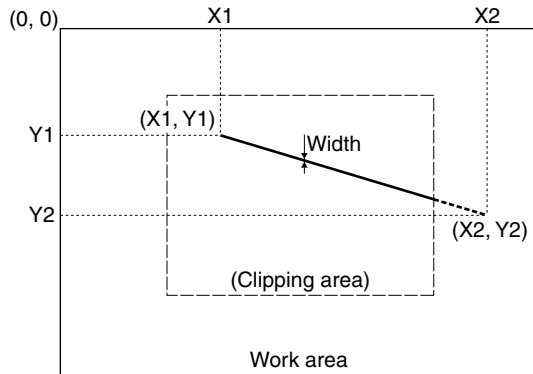


Figure 27.3.2.2 Straight Line Drawing

Oblique lines with two pixels or more line width are drawn as follows:

- (1) When $315^\circ < \text{slope} < 45^\circ$ ($135^\circ < \text{slope} < 225^\circ$): Both ends are drawn vertically.
- (2) When $45^\circ \leq \text{slope} \leq 135^\circ$ ($225^\circ \leq \text{slope} \leq 315^\circ$): Both ends are drawn horizontally.

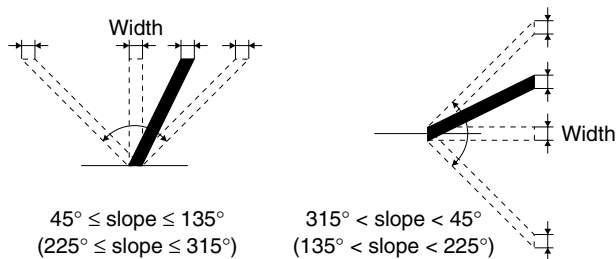


Figure 27.3.2.3 Drawing Oblique Line

When an odd number is specified as the line width, the X and Y coordinates specified becomes the center of the line width. When an even number is specified as the line width, the left or upper side from the specified axis will be thicker by one pixel than the right or lower side.

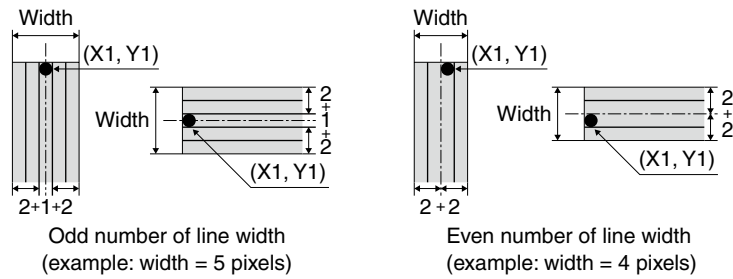


Figure 27.3.2.4 Drawing Line with Even Number of Line Width

Rectangle/quadrilateral/triangle

Rectangle drawing / RECT command (0x12)

Draws a rectangle with the colored line specified. Specify the X and Y coordinates of the upper left and lower right corners. The line width can also be specified in number of pixels.

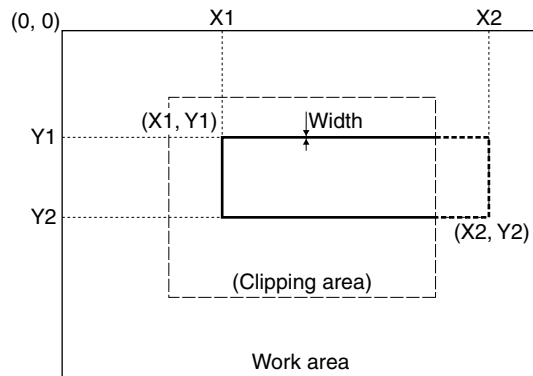


Figure 27.3.2.5 Drawing Rectangular Line

The line drawing method for an even number of line width is the same as that of straight line drawing.

Solid filled rectangle drawing / RECT_FILL command (0x17)

Draws a solid filled rectangle specified with the X and Y coordinates of the upper left and lower right corners, and a color.

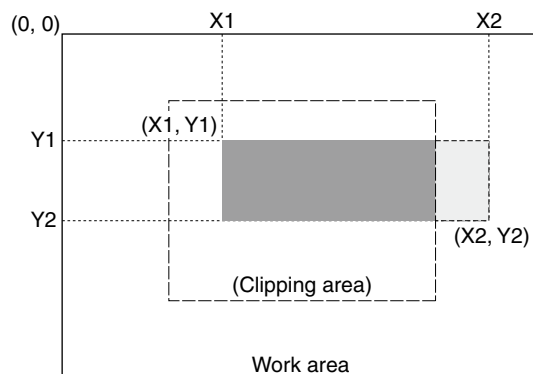


Figure 27.3.2.6 Drawing Solid Filled Rectangle

Solid filled quadrilateral drawing / QUAD_FILL command (0x18)

Draws a solid filled quadrilateral specified with the X and Y coordinates of the four vertices, and a color.

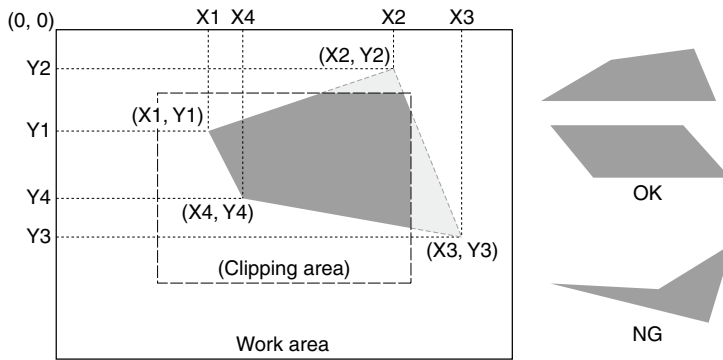


Figure 27.3.2.7 Drawing Solid Filled Quadrilateral

This command supports convex quadrilaterals only. An error results if a concave quadrilateral is specified when calculation error interrupts are enabled. The X and Y coordinates of four vertices must be specified in a clockwise direction.

Solid filled triangle drawing / TRI_FILL command (0x16)

Draws a solid filled triangle specified with the X and Y coordinates of the three vertices, and a color.

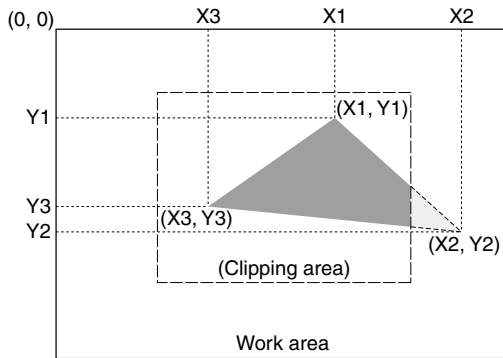


Figure 27.3.2.8 Drawing Solid Filled Triangle

The X and Y coordinates of three vertices must be specified in a clockwise direction.

Circle

Circle drawing / CIRCLE command (0x1b)

Draws the circumference of a circle specified with the center coordinates, radius, and a line color. The line width can be specified in number of pixels. Furthermore, this command allows selection of the arcs to be drawn from the eight segments split by 45° (multiple segments can be selected).

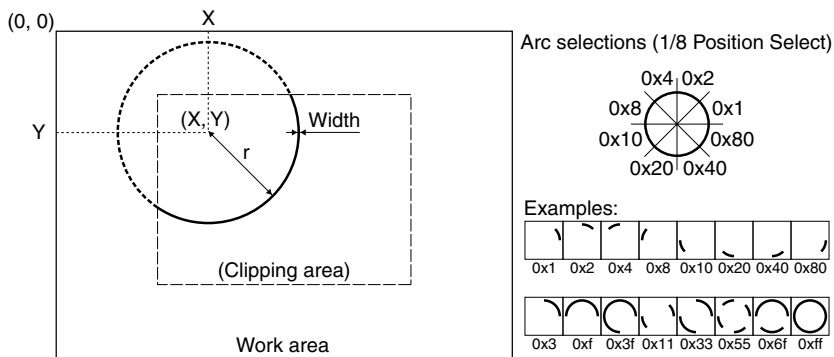


Figure 27.3.2.9 Circle Drawing

Solid filled circle drawing / CIRCLE_FILL command (0x1c)

Draws a solid filled circle specified with the center coordinates, radius, and a color. Furthermore, this command allows selection of the circular sections to be drawn from the eight portions split by 45° (multiple portions can be selected).

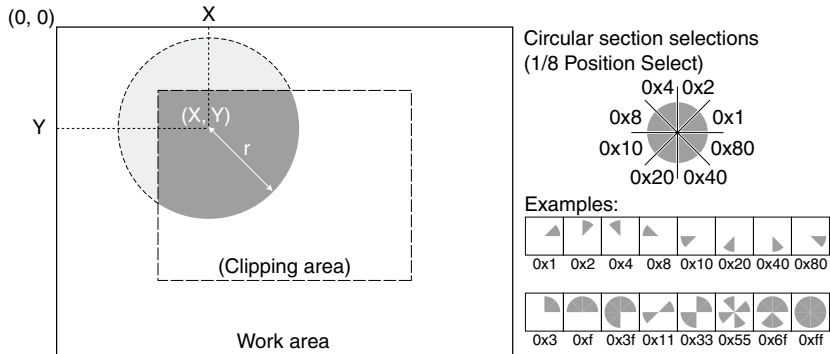


Figure 27.3.2.10 Drawing Solid Filled Circle

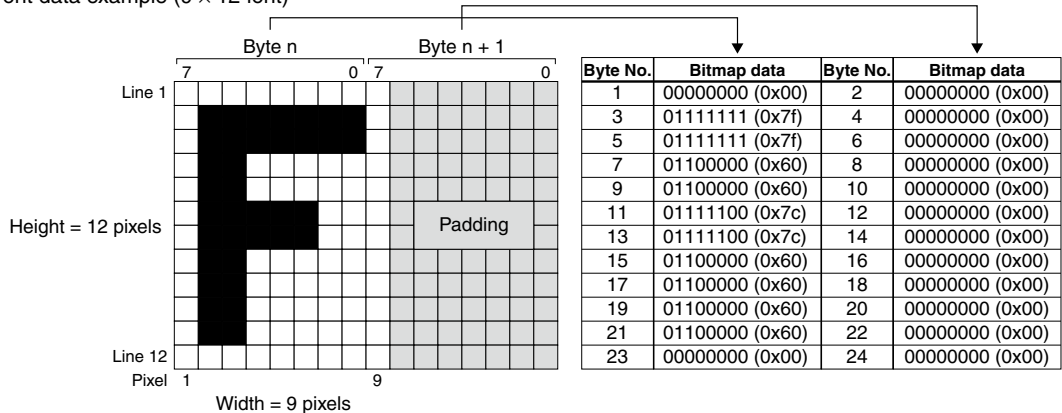
27.3.3 Drawing Characters and Symbols

By loading bitmap font data in the memory, characters and symbols can be drawn on the work area.

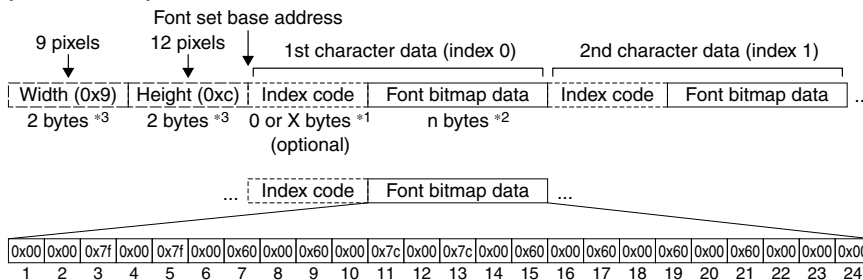
Font configuration / FONT_CFG command (0x20)

Sets the font information, such as the memory location where the font data (character set) is stored, character and background colors, and font bitmap size, to the GE module. This setting must be performed before starting character drawing.

Font data example (9 × 12 font)



Font data layout in memory



*1: X (index code size in bytes) must be specified as the Font Index Offset parameter in the font configuration command.
 *2: $n = \text{int}\left(\frac{\text{Width} + 7}{8}\right) \times \text{Height}$ [bytes] (24 bytes in this example)
 *3: Font base address is the start address of the first character data. The font header is not started from the font base address.

Figure 27.3.3.1 Font Data Format

The GE allows the user to add user configured index data at the beginning of each character data. In order for the GE to skip the indexes, the index size must be specified in an argument of the font configuration command.

Note: Font data must be located in IVRAM (Area 3) or an external memory located in Areas 13 to 22. IRAM (Area 0) cannot be used to execute command lists.

Character drawing / CHAR command (0x21)

Draws a character or symbol specified by an index (sequential numbers starting from 0 for the 1st character) in the character set at the location specified with the coordinates. The command also allows specification of drawing effects such as resizing, tiling, and rotation as well as general drawing effects.

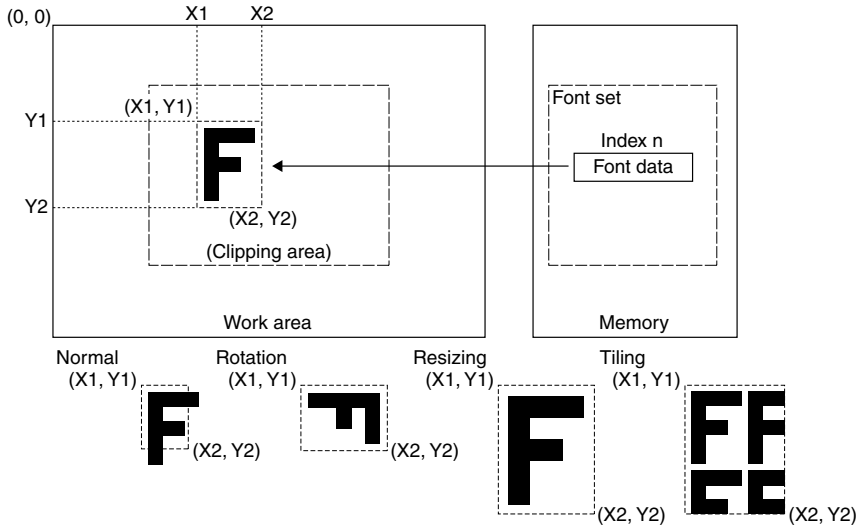


Figure 27.3.3.2 Character Drawing

27.3.4 Decompression/Copy Functions

The GE module supports functions to decompress compressed image data, copy and transfer image data.

Decompression / DECOMP command (0x22)

Decompresses the compressed image data (Run length encoded data) located at the specified memory address and draws the image within the area specified with the X and Y coordinates of the upper left and lower right corners in the work area. The GE also supports uncompressed data. The command allows specification of drawing effects such as resizing, tiling, and rotation as well as general drawing effects.

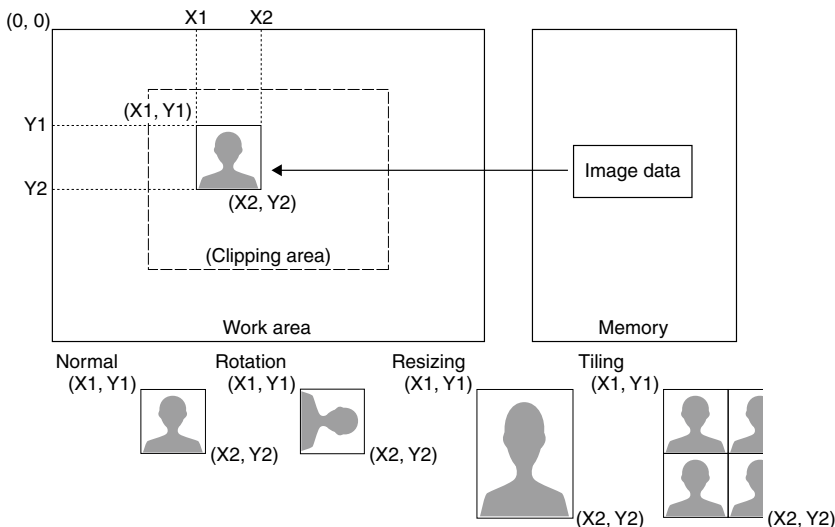
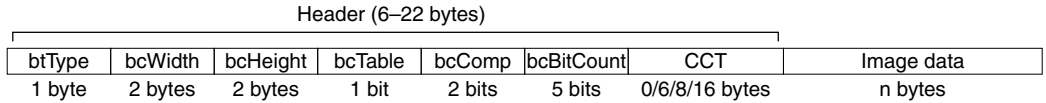


Figure 27.3.4.1 Decompression and Drawing

Image data format

The GE module supports PackBits format compressed data (MSB first) and uncompressed data only. The figure below shows the image data format.



- btType: Fixed at 0x80
- bcWidth: Image width (number of pixels)
- bcHeight: Image height (number of pixels)
- bcTable: CCT enable (1: Enable, 0: Disable)
- bcComp: Compression type

bcComp	Type
Other	Reserved
0b01	PackBits
0b00	Uncompressed

bcBitCount: Color depth

bcBitCount	Color depth
0b10000	16 bpp
0b01000	8 bpp
0b00100	4 bpp
0b00010	2 bpp
0b00001	1 bpp

CCT: Color conversion table (not included when bcTable = 0)
 4 bpp image data: 16-byte 4 to 8 bpp conversion data
 2 bpp image data: 8-byte 2 to 8 bpp and 2 to 4 bpp conversion data
 1 bpp image data: 6-byte 1 to 8 bpp, 1 to 4 bpp, and 1 to 2 bpp conversion data

Color depth (size)	CCT (0/6/8/16 bytes)															
	Byte															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4 bpp (16 bytes)	4 to 8 bpp table															
2 bpp (8 bytes)	Image data ...								2 to 4 bpp table				2 to 8 bpp table			
1 bpp (6 bytes)	Image data ...						1 to 2 bpp table				1 to 4 bpp table		1 to 8 bpp table			
None (-)	Image data ...															

Figure 27.3.4.2 Image Data Format

Note: Image data must be located in IVRAM (Area 3) or an external memory located in Areas 13 to 22. IRAM (Area 0) cannot be used to execute command lists.

Automatic color depth (bpp mode) conversion

If the image data has a color depth (1/2/4 bpp) different from the bpp mode set in the GE module, the color data is automatically converted according to the GE bpp mode before the image is written to the work area (color reduction is not supported).

The GE module uses one of the two color conversion tables (CCT and CCT1) for this conversion.

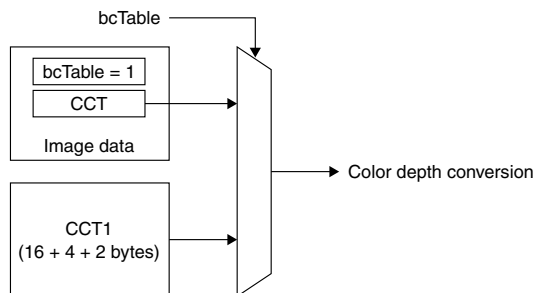


Figure 27.3.4.3 Configuration of Color Conversion Tables

When the color conversion table in the header (see Figure 27.3.4.2) of the image data to be decompressed/drawn is enabled (bcTable = 1), the CCT contents are used for color depth conversion. Although the header in 2 bpp and 1 bpp image data includes two or more conversion data, only the data for converting into the GE bpp mode is used.

Table 27.3.4.1 CCT1 Configuration

Table address	4 bpp data (4 to 8 bpp conversion data)	2 bpp data (2 to 8 or 2 to 4 bpp conversion data)	1 bpp data (1 to 8, 1 to 4 or 1 to 2 bpp conversion data)
0x302910	8-bit color data 0		
0x302911	8-bit color data 1		
0x302912	8-bit color data 2		
0x302913	8-bit color data 3		
0x302914	8-bit color data 4		
0x302915	8-bit color data 5		
0x302916	8-bit color data 6		
0x302917	8-bit color data 7		
0x302918	8-bit color data 8		
0x302919	8-bit color data 9		
0x30291a	8-bit color data 10		
0x30291b	8-bit color data 11		
0x30291c	8-bit color data 12		
0x30291d	8-bit color data 13		
0x30291e	8-bit color data 14		
0x30291f	8-bit color data 15		
0x302920		8 or 4-bit color data 0	
0x302921		8 or 4-bit color data 1	
0x302922		8 or 4-bit color data 2	
0x302923		8 or 4-bit color data 3	
0x302924			8, 4, or 2-bit color data 0
0x302925			8, 4, or 2-bit color data 1

CCT1 is located at 22 byte from address 0x302910. This table is used for converting color depth when the color conversion table in the header of the image data to be decompressed/drawn is disabled (bcTable = 0). Conversion data must be written to the addresses shown above in advance.

Converting into 16 bpp colors

When the GE is set to 16 bpp mode, color depth is converted as follows:

- If the original data is made for 8 bpp mode, the GE draws the image with the source color on 16 bpp directly.
- If the original data is made for 1/2/4 bpp mode, it is converted into 8 bpp by the color depth conversion using CCT or CCT1 and then the 8 to 16 bpp conversion shown below is performed.

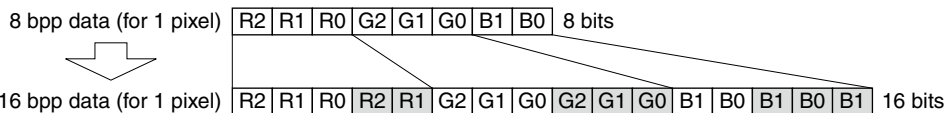


Figure 27.3.4.4 8 to 16 bpp Conversion

Note: The color depth conversion function does not support color reduction. Therefore, GE cannot draw image data made for a higher bpp mode than that of the GE.

8 and 16 bpp image data do not support color depth conversion and are extended without color conversion. The bcTable bit is ignored and the data header cannot contain CCT. An unexpected drawing will result if CCT exists in 8 or 16 bpp image data .

Table 27.3.4.2 Color Depth Conversion

		Image color depth				
		16 bpp	8 bpp	4 bpp	2 bpp	1 bpp
GE color depth	16 bpp	Not converted	8 to 16 bpp conversion	CCT used (8 bits × 16) + 8 to 16 bpp conversion	CCT used (8 bits × 4) + 8 to 16 bpp conversion	CCT used (8 bits × 2) + 8 to 16 bpp conversion
	8 bpp	–	Not converted	CCT used (8 bits × 16)	CCT used (8 bits × 4)	CCT used (8 bits × 2)
	4 bpp	–	–	Not converted	CCT used (8 bits × 4)	CCT used (8 bits × 2)
	2 bpp	–	–	–	Not converted	CCT used (8 bits × 2)
	1 bpp	–	–	–	–	Not converted

–: Cannot be drawn. CCT: CCT in the file header or CCT1

Copy / COPY command (0x29)

Copies the rectangular area specified with the X and Y coordinates of the upper left and lower right corners to the destination area specified within the work area. Drawing effects such as clipping and transparency that apply to the destination can also be specified.

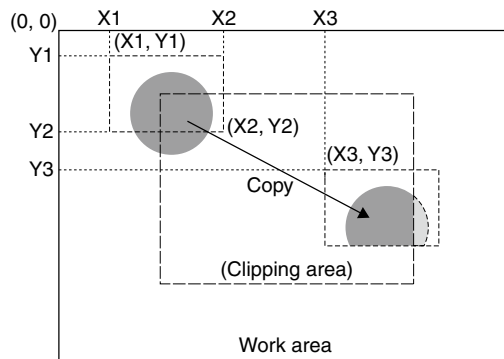


Figure 27.3.4.5 Image Copy

This copy function can guarantee that the source image will be copied to the destination properly even if the source and destination areas overlap one another.

Block transfer / BLKCOPY command (0x2a)

Transfers image data block between a specified area within the work area and memory or a built-in RAM LCD driver (via USIL). The source and destination can be selected from four combinations in a command argument.

1. Memory^{*1} → Work area
2. Work area → Memory^{*1}
3. Work area → USIL^{*2} → Built-in RAM LCD driver/panel (little endian)
4. Work area → USIL^{*2} → Built-in RAM LCD driver/panel (big endian)

*1: IVRAM (Area 3) or an external memory in Areas 13 to 22

*2: LCD SPI mode or LCD parallel I/F mode

The USIL transmit data buffer address should be specified as the destination.

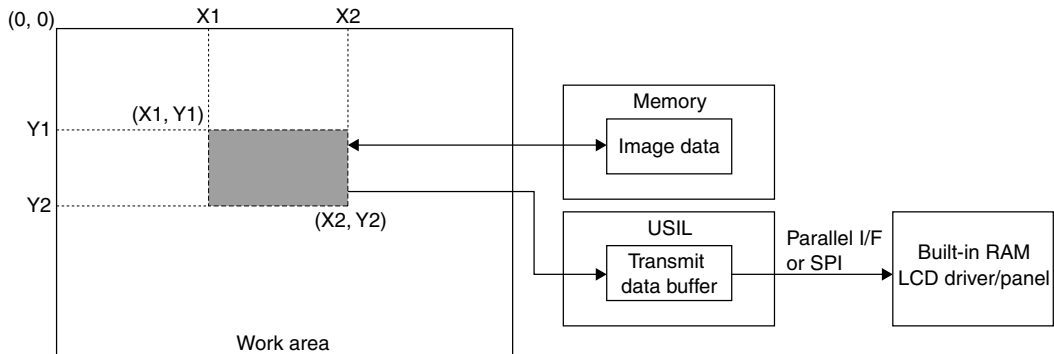


Figure 27.3.4.6 Image Data Block Transfer

To support various built-in RAM LCD panels via the USIL module, TF_TYPE/GE_DISP_CFG register is provided to configure the transfer data type. For more information on the data types, see the GE_DISP_CFG register description in the “Control Register Details” section.

27.3.5 Drawing Effects

A command argument (Effect Setting) is provided to specify drawing effects to apply when the GE module executes a drawing command to draw an object/image on the work area.

Table 27.3.5.1 lists the effects supported by the commands.

Table 27.3.5.1 Drawing Effects Supported by Commands

Drawing command	Drawing effect (Effect Setting parameters)											
	Clipping (Clip Enable)	Transparency (Transparent Enable)	Sync with LCD display (Sync Enable)	Writing effect (Write Effect Setting)				Palette conversion (Palette Select)	Arc/circular section (1/8 Position Select)	Resizing/tiling (Resize/Tile Select)	Rotation (Rotation Select)	Transfer source/destination (Memory Area Select)
				Normal (Fill)	XOR	Mesh	Rewrite					
Dot	○	○	○	○	○	○	○	○	-	-	-	-
Straight line	○	○	○	○	○	○	○	○	-	-	-	-
Rectangle	○	○	○	○	○	○	○	○	-	-	-	-
Solid filled triangle	○	○	○	○	○	○	○	○	-	-	-	-
Solid filled rectangle	○	○	○	○	○	○	○	○	-	-	-	-
Solid filled quadrilateral	○	○	○	○	○	○	○	○	-	-	-	-
Circle	○	○	○	○	○	○	○	○	○	-	-	-
Solid filled circle	○	○	○	○	○	○	○	○	○	-	-	-
Character	○	○	○	○	○	○	-	○	-	○	○	-
Decompression	○	○	○	○	○	○	-	○	-	○	○	-
Copy	○	○	○	○	○	○	-	○	-	-	-	-
Block transfer	-	○*	○*	○	○	○*	-	○	-	-	-	○

○: Can be specified. -: Cannot be specified.

*: The block transfer command supports these effects only when data is transferred from a memory to VRAM. These effect cannot be used when data is transferred from VRAM to a memory.

The following describes the drawing effects individually.

Clipping

The clipping function can be enabled or disabled.

Clip Enable bit = 1: Clipping enabled

Clip Enable bit = 0: Clipping disabled

When the clipping function is enabled, the GE draws only within the clipping area (see Section 27.3.1) configured with the GE_CLIP_ST and GE_CLIP_END registers. When disabled, the GE can draw across the whole work area.

Transparency

Transparency can be enabled or disabled.

Transparent Enable bit = 1: Transparency enabled

Transparent Enable bit = 0: Transparency disabled

When transparency is enabled, the pixels with the transparent color specified in objects/images are not drawn on the work area and the current pixel color is left unchanged.

The transparent color must be set to MAGIC_COL[15:0]/GE_MAGIC register in advance. A color within the effective range for the bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register) can only be specified.

Note that transparent color comparison is performed with the pixel color after being converted via CCT.

Sync with LCD display

The LCDC synchronization function can be enabled or disabled.

Sync Enable bit = 1: Synchronized

Sync Enable bit = 0: Not synchronized

When this function is enabled (synchronous mode), the GE starts drawing objects/images (writing to the work area) in sync with the LCD non-display period start timings and drawings are performed in the non-display periods. This makes it possible to eliminate screen flicker. When rewriting the area currently displayed on the LCD panel, normally enable the LCDC synchronization function.

Before using this function, select either vertical non-display period or horizontal non-display period as the rewrite period using SYNC_TYPE/GE_DISP_CFG register. Horizontal non-display period is selected when SYNC_TYPE is 0 (default) and vertical non-display period is selected when it is set to 1.

Note, however, that drawing time may exceed the non-display period if it is complicated.

When this function is disabled (asynchronous mode), the GE draws objects/images without a delay. Asynchronous mode is useful when drawing objects/images on an area out of the LCD display range or high-speed processing is required.

Writing effects

When the GE writes object/image data to the work area, a writing effect can be applied to the data to be written. The GE provides four effects listed in the table below and one of them can be selected.

Table 27.3.5.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

The writing effect is applied to the data immediately before being written to the work area after other drawing effects have been applied.

Normal (Fill) (Write Effect Setting[2:0] bits = 0x0)

No writing effect is applied. Image data is written to the work area without being modified.

XOR (Write Effect Setting[2:0] bits = 0x1)

Performs an XOR operation on the data to be written to and the current data on the work area, and writes the results to the work area. If the pixels located at the same position have the same color, the pixel is drawn in black.

Mesh (Write Effect Setting[2:0] bits = 0x2)

Overlays a mesh pattern on the object when written to the work area. This function can be used to produce a halftone and to highlight display contents.

The mesh pattern is configured using MESH_RW[1:0]/GE_MESH register for specifying the horizontal line width and MESH_CW[1:0]/GE_MESH register for specifying the vertical line width.

Table 27.3.5.3 Mesh Pattern Settings

MESH_RW[1:0]/MESH_CW[1:0]	Mesh size
0x3	4 pixels
0x2	Reserved
0x1	2 pixels
0x0	1 pixel

(Default: 0x0)

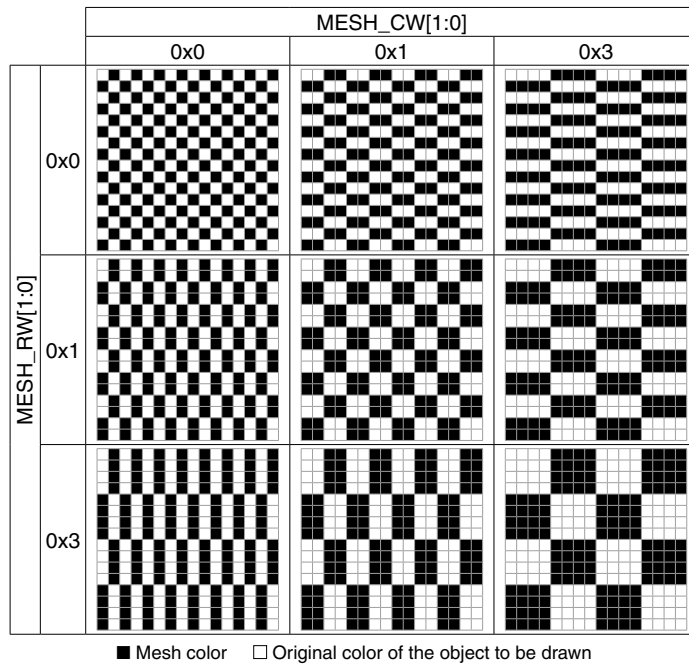


Figure 27.3.5.1 List of Mesh Patterns

The mesh color should be set to MESH_COL[15:0]/GE_MESH register. A color within the effective range for the bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register) can only be specified.

Note: Be sure to avoid setting both the mesh color and transparent color to the same. To produce a similar effect, specify the drawing color as the transparent color.

The mesh pattern is generated relative to the work area origin (0, 0). Therefore, a continuous mesh pattern will be generated even if two or more objects with mesh specified overlap one another.

Note that the work area origin (0, 0) is expressed by physical coordinates (before VRAM rotation), not logical coordinates (after VRAM rotation).

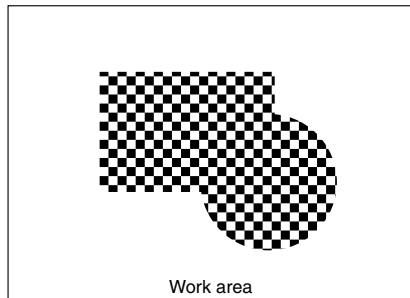


Figure 27.3.5.2 Mesh Effect Applied to Multiple Objects

Rewrite (Write Effect Setting[2:0] bits = 0x3)

Reads the work area data to be rewritten with the drawing command and performs color conversion via the palette selected by the Palette Select[1:0] bits. Then writes the converted data back to the work area. The palette only affects the color in the area to be rewritten.

The drawing color, which is specified in the color parameters in the command or determined by other drawing effects than palette, is ignored. However, if the color depth, which is ignored in the rewrite processing, exceeds the current bpp mode, a color depth over error will be generated.

Palette conversion

The GE includes three palettes for changing the color of drawing objects before writing to the work area. Set the Palette Select[1:0] bits to select the palette to be used. When the Palette Select[1:0] bits are set to 0x0, no palette is used and the original color data is written to the work area through the writing effect specified.

Table 27.3.5.4 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

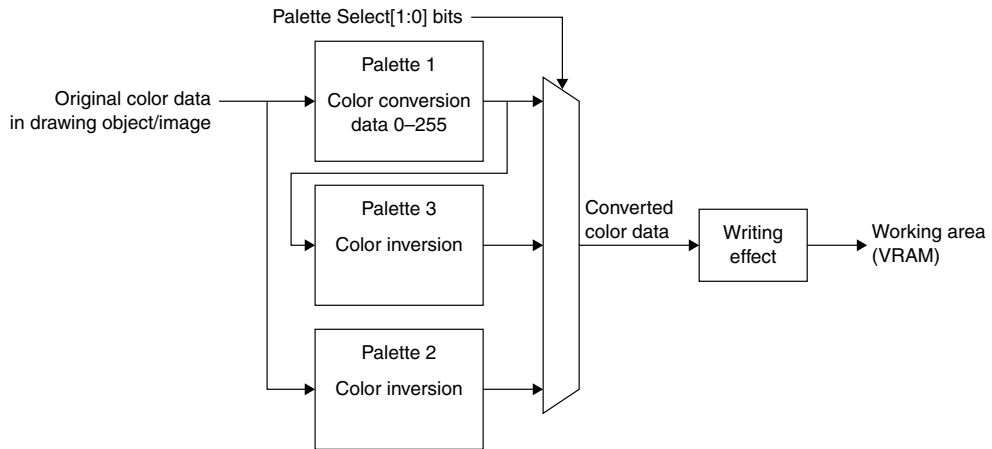


Figure 27.3.5.3 Palette Configuration

Palette 1

Specifying Palette 1 substitutes the color defined in the palette for the drawing color specified in the command or original color in the image data when drawing in 1/2/4/8 bpp mode.

Palette 1 is a 256-byte table and is located from address 0x302800 to 0x3028ff. The entries of 256 bytes correspond to original color data 0 to 255. Write the color data by which the original color will be substituted to each entry. 16 bpp mode does not use Palette 1 even if it is specified.

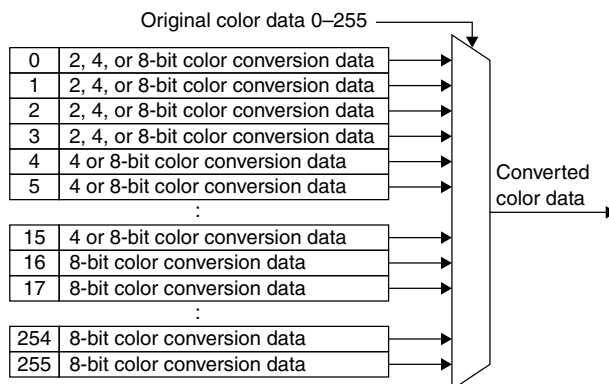


Figure 27.3.5.4 Palette 1 Configuration

Palette 2

Palette 2 is used to invert the specified drawing color or original colors in images.

Example: 8 bpp mode

0x00 → 0xff 0x0f → 0xf0 0x7f → 0x80

Palette 3

Palette 3 converts the drawing color using Palette 1 and then inverts the converted color data. The 16 bpp mode does not use Palette 3 even if it is specified.

Arc/Circular section

The circle command and the solid filled circle command allow selection of one or more arcs and circular sections to be drawn from eight locations split by 45° using the 1/8 Position Select[7:0] bits.

Set the 1/8 Position Select[7:0] bits corresponding to the arcs and circular sections to be drawn to 1 and execute the command. To draw a complete circle, set all the 1/8 Position Select[7:0] bits to 1.

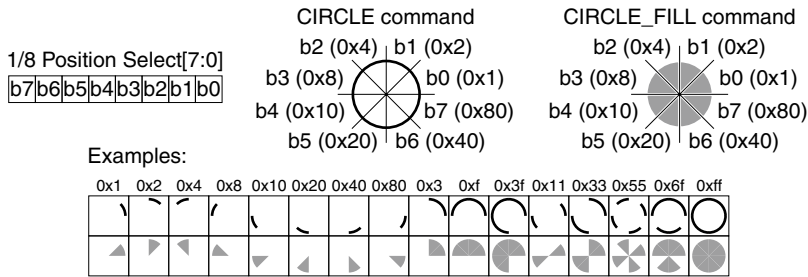


Figure 27.3.5.5 Arc/Circular Section Selections

Resizing/Tiling

The character and decompression commands allow specification of drawing areas different from the original size. The Resize/Tile Select[1:0] bits are used to select how the character/image is drawn in this area.

Table 27.3.5.5 Resizing/Tiling Selections

Resize/Tile Select[1:0] bits	Drawing effect
0x3	Reserved
0x2	Tiling
0x1	Resizing
0x0	Normal

Normal

The character/image is drawn from origin in the original size.

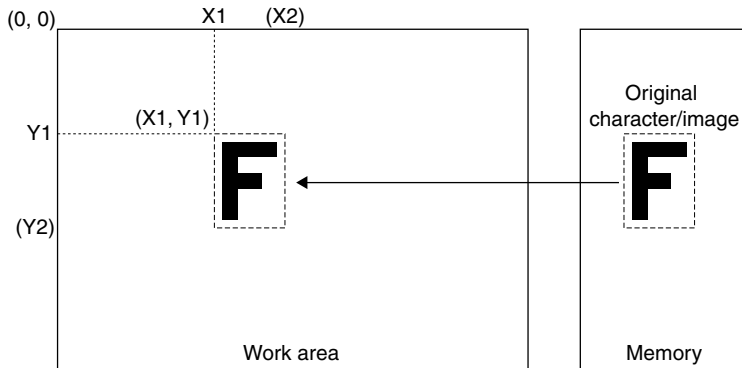


Figure 27.3.5.6 Drawing Character/Image in Normal Size

Resizing

Draws the character/image by resizing the original data according to the drawing area specified by the command. A drawing area with a different horizontal to vertical ratio can be specified.

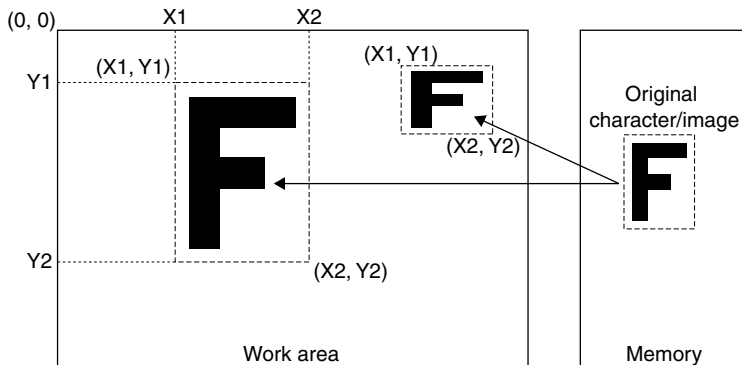


Figure 27.3.5.7 Resizing Characters/Images

Tiling

Draws the original size character/image to the specified drawing area with tiling. If a smaller drawing area than original size is specified, the part beyond the area is not drawn.

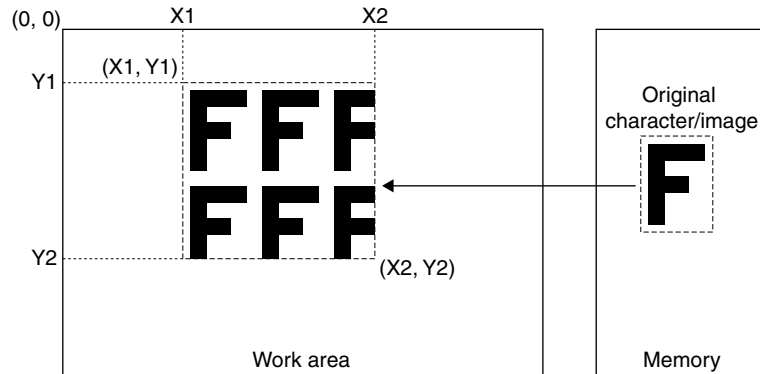


Figure 27.3.5.8 Tiling Characters/Images

Rotation

The character and decompression commands can rotate the character/image before drawing. The rotation angle can be selected using Rotation Select[1:0] bits.

Table 27.3.5.6 Angle of Character/Image Rotation

Rotation Select[1:0] bits	Rotation angle
0x3	270°
0x2	180°
0x1	90°
0x0	0°

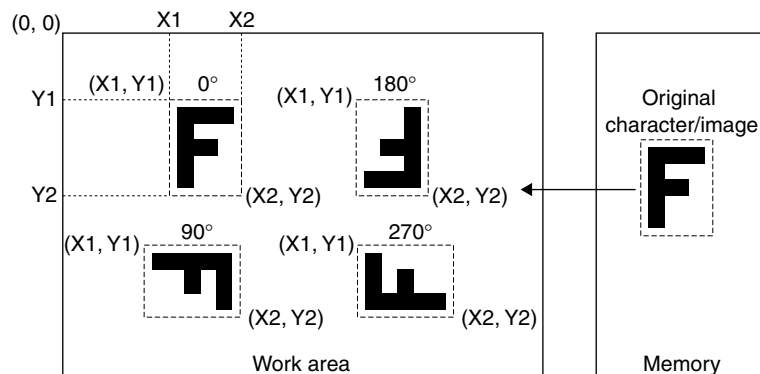


Figure 27.3.5.9 Character/Image Rotation

Transfer source/destination

The block transfer command allows selection of the source and destination for image data transfer using the Memory Select[1:0] bits. This command performs drawing data block transfer between the specified area in the work area and memory or a built-in RAM LCD driver (via USIL). The source and destination can be selected from four types in a command argument.

Table 27.3.5.7 Block Transfer Specification

Memory Select[1:0] bits	Source	Destination
0x3	Work area	USIL (big endian) *1
0x2		USIL (little endian) *1
0x1		Memory *2
0x0	Memory *2	Work area

*1: Data is transferred to the built-in RAM LCD driver/panel via USIL set to LCD SPI mode or LCD parallel I/F mode. Specify the USIL transmit data buffer address as the destination.

*2: IVRAM (Area 3) or an external memory located in Areas 13 to 22

27.4 Controlling GE

The GE sequentially executes the commands written to the memory to draw objects/images. The drawing area, mesh, and transparent color are set using the registers. To alter the registers while commands are being executed, a command to write data to the registers is provided.

This section explains initializing the GE and executing commands. For detailed information on the commands and registers, see Sections 27.5 and 27.6, respectively.

27.4.1 Initial Settings

The GE enters reset state after an initial reset. The operating clock is supplied from the CMU.

The following shows the procedure required to use the GE.

(1) Canceling GE reset state

Write 0 to GE_CRST/GE_CTL register to cancel GE cold reset status.

(2) Initializing the GE registers

Make the following settings using the GE registers.

1. Select a bpp mode and LCD sync condition. (GE_DISP_CFG register)
2. Select an angle of VRAM rotation. (GE_ROTATE register)
3. Set the work area. (GE_WK_ADDR and GE_WK_SIZE registers)
4. Set the clipping area. (GE_CLIP_ST and GE_CLIP_END registers)*
5. Select a mesh pattern. (GE_MESH register)*
6. Set the transparent color. (GE_MAGIC register)*
7. Set Palette 1 contents. (GE_PALETTE1 register)*
8. Sets CCT1 (color conversion table 1) contents. (GE_CCT1_4BIT, GE_CCT1_2BIT, and GE_CCT1_1BIT registers)*
9. Set the command start address. (GE_CMD_ADDR register)
10. Enable GE interrupts. (GE_IE register)

* It is not necessary to set these registers if the functions are not used.

(3) Programming drawing commands

Write the commands to be executed to the memory (beginning with the command start address). IVRAM (Area 3) or an external memory located in Areas 13 to 22 can be used for executing drawing commands. The commands written to IRAM (Area 0) cannot be executed.

Write the command for drawing the backmost object/image first. Commands and arguments must be written successively without a blank.

To alter the GE registers while commands are being executed, use the GE_CONFIG command.

Use a STOP command at the position to stop the command execution. The GE provides four STOP commands, STOP1 to STOP4. The application can confirm which STOP command is used after a command list has been executed (an interrupt has occurred) by reading the interrupt flag (EXE_END[2:0]/GE_IF3 register).

Note: Commands must begin from a word (32-bit) boundary address.

27.4.2 Command Execution and Termination

Starting execution

Writing 1 to GE_RUN/GE_CTL register starts command execution. The GE fetches the commands and arguments in 32-bit units beginning from the address set in the GE_CMD_ADDR register, and executes them successively. The GE_CMD_ADDR register contents are updated by fetching each command.

Terminating execution

Termination by a STOP command

The GE command stops executing a command list when it executes a STOP command (STOP1 to STOP4).

Suspending via software

Writing 1 to GE_STOP/GE_CTL register can forcibly stop executions. The GE stops executing after the current command being executed has finished.

Suspending by fetching an undefined command

The GE stops executing when it fetches an undefined command.

Termination by an unexpected operation

The GE stops executing a command list when it cannot continue for some reason. In this case, the GE must be reset (hot reset).

An interrupt can be generated upon completion of a command list execution. To identify the cause that has terminated command execution, read EXE_END[2:0]/GE_IF3 register in the interrupt handler routine. EXE_END[2:0] must be cleared by writing 0x0 or 0xff in byte to address 0x30244a.

Note that termination by a STOP command (EXE_END[2:0] = 0x1–0x4) may be issued after GE starts subsequent drawing. The STOP commands may not set GE into idle state, as the next command list is ready to run due to the GE pipeline process before executing a STOP command.

Table 27.4.2.1 Cause of Termination

EXE_END[2:0]	Cause of termination
0x7	Unexpected operation
0x6	Undefined command
0x5	Software
0x4	STOP4 command
0x3	STOP3 command
0x2	STOP2 command
0x1	STOP1 command
0x0	Not terminated

(Default: 0x0)

Resuming execution

After a command execution has stopped, the GE_CMD_ADDR register indicates the next command address that follows the command that has stopped execution except when stopped due to an unexpected operation. Therefore, writing 1 to GE_RUN resumes execution from that command.

However, when an undefined command termination has occurred due to too many or less arguments, the GE_CMD_ADDR register may not point to the following command address. Therefore, be sure of the command start address even if a command execution is stopped due to an undefined command.

Writing 1 to GE_RUN while commands are being executed will be ignored. To resume executing the subsequent commands after the current execution stops, write 1 to GE_RUN after an end-of-execution interrupt has occurred.

27.4.3 Status Check During Execution

The GE provides four status bits, GE_STS, BUS_STS, CALC_STS, and DRAW_STS, in the GE_CTL register. Read these status bits to check the execution status.

GE_STS

GE_STS indicates whether the GE is executing commands or not. It is set to 1 when the GE starts executing commands by writing 1 to GE_RUN and is reset to 0 when the GE has finished/suspended drawing operations. In fact, GE_STS is the result of OR between BUS_STS, CALC_STS, and DRAW_STS.

BUS_STS

BUS_STS indicates the GE bus operation status. It is set to 1 while the GE is fetching a command, reading font or image data, or writing data to the working area, memory, or USIL. It is reset to 0 upon completion of the above bus operation.

Note: BUS_STS is set after lapse of 3 cycles from a trigger by GE_RUN. To avoid reading undefined BUS_STS, insert three or more “nop” instructions after setting GE_RUN to 1.

CALC_STS

CALC_STS indicates the GE calculator status. It is set to 1 when the GE starts initialization or when the GE starts checking command parameters. It is reset to 0 after the GE has finished initialization or the parameter check.

DRAW_STS

DRAW_STS indicates the VRAM writing status. It is set to 1 when the GE starts writing to the work area (VRAM) and is reset to 0 after the writing has been finished.

27.4.4 Updated Range in Work Area

The area that has been rewritten by a drawing command can be checked using the GE_UPDT_ST and GE_UPDT_END registers.

These registers retain the upper left and lower right coordinate values of the rectangular region that contains the objects/images that have been drawn.

X coordinate of upper left corner: UPDT_UPL_X[11:0]/GE_UPDT_ST register

Y coordinate of upper left corner: UPDT_UPL_Y[11:0]/GE_UPDT_ST register

X coordinate of lower right corner: UPDT_LWR_X[11:0]/GE_UPDT_END register

Y coordinate of lower right corner: UPDT_LWR_Y[11:0]/GE_UPDT_END register

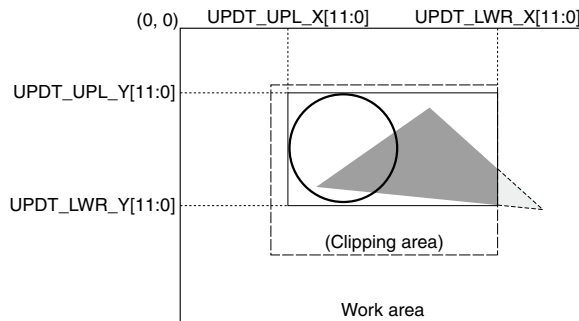


Figure 27.4.4.1 Updated Area

The GE_UPDT_ST and GE_UPDT_END registers will be cleared by writing any value.

- Notes:**
- The coordinates of the updated area can be obtained in the following cases:
 - At least one opaque dot of the specified object is located in the drawing area.
 - As for a transparent object in the drawing area, a copy command or a block transfer command is executed, or the writing effect of rewrite or mesh specifies the transparent color.
 - The coordinates specified for drawing are logical values without consideration of the rotation angle set by VWIN_ROT[1:0]/GE_ROTATE register. The GE converts the logical coordinates into physical values according to the rotation information before it writes to the VRAM. The update area coordinates are physical values after being rotated, therefore the update area coordinates may differ from those specified by the commands.

27.4.5 Errors

Errors that may occur during command execution are divided into calculation errors and drawing errors, and a respective interrupt can be generated when an error has been detected.

Calculation errors

If the GE detects an error while it is checking command arguments, the GE sets the error number to CALC_ERR[3:0]/GE_IF1 register. A calculation error interrupt can be generated at this point. Read CALC_ERR[3:0] in the interrupt handler routine to identify the error that occurred. The CALC_ERR[3:0] bits that have been set are cleared by writing 0x0 or 0xff in byte to address 0x302448.

Table 27.4.5.1 List of Calculation Errors

CALC_ERR[3:0]	Error	
0xf to 0x9	Reserved	–
0x8	Decompress	A Decompress flag (0x80) exists in the image data specified by a decompression command. *3
0x7	Picture header	The header of the image data specified by a decompression command is invalid.
0x6	Font size	The character width or height specified by a character command is 0.
0x5	Radius	The radius specified by a circle/solid filled circle command is invalid (or less than 1).
0x4	Circle location	No arc/circular section to be drawn is specified in a circle/solid filled circle command.
0x3	Concave	An illegal quadrilateral (concave) is specified in a solid filled quadrilateral command. *1 *2
0x2	Coordinates	The area specified as the copy source in a copy command or clipping area in a register configuration command is beyond the work area.
0x1	Trace width	An illegal line width (less than 1 or a value that exceeds the object size) is specified in a object (line) drawing command.
0x0	No error	No error has occurred.

*1 A calculation error and a drawing error (No VRAM write error) have occurred at the same time under the conditions shown below.

- a. When a Concave error (CALC_ERR[3:0] = 0x3) has occurred while calculation error interrupts are enabled (GE_ERR_IE1/GE_IE register = 1)
- b. When a Decompress error (CALC_ERR[3:0] = 0x8) has occurred and the data placed at the front of the invalid decompress flag (0x80) has not been written to the VRAM (drawing on outside the work area/clip area or drawing with the transparent color)

No drawing error will occur when another calculation error has occurred.

*2 Concave errors can be detected only when calculation error interrupts are enabled (GE_ERR_IE1 = 1). Other calculation errors can always be detected regardless of the GE_ERR_IE1 setting.

*3 The image data in which an error occurs will be drawn with the last valid color.

Note: Occurrence of a calculation error does not terminate command execution if calculation error interrupts are disabled (GE_ERR_IE1 = 0). Although the command in which an error has occurred is not executed, the subsequent commands are executed normally. In this case, CALC_ERR[3:0] is not cleared. Be sure to clear CALC_ERR[3:0] before executing the subsequent command list. When calculation error interrupts are enabled (GE_ERR_IE1 = 1), occurrence of an error terminates command execution. In this case, no end-of-execution interrupt will occur.

Drawing errors

If an error is detected while the GE is writing data to the VRAM, the GE sets the corresponding error bit in DRAW_ERR[3:0]/GE_IF2 register. A drawing error interrupt can be generated at this point. Read DRAW_ERR[3:0] in the interrupt handler routine to identify the error that occur. The DRAW_ERR[3:0] bits that have been set are cleared by writing 1.

Table 27.4.5.2 List of Drawing Errors

DRAW_ERR[3:0]	Error	
0b1000	No VRAM write	Data has not been written to the VRAM. The following shows leading causes: 1. All the lines of the specified object are located out of the drawing area (except when a block transfer command or a drawing command with rewrite effect is out of the range). 2. The object/image has only the transparent color (except when a copy command or a block transfer command is executed, or the writing effect of rewrite or mesh specifies the transparent color). 3. A Concave error (CALC_ERR[3:0] = 0x3)* has occurred when calculation error interrupts are enabled (GE_ERR_IE1 = 1).
0b0100	Color depth over	The specified drawing color or a pixel color in the image data is out of the effective range for the bpp mode set in the GE.

DRAW_ERR[3:0]	Error	
0b0010	Clipping area over	The object/image drawn by a command with clipping area enabled is located outside the clipping area. If a part of an object/image is beyond the clipping area, the drawing within the clipping area is performed without an error. Note, however, that error will results if a copy command, a block transfer command, or the writing effect of rewrite follows a drawing operation by another command or a Fill/Mesh/XOR writing effect.
0b0001	Work area over	The object/image drawn by a command with clipping area disabled is located outside the work area. If a part of an object/image is beyond the work area, the drawing within the work area is performed without an error. Note, however, that error will results if a copy command, a block transfer command, or the writing effect of rewrite follows a drawing operation by another command or a Fill/Mesh/XOR writing effect.

* No drawing error occurs when another calculation error has occurred.

The command execution continues even if a drawing error and its interrupt occur regardless of whether drawing error interrupts are enabled or not.

27.4.6 GE Reset

In order to initialize and relinquish the bus used, the GE can be reset via software. The GE supports two reset methods, hot reset and cold reset.

Hot reset

Hot reset initializes the GE with the command and work area address information maintained. The bus is relinquished. Perform hot reset if a command execution is terminated due to an unexpected operation.

To perform hot reset, write 1 to GE_HRST/GE_CTL register. GE_HRST retains 1 while the GE is placed into reset state and it returns to 0 upon completion of the initialization. An execution start request using GE_RUN/GE_CTL register is accepted even if the GE is in reset state and the command execution starts after the initialization has completed.

The following information must be reset after a hot reset.

- Font (font configuration command)
- Clipping area (GE_CLIP_ST and GE_CLIP_END registers)

Cold reset

Cold reset initializes the GE completely.

To perform cold reset, write 1 to GE_CRST/GE_CTL register.

The reset state must be canceled by writing 0 to GE_CRST. If the initialization has not finished yet at this time, DRAW_STS retains 1 and it returns to 0 upon completion of the initialization. An execution start request using GE_RUN/GE_CTL register is accepted even if the GE is in initialization state and the command execution starts after the initialization has completed. After a cold reset, all the information must be reset.

Follow the procedure shown below to perform a cold reset.

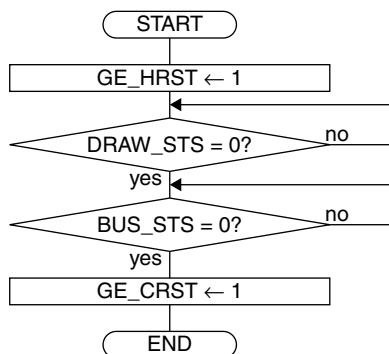


Figure 27.4.6.1 Cold Reset Flow

27.4.7 GE Interrupts

The GE module includes a function for generating the following three different types of interrupts.

- End-of-execution interrupt
- Drawing error interrupt
- Calculation error interrupt

The GE module outputs two interrupt signals to the interrupt controller (ITC). One of them is the end-of-execution interrupt signal and the other is the error interrupt signal shared by the drawing error and calculation error interrupt causes. Inspect the interrupt flags to determine which error interrupt cause occurred.

End-of-execution interrupt

To use this interrupt, set GE_END_IE/GE_IE register to 1. If GE_END_IE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When a command list execution has been terminated, the GE module sets the cause of termination to EXE_END[2:0]/GE_IF3 register (see Table 27.4.2.1). If end-of-execution interrupts are enabled (GE_END_IE = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met.

You can inspect EXE_END[2:0] in the GE end-of-execution interrupt handler routine to determine the cause of termination.

Drawing error interrupt

To use this interrupt, set GE_ERR_IE0/GE_IE register to 1. If GE_ERR_IE0 is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If an error has been detected while the GE is writing data to VRAM by executing a drawing command, the GE module sets the cause of error to DRAW_ERR[3:0]/GE_IF2 register (see Table 27.4.5.2). If drawing error interrupts are enabled (GE_ERR_IE0 = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met.

You can inspect DRAW_ERR[3:0] in the GE error interrupt handler routine to determine which drawing error occurred.

Calculation error interrupt

To use this interrupt, set GE_ERR_IE1/GE_IE register to 1. If GE_ERR_IE1 is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If an error has been detected while the GE is checking arguments by executing a drawing command, the GE module sets the cause of error to CALC_ERR[3:0]/GE_IF1 register (see Table 27.4.5.1). If calculation error interrupts are enabled (GE_ERR_IE1 = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met.

You can inspect CALC_ERR[3:0] in the GE error interrupt handler routine to determine which calculation error occurred.

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

The table below lists the relationship between the interrupt enable bit settings and the interrupts that may be generated.

Table 27.4.7.1 Interrupt Enable Bit Settings and Interrupts

GE_END_IE	GE_ERR_IE1	GE_ERR_IE0	Calculation error	Drawing error	Interrupts that may be generated	Command execution
1	1	1	Occurred*	Occurred*	Calculation error interrupt and drawing error interrupt	Stopped
			Occurred	Not occurred	Calculation error interrupt	Stopped
			Not occurred	Occurred	Drawing error interrupt	Continued
			Not occurred	Not occurred	End-of-execution interrupt (at the end of command list)	Completed
1	1	0	Occurred	Occurred/ Not occurred	Calculation error interrupt	Stopped
			Not occurred	Occurred/ Not occurred	End-of-execution interrupt (at the end of command list)	Completed

GE_END_IE	GE_ERR_IE1	GE_ERR_IE0	Calculation error	Drawing error	Interrupts that may be generated	Command execution
1	0	1	Occurred/	Occurred	Drawing error interrupt	Continued
			Not occurred		End-of-execution interrupt (at the end of command list)	Completed
1	0	0	Occurred/	Not occurred	End-of-execution interrupt (at the end of command list)	Completed
			Not occurred		End-of-execution interrupt (at the end of command list)	Completed
0	1	1	Occurred*	Occurred*	Calculation error interrupt and drawing error interrupt	Stopped
			Occurred		Calculation error interrupt	Stopped
			Not occurred	Occurred	Drawing error interrupt	Continued/ Completed
0	1	0	Not occurred	Not occurred	None	Completed
			Occurred		Occurred/ Not occurred	Calculation error interrupt
0	0	1	Not occurred	Occurred/ Not occurred	None	Completed
			Occurred/ Not occurred		Occurred	Drawing error interrupt
0	0	0	Occurred/ Not occurred	Occurred/ Not occurred	None	Completed
			Occurred/ Not occurred		Occurred/ Not occurred	None

* A calculation error and a drawing error (No VRAM write error) have occurred at the same time under the conditions shown below.

- a. When a Concave error (CALC_ERR[3:0] = 0x3) has occurred while calculation error interrupts are enabled (GE_ERR_IE1/GE_IE register = 1)
- b. When a Decompress error (CALC_ERR[3:0] = 0x8) has occurred and the data placed at the front of the invalid decompress flag (0x80) has not been written to the VRAM (drawing on outside the work area/clip area or drawing with the transparent color)

No drawing error will occur when another calculation error has occurred.

27.5 Command Details

The GE executes the drawing commands programmed in the memory to draw objects/images in the work area. This section explains the all commands provided in the GE individually.

27.5.1 Command Format

Each command consists of a 32-bit command code and maximum five 32-bit arguments. The command code contains a 16-bit start code (0xffff) and a 16-bit command ID (0x1 to 0x2a) that specifies a drawing object/image or control function. Arguments have a different configuration by the command and they follow the command code.

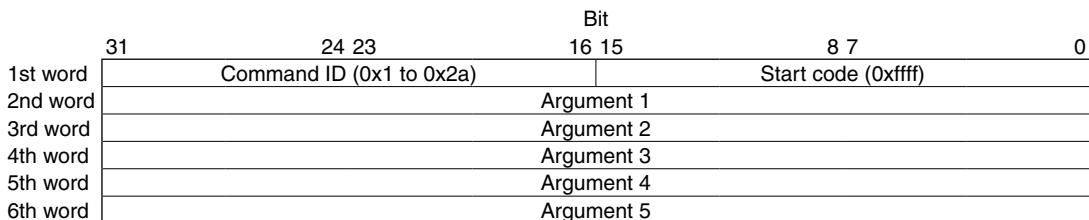


Figure 27.5.1.1 Command Format

- Notes:**
- Do not insert any blank and other characters between commands.
 - Do not omit the arguments.
 - The command must begin from a word (32-bit) boundary address.
 - Command lists must be programmed in IVRAM (Area 3) or an external memory located in Areas 13 to 22. IRAM (Area 0) cannot be used to execute command lists.

Command start address



Figure 27.5.1.2 Command Lists in Memory

27.5.2 List of Commands

Table 27.5.2.1 List of Commands

Command	1st word		2nd word		3rd word		4th word		5th word		6th word	
	H	L	H	L	H	L	H	L	H	L	H	L
	ID	Start	Argument 1		Argument 2		Argument 3		Argument 4		Argument 5	
STOP1 (Stop 1)	0x1	0xffff										
STOP2 (Stop 2)	0x2	0xffff										
STOP3 (Stop 3)	0x3	0xffff										
STOP4 (Stop 4)	0x4	0xffff										
GE_CONFIG (GE configuration)	0x8	0xffff	Address		Data							
DOT (Dot)	0x10	0xffff	Effect Setting	Color	Y	X						
LINE (Straight line)	0x11	0xffff	Effect Setting	Color	reserved	Line Width	Y1	X1	Y2	X2		
RECT (Rectangle)	0x12	0xffff	Effect Setting	Color	reserved	Line Width	Y1	X1	Y2	X2		
TRI_FILL (Solid filled triangle)	0x16	0xffff	Effect Setting	Color	Y1	X1	Y2	X2	Y3	X3		
RECT_FILL (Solid filled rectangle)	0x17	0xffff	Effect Setting	Color	Y1	X1	Y2	X2				
QUAD_FILL (Solid filled quadrilateral)	0x18	0xffff	Effect Setting	Color	Y1	X1	Y2	X2	Y3	X3	Y4	X4
CIRCLE (Circle)	0x1b	0xffff	Effect Setting	Color	Line Width	r	Y	X				
CIRCLE_FILL (Solid filled circle)	0x1c	0xffff	Effect Setting	Color	reserved	r	Y	X				
FONT_CFG (Font configuration)	0x20	0xffff	Index Offset	Font Size	Text Color	BG Color	Font Set Base Address					
CHAR (Character)	0x21	0xffff	Effect Setting	Font Index	Y1	X1	Y2	X2				
DECOMP (Decompression)	0x22	0xffff	Effect Setting	reserved	Start Address		Y1	X1	Y2	X2		
COPY (Copy)	0x29	0xffff	Effect Setting	reserved	SRC Y1	SRC X1	SRC Y2	SRC X2	DST Y	DST X		
BLKCOPY (Block transfer)	0x2a	0xffff	Effect Setting	reserved	Address		Y1	X1	Y2	X2		

27.5.3 STOP1–4 Commands (Stop)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x1 (STOP1) 0x2 (STOP2) 0x3 (STOP3) 0x4 (STOP4)																0xffff															

The STOP (STOP1 to STOP4) command disables the GE to fetch the subsequent commands. Write a STOP command at the end of a command list. The GE stops command execution when it has executed a STOP command. At the same time an end-of-execution interrupt occurs if it has been enabled. After a command execution has stopped, the GE_CMD_ADDR register (command start address) indicates the next command address that follows the command that has stopped execution. Therefore, writing 1 to GE_RUN/GE_CTL register resumes execution from that command.

There is no difference on the stop function of STOP1 to STOP4. However, by reading EXE_END[2:0]/GE_IF3 register after an end-of-execution interrupt has occurred, the STOP (STOP1 to STOP4) command (or the cause of termination), which has stopped command execution, can be identified.

27.5.4 GE_CONFIG Command (GE Configuration)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x8																0xffff															
ARG1	Address																															
ARG2	Data																															

The GE_CONFIG command writes the specified data to the specified address. This enables resetting clipping area and transparent color during command execution.

Argument 1 (Address)

D[31:0] Address[31:0] Bits

Specifies the address of the GE register to be altered.

Argument 2 (Data)

D[31:0] Data[31:0] Bits

Specifies the 32-bit data to be written to the GE register.

Note: The GE_CONFIG command performs a 32-bit writing to the specified address. Therefore, be sure to specify a word boundary address and a 32-bit writing data.

When the clip area specified by the GE_CONFIG command is beyond the work area, a calculation error will occur. Clip area size must be equal to or less than that of the work area.

27.5.5 DOT Command (Dot Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x10																0xffff															
ARG1	Effect Setting																Color															
	reserved										Clip Enable	Transparent Enable	Palette Select	Sync Enable	Write Effect Setting																	
ARG2	Y Coordinate																X Coordinate															

The DOT command fills the pixel at the X and Y coordinates specified with the specified color.

Argument 1 (Drawing effects and color)

D[31:24] Reserved

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

- 1: Clipping enabled
- 0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

- 1: Transparency enabled
- 0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.5.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.5.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Color[15:0] Bits

Specifies the drawing color.

The effective bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

16 bpp mode: Color[15:0] (D[15:0]) bits are effective

8 bpp mode: Color[7:0] (D[7:0]) bits are effective

4 bpp mode: Color[3:0] (D[3:0]) bits are effective

2 bpp mode: Color[1:0] (D[1:0]) bits are effective

1 bpp mode: Color0 (D0) bit is effective

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 2 (X and Y coordinates)**D[31:16] Y Coordinate[15:0] Bits**

Specifies the Y coordinate of the dot to be drawn.

D[15:0] X Coordinate[15:0] Bits

Specifies the X coordinate of the dot to be drawn.

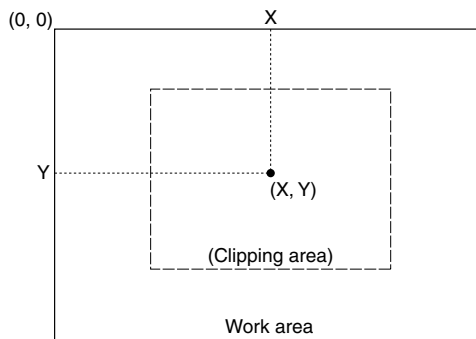


Figure 27.5.5.1 Dot Drawing

27.5.6 LINE Command (Straight Line Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x11																0xffff															
ARG1	Effect Setting																Color															
	reserved																															
ARG2	reserved																Line Width															
ARG3	Y1 Coordinate																X1 Coordinate															
ARG4	Y2 Coordinate																X2 Coordinate															

The LINE command draws a line between the two points specified in the X and Y coordinates with the specified color. The line width can also be specified in number of pixels. Not only horizontal and vertical lines but also oblique lines can be drawn according to the specified coordinates.

Argument 1 (Drawing effects and color)

D[31:24] Reserved

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.6.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.6.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7-0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Color[15:0] Bits

Specifies the drawing color.

The effective bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

16 bpp mode: Color[15:0] (D[15:0]) bits are effective

8 bpp mode: Color[7:0] (D[7:0]) bits are effective

4 bpp mode: Color[3:0] (D[3:0]) bits are effective

2 bpp mode: Color[1:0] (D[1:0]) bits are effective

1 bpp mode: Color0 (D0) bit is effective

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 2 (Line width)

D[31:12] Reserved

D[11:0] Line Width[11:0] Bits

Specifies the line width in number of pixels. (See Section 27.3.2.)

Argument 3 (Coordinates of starting point)

D[31:16] Y1 Coordinate[15:0] Bits

Specifies the Y coordinate of the starting point.

D[15:0] X1 Coordinate[15:0] Bits

Specifies the X coordinate of the starting point.

Argument 4 (Coordinates of endpoint)

D[31:16] Y2 Coordinate[15:0] Bits

Specifies the Y coordinate of the endpoint.

D[15:0] X2 Coordinate[15:0] Bits

Specifies the X coordinate of the endpoint.

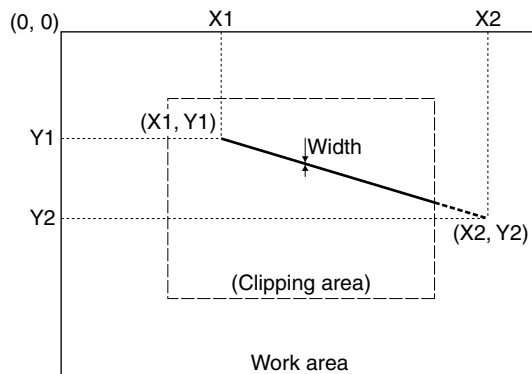


Figure 27.5.6.1 Straight Line Drawing

27.5.7 RECT Command (Rectangle Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
CMD	0x12																0xffff																					
ARG1	reserved																Effect Setting						Color															
																	Clip Enable	Transparent Enable	Palette Select	Sync Enable	Write Effect Setting																	
ARG2	reserved																Line Width																					
ARG3	Y1 Coordinate																X1 Coordinate																					
ARG4	Y2 Coordinate																X2 Coordinate																					

27 GRAPHICS ENGINE (GE)

The RECT command draws a rectangle with the colored line specified. Specify the X and Y coordinates of the upper left and lower right corners. The line width can also be specified in number of pixels.

Argument 1 (Drawing effects and color)

D[31:24] Reserved

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.7.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.7.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Color[15:0] Bits

Specifies the drawing color.

The effective bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

16 bpp mode: Color[15:0] (D[15:0]) bits are effective

8 bpp mode: Color[7:0] (D[7:0]) bits are effective

4 bpp mode: Color[3:0] (D[3:0]) bits are effective

2 bpp mode: Color[1:0] (D[1:0]) bits are effective

1 bpp mode: Color0 (D0) bit is effective

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 2 (Line width)

D[31:12] Reserved

D[11:0] Line Width[11:0] Bits

Specifies the line width in number of pixels. (See Section 27.3.2.)

Argument 3 (Coordinates of upper left corner)**D[31:16] Y1 Coordinate[15:0] Bits**

Specifies the Y coordinate of the upper left corner.

D[15:0] X1 Coordinate[15:0] Bits

Specifies the X coordinate of the upper left corner.

Argument 4 (Coordinates of lower right corner)**D[31:16] Y2 Coordinate[15:0] Bits**

Specifies the Y coordinate of the lower right corner.

D[15:0] X2 Coordinate[15:0] Bits

Specifies the X coordinate of the lower right corner.

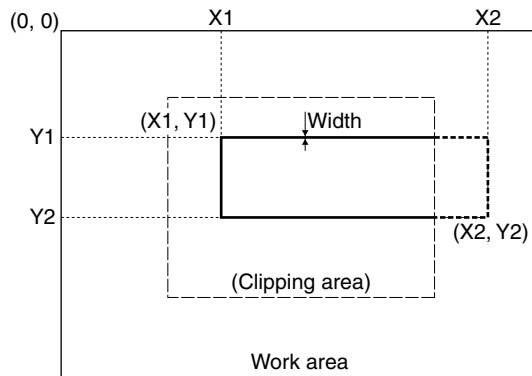


Figure 27.5.7.1 Rectangle Drawing

27.5.8 TRI_FILL Command (Solid Filled Triangle Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
CMD	0x16																0xffff																				
ARG1	Effect Setting																Color																				
	reserved																																				
																	Clip Enable	Transparent Enable	Palette Select	Sync Enable	Write Effect Setting																
ARG2	Y1 Coordinate																X1 Coordinate																				
ARG3	Y2 Coordinate																X2 Coordinate																				
ARG4	Y3 Coordinate																X3 Coordinate																				

The TRI_FILL command draws a solid filled triangle specified with the X and Y coordinates of the three vertices, and a color.

Argument 1 (Drawing effects and color)**D[31:24] Reserved****D23 Clip Enable Bit**

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.8.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.8.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Color[15:0] Bits

Specifies the drawing color.

The effective bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

16 bpp mode: Color[15:0] (D[15:0]) bits are effective

8 bpp mode: Color[7:0] (D[7:0]) bits are effective

4 bpp mode: Color[3:0] (D[3:0]) bits are effective

2 bpp mode: Color[1:0] (D[1:0]) bits are effective

1 bpp mode: Color0 (D0) bit is effective

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 2 (Coordinates of 1st vertex)**D[31:16] Y1 Coordinate[15:0] Bits**

Specifies the Y coordinate of the 1st vertex.

D[15:0] X1 Coordinate[15:0] Bits

Specifies the X coordinate of the 1st vertex.

Argument 3 (Coordinates of 2nd vertex)**D[31:16] Y2 Coordinate[15:0] Bits**

Specifies the Y coordinate of the 2nd vertex.

D[15:0] X2 Coordinate[15:0] Bits

Specifies the X coordinate of the 2nd vertex.

Argument 4 (Coordinates of 3rd vertex)**D[31:16] Y3 Coordinate[15:0] Bits**

Specifies the Y coordinate of the 3rd vertex.

D[15:0] X3 Coordinate[15:0] Bits

Specifies the X coordinate of the 3rd vertex.

- Notes:**
- The coordinates of three vertices must be specified in a clockwise direction. No error occurs when specified in a counterclockwise direction, note, however, that an unexpected drawing will result.
 - Be sure to avoid specifications in which three vertices are aligned in a line. No error occurs in this case, note, however, that an unexpected drawing will result. If the same coordinates are specified for two or three vertices, a line or dot will be drawn.

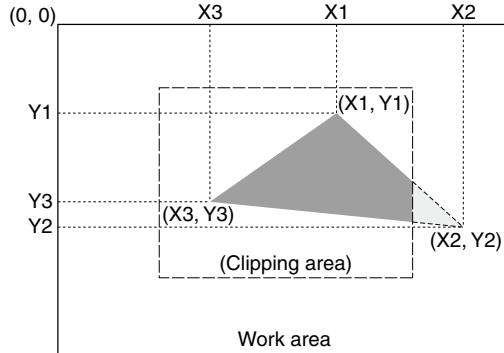


Figure 27.5.8.1 Drawing Solid Filled Triangle

27.5.9 RECT_FILL Command (Solid Filled Rectangle Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x17																0xffff															
ARG1	Effect Setting																Color															
	reserved																															
	Clip Enable																Transparent Enable															
	Palette Select																Sync Enable															
	Write Effect Setting																															
ARG2	Y1 Coordinate																X1 Coordinate															
ARG3	Y2 Coordinate																X2 Coordinate															

The RECT_FILL command draws a solid filled rectangle specified with the X and Y coordinates of the upper left and lower right corners, and a color.

Argument 1 (Drawing effects and color)

D[31:24] Reserved

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.9.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.9.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Color[15:0] Bits

Specifies the drawing color.

The effective bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

16 bpp mode: Color[15:0] (D[15:0]) bits are effective

8 bpp mode: Color[7:0] (D[7:0]) bits are effective

4 bpp mode: Color[3:0] (D[3:0]) bits are effective

2 bpp mode: Color[1:0] (D[1:0]) bits are effective

1 bpp mode: Color0 (D0) bit is effective

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 2 (Coordinates of upper left corner)

D[31:16] Y1 Coordinate[15:0] Bits

Specifies the Y coordinate of the upper left corner.

D[15:0] X1 Coordinate[15:0] Bits

Specifies the X coordinate of the upper left corner.

Argument 3 (Coordinates of lower right corner)

D[31:16] Y2 Coordinate[15:0] Bits

Specifies the Y coordinate of the lower right corner.

D[15:0] X2 Coordinate[15:0] Bits

Specifies the X coordinate of the lower right corner.

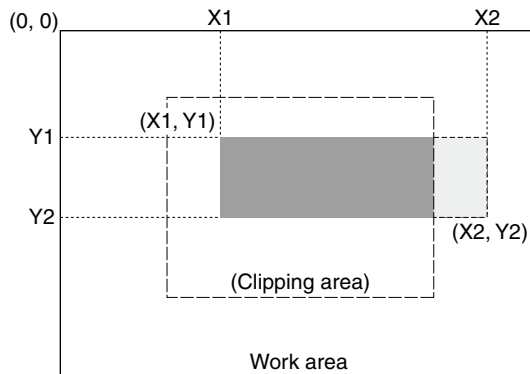


Figure 27.5.9.1 Drawing Solid Filled Rectangle

27.5.10 QUAD_FILL Command (Solid Filled Quadrilateral Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x18																0xffff															
ARG1	Effect Setting																Color															
	reserved																															
ARG2	Y1 Coordinate																X1 Coordinate															
ARG3	Y2 Coordinate																X2 Coordinate															
ARG4	Y3 Coordinate																X3 Coordinate															
ARG5	Y4 Coordinate																X4 Coordinate															

The QUAD_FILL command draws a solid filled quadrilateral specified with the X and Y coordinates of the four vertices, and a color.

Argument 1 (Drawing effects and color)

D[31:24] Reserved

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.10.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.10.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7-0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Color[15:0] Bits

Specifies the drawing color.

The effective bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

- 16 bpp mode: Color[15:0] (D[15:0]) bits are effective
- 8 bpp mode: Color[7:0] (D[7:0]) bits are effective
- 4 bpp mode: Color[3:0] (D[3:0]) bits are effective
- 2 bpp mode: Color[1:0] (D[1:0]) bits are effective
- 1 bpp mode: Color0 (D0) bit is effective

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 2 (Coordinates of 1st vertex)

D[31:16] Y1 Coordinate[15:0] Bits

Specifies the Y coordinate of the 1st vertex.

D[15:0] X1 Coordinate[15:0] Bits

Specifies the X coordinate of the 1st vertex.

Argument 3 (Coordinates of 2nd vertex)

D[31:16] Y2 Coordinate[15:0] Bits

Specifies the Y coordinate of the 2nd vertex.

D[15:0] X2 Coordinate[15:0] Bits

Specifies the X coordinate of the 2nd vertex.

Argument 4 (Coordinates of 3rd vertex)

D[31:16] Y3 Coordinate[15:0] Bits

Specifies the Y coordinate of the 3rd vertex.

D[15:0] X3 Coordinate[15:0] Bits

Specifies the X coordinate of the 3rd vertex.

Argument 5 (Coordinates of 4th vertex)

D[31:16] Y4 Coordinate[15:0] Bits

Specifies the Y coordinate of the 4th vertex.

D[15:0] X4 Coordinate[15:0] Bits

Specifies the X coordinate of the 4th vertex.

- Notes:**
- This command supports convex quadrilaterals only and does not draw concave quadrilaterals. An error results and nothing will be drawn if a concave quadrilateral is specified when calculation error interrupts are enabled (GE_ERR_IE1 = 1). When GE_ERR_IE1 = 0 (interrupt disabled), no error occurs but an unexpected drawing will result.
 - The coordinates of four vertices must be specified in a clockwise direction. No error occurs when specified in a counterclockwise direction, note, however, that an unexpected drawing will result.
 - Be sure to avoid specifications in which four vertices are aligned in a line. No error occurs in this case, note, however, that an unexpected drawing will result. If the same coordinates are specified for two, three, or four vertices, a triangle, line, or dot will be drawn.

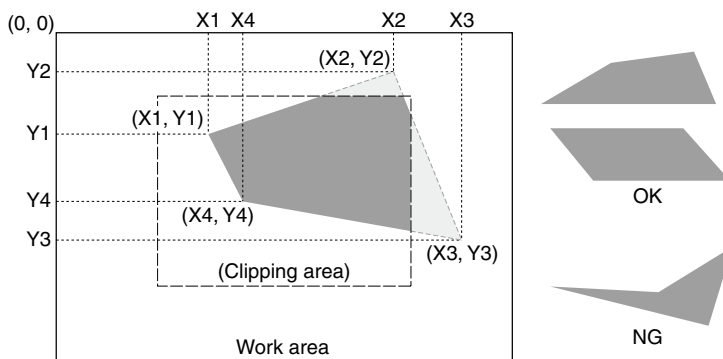


Figure 27.5.10.1 Drawing Solid Filled Quadrilateral

27.5.11 CIRCLE Command (Circle Drawing)

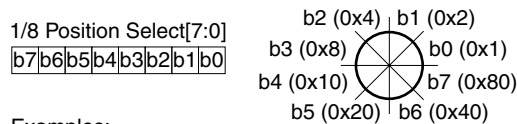
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x1b																0xffff															
ARG1	Effect Setting																Color															
	1/8 Position Select								Clip Enable	Transparent Enable	Palette Select		Sync Enable	Write Effect Setting																		
ARG2	reserved				Line Width								reserved				Radius															
ARG3	Y Coordinate																X Coordinate															

The CIRCLE command draws the circumference of a circle specified with the center coordinates, radius, and a line color. The line width can be specified in number of pixels. Furthermore, this command allows selection of the arcs to be drawn from the eight segments split by 45°.

Argument 1 (Drawing effects and color)

D[31:24] 1/8 Position Select[7:0] Bits

Selects one or more arcs to be drawn from eight locations split by 45°. Set the 1/8 Position Select[7:0] bits corresponding to the arcs to be drawn to 1 and execute the command. To draw a complete circle, set all the 1/8 Position Select[7:0] bits to 1.



Examples:

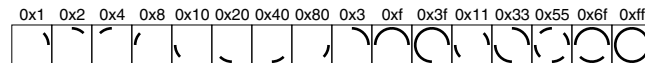


Figure 27.5.11.1 Arc Selections

Note: When two or more arcs are drawn with separate commands, the pixels at the boundary are overwritten. This must be taken into consideration when using a writing effect, XOR in particular. If arcs are drawn in the same color, the XOR writing effect clears the overlaid boundary pixels.

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.11.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.11.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Color[15:0] Bits

Specifies the drawing color.

The effective bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

16 bpp mode: Color[15:0] (D[15:0]) bits are effective

8 bpp mode: Color[7:0] (D[7:0]) bits are effective

4 bpp mode: Color[3:0] (D[3:0]) bits are effective

2 bpp mode: Color[1:0] (D[1:0]) bits are effective

1 bpp mode: Color0 (D0) bit is effective

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 2 (Line width and radius)

D[31:28] Reserved

D[27:16] Line Width[11:0] Bits

Specifies the line width in number of pixels.

D[15:12] Reserved

D[11:0] Radius[11:0] Bits

Specifies the radius in number of pixels.

Argument 3 (Coordinates of center)

D[31:16] Y Coordinate[15:0] Bits

Specifies the Y coordinate of the center.

D[15:0] X Coordinate[15:0] Bits

Specifies the X coordinate of the center.

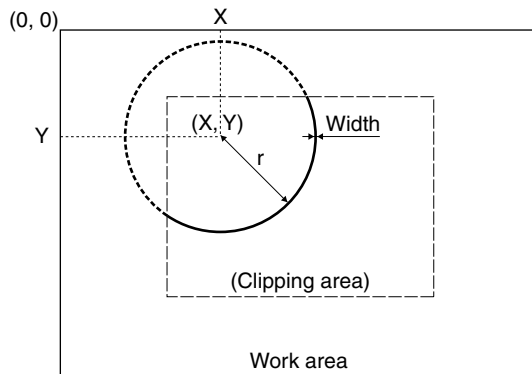


Figure 27.5.11.2 Circle Drawing

27.5.12 CIRCLE_FILL Command (Solid Filled Circle Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x1c																0xffff															
ARG1	Effect Setting																Color															
	1/8 Position Select							Clip Enable	Transparent Enable	Palette Select		Sync Enable	Write Effect Setting																			
ARG2	reserved																Radius															
ARG3	Y Coordinate																X Coordinate															

The CIRCLE_FILL command draws a solid filled circle specified with the center coordinates, radius, and a color. Furthermore, this command allows selection of the circular sections to be drawn from the eight portions split by 45°.

Argument 1 (Drawing effects and color)

D[31:24] 1/8 Position Select[7:0] Bits

Selects one or more circular sections to be drawn from eight locations split by 45°. Set the 1/8 Position Select[7:0] bits corresponding to the circular sections to be drawn to 1 and execute the command. To draw a complete circle, set all the 1/8 Position Select[7:0] bits to 1.

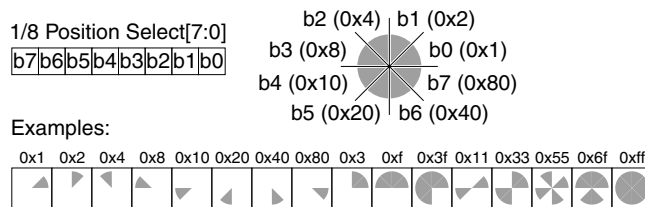


Figure 27.5.12.1 Circular Section Selections

Note: When two or more circular sections are drawn with separate commands, the pixels at the boundary are overwritten. This must be taken into consideration when using a writing effect, XOR in particular. If circular sections are drawn in the same color, the XOR writing effect clears the overlaid boundary pixels.

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

- 1: Clipping enabled
- 0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

- 1: Transparency enabled
- 0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.12.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

- 1: Enabled (Synchronized)
- 0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.12.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x4	Reserved
0x3	Rewrite
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Color[15:0] Bits

Specifies the drawing color.

The effective bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

16 bpp mode: Color[15:0] (D[15:0]) bits are effective

8 bpp mode: Color[7:0] (D[7:0]) bits are effective

4 bpp mode: Color[3:0] (D[3:0]) bits are effective

2 bpp mode: Color[1:0] (D[1:0]) bits are effective

1 bpp mode: Color0 (D0) bit is effective

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 2 (Radius)

D[31:12] Reserved

D[11:0] Radius[11:0] Bits

Specifies the radius in number of pixels.

Argument 3 (Coordinates of center)

D[31:16] Y Coordinate[15:0] Bits

Specifies the Y coordinate of the center.

D[15:0] X Coordinate[15:0] Bits

Specifies the X coordinate of the center.

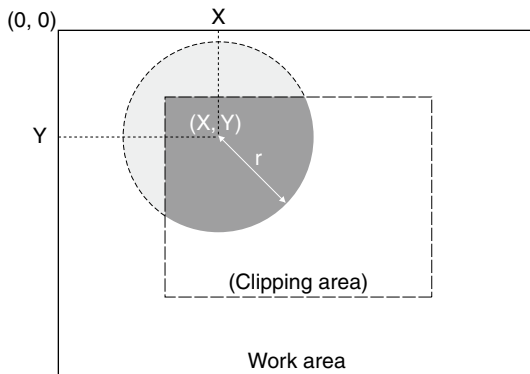


Figure 27.5.12.2 Drawing Solid Filled Circle

27.5.13 FONT_CFG Command (Font Configuration)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x20																0xffff															
ARG1	reserved								Font Index Offset								reserved	Height				reserved	Width									
ARG2	Character Color																Background color															
ARG3	Font Set Base Address																															

The FONT_CFG command sets the font information, such as the memory location where the font data (character set) is stored, character and background colors, and font bitmap size, to the GE module. This setting must be performed before starting character drawing. For information on the font data format, see Section 27.3.3.

Argument 1 (Font configuration)

D[31:24] Reserved

D[23:16] Font Index Offset[7:0] Bits

Specifies the index code size in bytes for each character data in the font set to be used. If the character data consists of bitmap data only without index code, specify as 0x0.

D15 Reserved

D[14:8] Height[6:0] Bits

Specifies the height of the font in number of pixels.

D7 Reserved

D[6:0] Width[6:0] Bits

Specifies the width of the font in number of pixels.

Argument 2 (Color settings)

D[31:16] Character Color[15:0] Bits

Specifies the drawing color for characters.

D[15:0] Background Color[15:0] Bits

Specifies the background color for characters.

The effective Character/Background Color bits depend on the GE bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register).

16 bpp mode: Character/Background Color[15:0] (D[31:16]/D[15:0]) bits are effective.

8 bpp mode: Character/Background Color[7:0] (D[23:16]/D[7:0]) bits are effective.

4 bpp mode: Character/Background Color[3:0] (D[19:16]/D[3:0]) bits are effective.

2 bpp mode: Character/Background Color[1:0] (D[17:16]/D[1:0]) bits are effective.

1 bpp mode: Character/Background Color0 (D16/D0) bit is effective.

The color data may be modified before being written to the work area according to the palette and/or writing effect settings. (See Section 27.3.5.)

Argument 3 (Font set base address)

D[31:0] Font Set Base Address[31:0] Bits

Specifies the memory address from which the font set data is stored.

27.5.14 CHAR Command (Character Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x21																0xffff															
ARG1	Effect Setting																Font Index															
	reserved	Resize/Tile Select	reserved	Rotation Select	Clip Enable	Transparent Enable	Palette Select	Sync Enable	Write Effect Setting																							
ARG2	Y1 Coordinate																X1 Coordinate															
ARG3	Y2 Coordinate																X2 Coordinate															

The CHAR command draws a character or symbol specified by an index in the character set at the specified location. The command also allows specification of drawing effects such as resizing, tiling, and rotation as well as general drawing effects.

Argument 1 (Drawing effects and character)**D[31:30] Reserved****D[29:28] Resize/Tile Select[1:0] Bits**

Specifies resizing or tiling effect for drawing the character. (See Section 27.3.5.)

Table 27.5.14.1 Resizing/Tiling Selections

Resize/Tile Select[1:0] bits	Drawing effect
0x3	Reserved
0x2	Tiling
0x1	Resizing
0x0	Normal

D[27:26] Reserved**D[25:24] Rotation Select[1:0] Bits**

Selects a character rotation angle. (See Section 27.3.5.)

Table 27.5.14.2 Angle of Character Rotation

Rotation Select[1:0] bits	Rotation angle
0x3	270°
0x2	180°
0x1	90°
0x0	0°

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.14.3 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCD synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.14.4 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x3	Reserved
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Font Index[15:0] Bits

Specifies the character to be drawn with the index in the font set.

The first character is specified as index 0 and the subsequent character indexes are incremented by character.

Argument 2 (Coordinates of upper left corner)

D[31:16] Y1 Coordinate[15:0] Bits

Specifies the Y coordinate of the character drawing area upper left corner.

D[15:0] X1 Coordinate[15:0] Bits

Specifies the X coordinate of the character drawing area upper left corner.

Argument 3 (Coordinates of lower right corner)

D[31:16] Y2 Coordinate[15:0] Bits

Specifies the Y coordinate of the character drawing area lower right corner.

D[15:0] X2 Coordinate[15:0] Bits

Specifies the X coordinate of the character drawing area lower right corner.

When the Resize/Tile Select[1:0] bits are set to 0x0 (normal drawing), the coordinate setting values of the lower right corner are ignored.

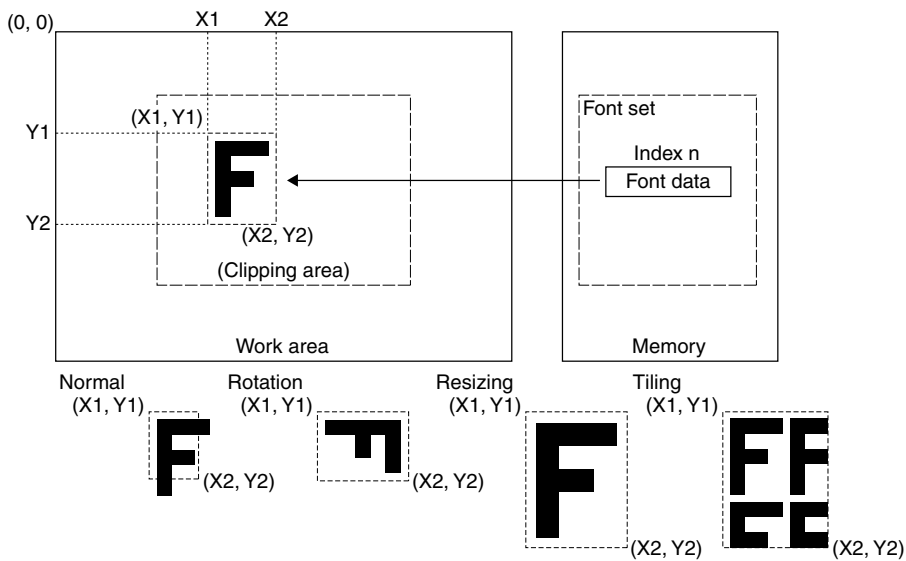


Figure 27.5.14.1 Character Drawing

27.5.15 DECOMP Command (Decompression and Drawing)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x22																0xffff															
ARG1	Effect Setting																reserved															
	reserved	Resize/Tile Select	reserved	Rotation Select	Clip Enable	Transparent Enable	Palette Select	Sync Enable	Write Effect Setting																							
ARG2	Start Address																															
ARG3	Y1 Coordinate																X1 Coordinate															
ARG4	Y2 Coordinate																X2 Coordinate															

The DECOMP command decompresses the compressed image data (Run length encoded data) located at the specified memory address and draws the image within the area specified with the X and Y coordinates of the upper left and lower right corners in the work area. The GE also supports uncompressed data. The command allows specification of drawing effects such as resizing, tiling, and rotation as well as general drawing effects.

If the image data has a color depth (1/2/4 bpp) different from the bpp mode set in the GE module, the color data is automatically converted according to the GE bpp mode before the image is written to the work area (color reduction is not supported).

For more information on the compressed data format and the automatic color depth conversion function, see Section 27.3.4.

Argument 1 (Drawing effects)

D[31:30] Reserved

D[29:28] Resize/Tile Select[1:0] Bits

Selects resizing or tiling effect for drawing the image. (See Section 27.3.5.)

Table 27.5.15.1 Resizing/Tiling Selections

Resize/Tile Select[1:0] bits	Drawing effect
0x3	Reserved
0x2	Tiling
0x1	Resizing
0x0	Normal

D[27:26] Reserved

D[25:24] Rotation Select[1:0] Bits

Selects an image rotation angle. (See Section 27.3.5.)

Table 27.5.15.2 Angle of Image Rotation

Rotation Select[1:0] bits	Rotation angle
0x3	270°
0x2	180°
0x1	90°
0x0	0°

D23 Clip Enable Bit

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.15.3 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.15.4 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x3	Reserved
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Reserved

Argument 2 (Image data start address)

D[31:0] Start Address[31:0] Bits

Specifies the memory address from which the image data is stored.

Argument 3 (Coordinates of upper left corner)

D[31:16] Y1 Coordinate[15:0] Bits

Specifies the Y coordinate of the image drawing area upper left corner.

D[15:0] X1 Coordinate[15:0] Bits

Specifies the X coordinate of the image drawing area upper left corner.

Argument 4 (Coordinates of lower right corner)

D[31:16] Y2 Coordinate[15:0] Bits

Specifies the Y coordinate of the image drawing area lower right corner.

D[15:0] X2 Coordinate[15:0] Bits

Specifies the X coordinate of the image drawing area lower right corner.

When the Resize/Tile Select[1:0] bits are set to 0x0 (normal drawing), the coordinate setting values of the lower right corner are ignored.

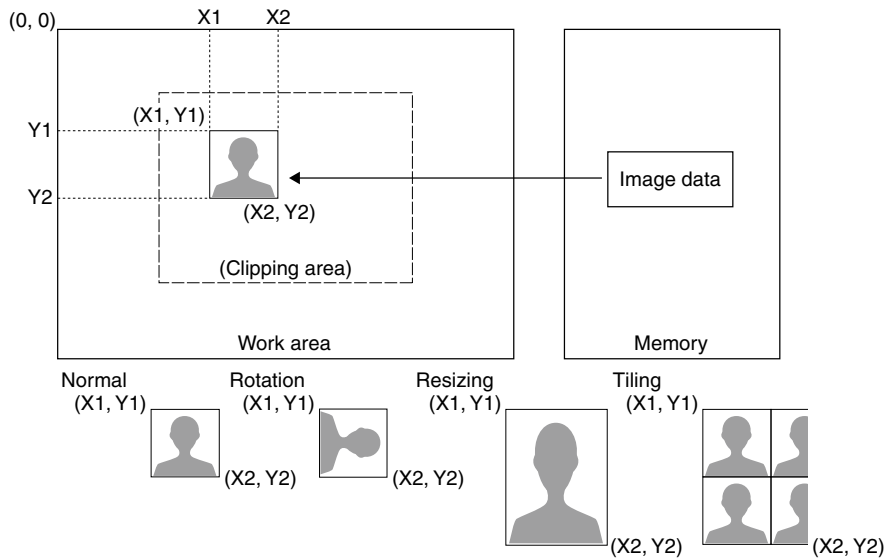


Figure 27.5.15.1 Decompression and Drawing

27.5.16 COPY Command (Copy)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x29																0xffff															
ARG1	Effect Setting																reserved															
	reserved																															
ARG2	Source Y1 Coordinate																Source X1 Coordinate															
ARG3	Source Y2 Coordinate																Source X2 Coordinate															
ARG4	Destination Y Coordinate																Destination X Coordinate															

The COPY command copies the rectangular area specified with the X and Y coordinates of the upper left and lower right corners to the destination area specified within the work area. This copy function can guarantee that the source image will be copied to the destination properly even if the source and destination areas overlap one another. Drawing effects such as clipping and transparency that apply to the destination can also be specified.

Argument 1 (Drawing effects)**D[31:24] Reserved****D23 Clip Enable Bit**

Enables or disables the clipping function. (See Section 27.3.5.)

1: Clipping enabled

0: Clipping disabled

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

1: Transparency enabled

0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette. (See Section 27.3.5.)

Table 27.5.16.1 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

1: Enabled (Synchronized)

0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect. (See Section 27.3.5.)

Table 27.5.16.2 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x3	Reserved
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

D[15:0] Reserved**Argument 2 (Coordinates of transfer source area upper left corner)****D[31:16] Source Y1 Coordinate[15:0] Bits**

Specifies the Y coordinate of the transfer source area upper left corner.

D[15:0] Source X1 Coordinate[15:0] Bits

Specifies the X coordinate of the transfer source area upper left corner.

Argument 3 (Coordinates of transfer source area lower right corner)**D[31:16] Source Y2 Coordinate[15:0] Bits**

Specifies the Y coordinate of the transfer source area lower right corner.

D[15:0] Source X2 Coordinate[15:0] Bits

Specifies the X coordinate of the transfer source area lower right corner.

Argument 4 (Coordinates of transfer destination area upper left corner)**D[31:16] Destination Y1 Coordinate[15:0] Bits**

Specifies the Y coordinate of the transfer destination area upper left corner.

D[15:0] Destination X1 Coordinate[15:0] Bits

Specifies the X coordinate of the transfer destination area upper left corner.

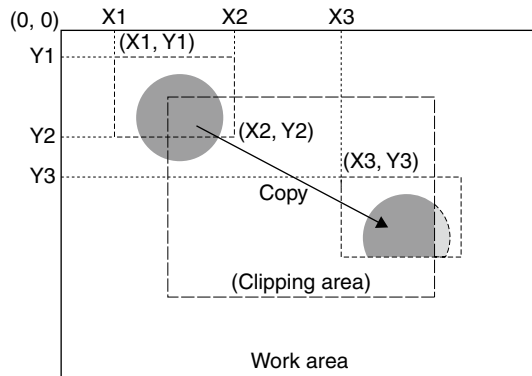


Figure 27.5.16.1 Image Copy

27.5.17 BLKCOPY Command (Block Transfer)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD	0x2a																0xffff															
ARG1	Effect Setting																reserved															
	reserved	Memory Select	reserved	Transparent Enable	Palette Select	Sync Enable	Write Effect Setting																									
ARG2	Address																															
ARG3	Y1 Coordinate																X1 Coordinate															
ARG4	Y2 Coordinate																X2 Coordinate															

The BLKCOPY command transfers image data block between a specified area within the work area and memory or a built-in RAM LCD driver (via USIL). The source and destination can be selected from four combinations in a command argument. Drawing effects such as palette conversion can be specified when sending data to the work area.

Argument 1 (Drawing effects)

D[31:28] Reserved

D[27:26] Memory Select[1:0] Bits

Selects the transfer source and destination.

Table 27.5.17.1 Block Transfer Specification

Memory Select[1:0] bits	Source	Destination
0x3	Work area	USIL (big endian) *1
0x2		USIL (little endian) *1
0x1		Memory *2
0x0	Memory *2	Work area

*1: Data is transferred to the built-in RAM LCD driver/panel via USIL set to LCD SPI mode or LCD parallel I/F mode.

Specify the USIL transmit data buffer address as the destination.

*2: IVRAM (Area 3) or an external memory located in Areas 13 to 22

D[25:23] Reserved

D22 Transparent Enable Bit

Enables or disables transparency. (See Section 27.3.5.)

- 1: Transparency enabled
- 0: Transparency disabled

D[21:20] Palette Select[1:0] Bits

Selects the palette to be used when converting colors with a palette during data transfer to the work area. (See Section 27.3.5.)

Table 27.5.17.2 Palette Selection

Palette Select[1:0] bits	Palette
0x3	Palette 3
0x2	Palette 2
0x1	Palette 1
0x0	Not used

Note: The palette conversions are effective only when data is transferred to the work area.

D19 Sync Enable Bit

Enables or disables the LCDC synchronization function. (See Section 27.3.5.)

- 1: Enabled (Synchronized)
- 0: Disabled (Not synchronized)

D[18:16] Write Effect Setting[2:0] Bits

Selects a writing effect to be applied when data is transferred to the work area. (See Section 27.3.5.)

Table 27.5.17.3 Writing Effect Selections

Write Effect Setting[2:0] bits	Writing effect
0x7–0x3	Reserved
0x2	Mesh
0x1	XOR
0x0	Normal (Fill)

Note: The Writing effects are effective only when data is transferred to the work area.

D[15:0] Reserved**Argument 2 (Memory address)****D[31:0] Address[31:0] Bits**

Specifies the transfer source/destination memory address or the USIL transmit data buffer address used as the transfer destination.

Argument 3 (Coordinates of transfer area upper left corner)**D[31:16] Y1 Coordinate[15:0] Bits**

Specifies the Y coordinate of the transfer area upper left corner (within work area).

D[15:0] X1 Coordinate[15:0] Bits

Specifies the X coordinate of the transfer area upper left corner (within work area).

Argument 4 (Coordinates of transfer area lower right corner)**D[31:16] Y2 Coordinate[15:0] Bits**

Specifies the Y coordinate of the transfer area lower right corner (within work area).

D[15:0] X2 Coordinate[15:0] Bits

Specifies the X coordinate of the transfer area lower right corner (within work area).

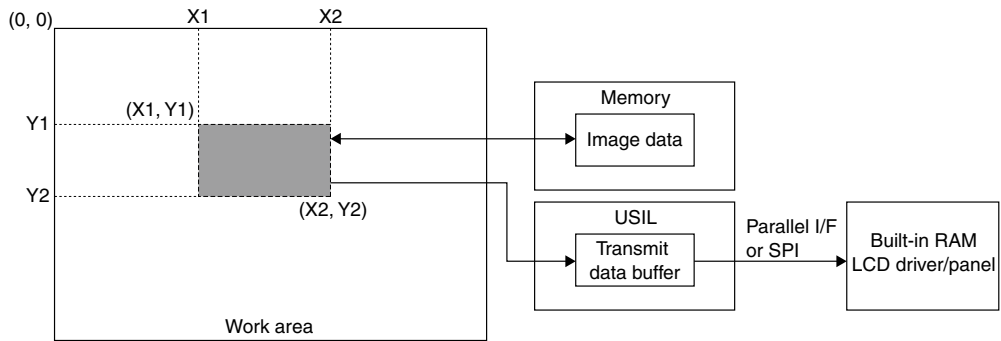


Figure 27.5.17.1 Image Data Block Transfer

27.6 Control Register Details

Table 27.6.1 List of GE Registers

Address	Register name		Function
0x30240c	GE_CMD_ADDR	GE Command Address Register	Set command list start address
0x302440	GE_CTL	GE Control Register	Control/indicate GE operation/status
0x302444	GE_IE	GE Interrupt Enable Register	Enable/disable GE interrupts
0x302448	GE_IF1	GE Interrupt Flag Register 1	Indicate calculation error status
0x302449	GE_IF2	GE Interrupt Flag Register 2	Indicate drawing error status
0x30244a	GE_IF3	GE Interrupt Flag Register 3	Indicate cause of termination
0x30244c	GE_REAL_W	VRAM Work Area Width Register	Indicate work area width after VRAM rotation
0x302450	GE_WK_ADDR	VRAM Work Area Start Address Register	Set work (drawing) area start address
0x302454	GE_WK_SIZE	VRAM Work Area Size Register	Set work area size
0x302458	GE_DISP_CFG	Display Configuration Register	Set color depth and LCDC sync type
0x30245c	GE_ROTATE	VRAM Rotation Control Register	Specify display rotation angle
0x302460	GE_CLIP_ST	Clipping Area Start Position Register	Specify upper left corner of clipping area
0x302464	GE_CLIP_END	Clipping Area End Position Register	Specify lower right corner of clipping area
0x302468	GE_MESH	Mesh Configuration Register	Configure mesh size and color
0x30246c	GE_MAGIC	Transparent Color Register	Specify transparent color
0x302470	GE_UPDT_ST	Updated Area Start Position Register	Indicate upper left corner of updated area
0x302474	GE_UPDT_END	Updated Area End Position Register	Indicate lower right corner of updated area
0x302800 –0x3028ff	GE_PALETTE1	Palette 1	Palette 1
0x302910 –0x30291f	GE_CCT1_4BIT	CCT1 4-bit Entries	CCT1 4 to 8 bpp conversion entries (user programmable)
0x302920 –0x302923	GE_CCT1_2BIT	CCT1 2-bit Entries	CCT1 2 to 4/8 bpp conversion entries (user programmable)
0x302924 –0x302925	GE_CCT1_1BIT	CCT1 1-bit Entries	CCT1 1 to 2/4/8 bpp conversion entries (user programmable)

The GE registers are described in detail below.

- Notes:**
- When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1.
 - Do not write any data to reserved words other than those listed in the table above.

GE Command Address Register (GE_CMD_ADDR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
GE Command Address Register (GE_CMD_ADDR)	0x30240c (32 bits)	D31–2	CMD_ADDR[31:2]	GE command address (Word boundary address)	0x0 to 0xfffffc	0x0	R/W	
		D1–0	CMD_ADDR[1:0]	Fixed at 0x0 (Cannot be altered.)			R	

D[31:0] CMD_ADDR[31:0]: GE Command Address Bits

Specifies the memory address from which the command list to be executed begins. (Default: 0x0)

The command start address must be a word boundary address in Areas 3–5, 7–10, 13–16, and 19–22.

The low-order 2 bits are always fixed at 0. The address in this register is incremented by the size of the command executed indicating the command address to be subsequently executed.

GE Control Register (GE_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
GE Control Register (GE_CTL)	0x302440 (32 bits)	D31-17	-	reserved	-	-	-	0 when being read.
		D16	GE_STS	GE operation status	1 Busy 0 Idle	0	R	
		D15-11	-	reserved	-	-	-	0 when being read.
		D10	BUS_STS	Bus operation status	1 Running 0 Stop	0	R	
		D9	CALC_STS	Calculator operation status	1 Running 0 Stop	0	R	
		D8	DRAW_STS	Pixel drawing status	1 Running 0 Stop	0	R	
		D7-4	-	reserved	-	-	-	0 when being read.
		D3	GE_STOP	GE stop control	1 Stop 0 -	0	R/W	
		D2	GE_RUN	GE run control	1 Run trigger 0 -	0	R/W	
		D1	GE_HRST	GE hot reset control	1 Reset 0 Normal mode	0	R/W	
		D0	GE_CRST	GE cold reset control	1 Cold reset 0 Normal mode	1	R/W	

D[31:17] Reserved

D16 GE_STS: GE Operation Status Bit

Indicates whether the GE is executing commands or not.

1 (R): Busy

0 (R): Idle (default)

GE_STS is set to 1 when the GE starts executing commands by writing 1 to GE_RUN and is reset to 0 when the GE has finished/suspended drawing operations. In fact, GE_STS is the result of OR between BUS_STS, CALC_STS, and DRAW_STS.

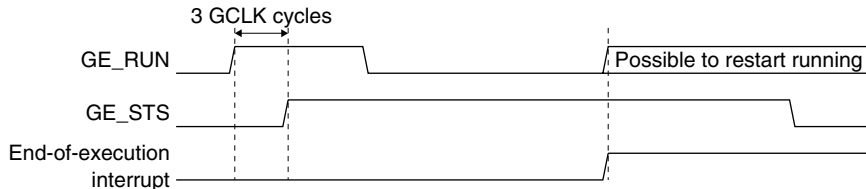


Figure 27.6.1 GE_STS Set Timing

D[15:11] Reserved

D10 BUS_STS: Bus Operation Status Bit

Indicates the GE bus operation status.

1 (R): Running

0 (R): Stopped (default)

BUS_STS indicates the GE bus operation status. It is set to 1 while the GE is fetching a command, reading font or image data, or writing data to the working area, memory, or USIL. It is reset to 0 upon completion of the above bus operation.

Note: BUS_STS is set after lapse of 3 cycles from a trigger by GE_RUN. To avoid reading undefined BUS_STS, insert three or more “nop” instructions after setting GE_RUN to 1.

D9 CALC_STS: Calculator Operation Status Bit

Indicates the GE calculator status.

1 (R): Running

0 (R): Stopped (default)

CALC_STS is set to 1 when the GE starts initialization or when the GE starts checking command parameters. It is reset to 0 after the GE has finished initialization or the parameter check.

D8 DRAW_STS: Pixel Drawing Status Bit

Indicates the VRAM writing status.

1 (R): Running

0 (R): Stopped (default)

DRAW_STS is set to 1 when the GE starts writing to the work area (VRAM) and is reset to 0 after the writing has finished.

D[7:4] Reserved

D3 GE_STOP: GE Stop Control Bit

Forcibly terminates command execution.

1 (R/W): Stop

0 (R/W): – (default)

When 1 is written to GE_STOP, the GE stops executing after the current command being executed has finished. GE_STOP is automatically reset to 0 after the command execution has finished.

D2 GE_RUN: GE Run Control Bit

Starts command execution.

1 (R/W): Run

0 (R/W): – (default)

When 1 is written to GE_RUN, the GE starts executing the command list beginning with the command start address set in the GE_CMD_ADDR register. GE_RUN is automatically reset to 0 when the GE starts fetching the 1st command.

D1 GE_HRST: GE Hot Reset Control Bit

Hot resets the GE.

1 (R/W): Reset

0 (R/W): Normal mode (default)

Hot reset initializes the GE with the command and work area address information maintained. The bus is relinquished. Perform hot reset if a command execution is terminated due to an unexpected operation. To perform hot reset, write 1 to GE_HRST. GE_HRST retains 1 and it returns to 0 upon completion of the initialization.

The following information must be reset after a hot reset.

- Font (font configuration command)
- Clipping area (GE_CLIP_ST and GE_CLIP_END registers)

D0 GE_CRST: GE Cold Reset Control Bit

Cold resets the GE.

1 (R/W): Cold reset (default)

0 (R/W): Normal mode

Cold reset initializes the GE completely.

To perform cold reset, write 1 to GE_CRST. The reset state must be canceled by writing 0 to GE_CRST. If the initialization has not finished yet at this time, GE_CRST retains 1 and it returns to 0 upon completion of the initialization. After a cold reset, all the information must be reset.

Note that the GE_CRST cannot be set to 1 if DRAW_STS is 1 (while the GE is writing data to the work area).

GE Interrupt Enable Register (GE_IE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
GE Interrupt Enable Register (GE_IE)	0x302444 (32 bits)	D31-17	–	reserved		–	–	0 when being read.
		D16	GE_END_IE	GE end-of-execution interrupt enable	1 Enable 0 Disable	0	R/W	
		D15-9	–	reserved		–	–	0 when being read.
		D8	GE_ERR_IE0	Drawing error interrupt enable	1 Enable 0 Disable	0	R/W	
		D7-1	–	reserved		–	–	0 when being read.
		D0	GE_ERR_IE1	Calculation error interrupt enable	1 Enable 0 Disable	0	R/W	

D[31:17] Reserved**D16 GE_END_IE: GE End-of-Execution Interrupt Enable Bit**

Enables or disables interrupts caused by termination of command execution.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

27 GRAPHICS ENGINE (GE)

Setting GE_END_IE to 1 enables end-of-execution interrupt requests to the ITC; setting to 0 disables interrupts.

D[15:9] Reserved

D8 GE_ERR_IE0: Drawing Error Interrupt Enable Bit

Enables or disables interrupts caused by a drawing error.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting GE_ERR_IE0 to 1 enables drawing error interrupt requests to the ITC; setting to 0 disables interrupts.

D[7:1] Reserved

D0 GE_ERR_IE1: Calculation Error Interrupt Enable Bit

Enables or disables interrupts caused by a calculation error.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Setting GE_ERR_IE1 to 1 enables calculation error interrupt requests to the ITC; setting to 0 disables interrupts.

GE Interrupt Flag Register 1 (GE_IF1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
GE Interrupt Flag Register 1 (GE_IF1)	0x302448 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.	
		D3–0	CALC_ERR[3:0]	Calculation error status	CALC_ERR[3:0]	Error	0x0	R/W	Reset by writing 0x0.
						0xf–0x9	reserved		
						0x8	Decompress		
						0x7	Picture header		
						0x6	Font size		
						0x5	Radius		
						0x4	Circle location		
						0x3	Concave		
						0x2	Coordinates		

D[7:4] Reserved

D[3:0] CALC_ERR[3:0]: Calculation Error Status Bits

Indicates whether an error has occurred in the calculation stage or not.

Table 27.6.2 List of Calculation Errors

CALC_ERR[3:0]	Error	
0xf to 0x9	Reserved	–
0x8	Decompress	A Decompress flag (0x80) exists in the image data specified by a decompression command. *3
0x7	Picture header	The header of the image data specified by a decompression command is invalid.
0x6	Font size	The character width or height specified by a character command is 0.
0x5	Radius	The radius specified by a circle/solid filled circle command is invalid (or less than 1).
0x4	Circle location	No arc/circular section to be drawn is specified in a circle/solid filled circle command.
0x3	Concave	An illegal quadrilateral (concave) is specified in a solid filled quadrilateral command. *1 *2
0x2	Coordinates	The area specified as the copy source in a copy command or clipping area in a register configuration command is beyond the work area.
0x1	Trace width	An illegal line width (less than 1 or a value that exceeds the object size) is specified in a object (line) drawing command.
0x0	No error	No error has occurred.

- *1 A calculation error and a drawing error (No VRAM write error) have occurred at the same time under the conditions shown below.
- When a Concave error (CALC_ERR[3:0] = 0x3) has occurred or invalid coordinates specification such as a counterclockwise direction has been performed while calculation error interrupts are enabled (GE_ERR_IE1/GE_IE register = 1)
 - When a Decompress error (CALC_ERR[3:0] = 0x8) has occurred and no previous image data does not exist in the VRAM

No drawing error will occur when another calculation error has occurred.

- *2 Concave errors can be detected only when calculation error interrupts are enabled (GE_ERR_IE1 = 1). Other calculation errors can always be detected regardless of the GE_ERR_IE1 setting.
- *3 The image data in which an error occurs will be drawn with the last valid color.

When a bit in CALC_ERR[3:0] is set, a calculation error interrupt request is output to the ITC if GE_ERR_IE1/GE_IE register has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied. CALC_ERR[3:0] is cleared to 0x0 by writing 0x0 or 0xff to address 0x302448.

Note: Occurrence of a calculation error does not terminate command execution if calculation error interrupts are disabled (GE_ERR_IE1 = 0). Although the command in which an error has occurred is not executed, the subsequent commands are executed normally. In this case, CALC_ERR[3:0] is not cleared. Be sure to clear CALC_ERR[3:0] before executing the subsequent command list.

When calculation error interrupts are enabled (GE_ERR_IE1 = 1), occurrence of an error terminates command execution. In this case, no end-of-execution interrupt will occur.

GE Interrupt Flag Register 2 (GE_IF2)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks	
GE Interrupt Flag Register 2 (GE_IF2)	0x302449 (8 bits)	D7-4	—	reserved		—	—	—	—	0 when being read.	
		D3	DRAW_ERR3	No VRAM write error flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D2	DRAW_ERR2	Color depth over error flag	1	Occurred	0	Not occurred	0	R/W	
		D1	DRAW_ERR1	Clipping area over error flag	1	Occurred	0	Not occurred	0	R/W	
		D0	DRAW_ERR0	Work area over error flag	1	Occurred	0	Not occurred	0	R/W	

D[7:4] Reserved

D3 DRAW_ERR3: No VRAM Write Error Flag Bit

Indicates whether a no VRAM write error has occurred in the drawing stage or not.

- 1 (R): Error has occurred
 0 (R): No error has occurred (default)
 1 (W): Flag is reset
 0 (W): Ignored

DRAW_ERR3 is set to 1 when drawing data has not been written to the VRAM. The following shows leading causes:

- All the lines of the specified object are located out of the drawing area (except when a block transfer command or a drawing command with rewrite effect is out of the range).
- The object/image has only the transparent color (except when a copy command or a block transfer command is executed, or the writing effect of rewrite or mesh specifies the transparent color).
- A concave error (CALC_ERR[3:0] = 0x3)* has occurred or invalid coordinates specification such as a counterclockwise direction has been performed when calculation error interrupts are enabled (GE_ERR_IE1/GE_IE register = 1).

(* No drawing error occurs when another calculation error has occurred.)

When DRAW_ERR3 is set to 1, a drawing error interrupt request is output to the ITC if GE_ERR_IE0 has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied.

DRAW_ERR3 is reset by writing 1.

The GE does not stop command execution even if this error has occurred.

D2 **DRAW_ERR2: Color Depth Over Error Flag Bit**

Indicates whether a color depth over error has occurred in the drawing stage or not.

1 (R): Error has occurred

0 (R): No error has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

DRAW_ERR2 is set to 1 if the specified drawing color or a pixel color in the image data is out of the effective range for the bpp mode set in the GE.

When DRAW_ERR2 is set to 1, a drawing error interrupt request is output to the ITC if GE_ERR_IE0 has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied.

DRAW_ERR2 is reset by writing 1.

The GE does not stop command execution even if this error has occurred.

D1 **DRAW_ERR1: Clipping Area Over Error Flag Bit**

Indicates whether a clipping area over error has occurred in the drawing stage or not.

1 (R): Error has occurred

0 (R): No error has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

DRAW_ERR1 is set to 1 if the whole of an object/image drawn by a command with clipping area enabled is located outside the clipping area. If a part of an object/image is beyond the clipping area, the drawing within the clipping area is performed without an error.

When DRAW_ERR1 is set to 1, a drawing error interrupt request is output to the ITC if GE_ERR_IE0 has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied.

DRAW_ERR1 is reset by writing 1.

The GE does not stop command execution even if this error has occurred.

D0 **DRAW_ERR0: Work Area Over Error Flag Bit**

Indicates whether a work area over error has occurred in the drawing stage or not.

1 (R): Error has occurred

0 (R): No error has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

DRAW_ERR0 is set to 1 if the whole of an object/image drawn by a command with clipping area disabled is located outside the work area. If a part of an object/image is beyond the work area, the drawing within the work area is performed without an error.

When DRAW_ERR0 is set to 1, a drawing error interrupt request is output to the ITC if GE_ERR_IE0 has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied.

DRAW_ERR0 is reset by writing 1.

The GE does not stop command execution even if this error has occurred.

GE Interrupt Flag Register 3 (GE_IF3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
GE Interrupt Flag Register 3 (GE_IF3)	0x30244a (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.	
		D2–0	EXE_END [2:0]	Cause of termination	EXE_END[2:0]	Cause	0x0	R/W	Reset by writing 0x0.
						0x7	Unexpected end		
						0x6	Undefined CMD		
						0x5	Software		
						0x4	STOP4 CMD		
						0x3	STOP3 CMD		
						0x2	STOP2 CMD		
						0x1	STOP1 CMD		
				0x0	Not completed				

D[7:3] Reserved

D[2:0] EXE_END[2:0]: Cause of Termination Bits

Indicates the cause that has terminated command list execution.

Table 27.6.3 Cause of Termination

EXE_END[2:0]	Cause of termination	
0x7	Unexpected operation	The command execution has been terminated due to an unexpected operation. In this case, the GE must be reset (hot reset).
0x6	Undefined command	The GE has fetched an undefined command.
0x5	Software	The command execution has been suspended by writing 1 to GE_STOP/GE_CTL register.
0x4	STOP4 command	The command execution has been suspended by executing a STOP (STOP1 to STOP4) command.
0x3	STOP3 command	
0x2	STOP2 command	
0x1	STOP1 command	
0x0	Not terminated	–

(Default: 0x0)

When a bit in EXE_END[2:0] is set, an end-of-execution interrupt request is output to the ITC if GE_END_IE/GE_IE register has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and C33 PE Core interrupt conditions are satisfied. EXE_END[2:0] is cleared to 0x0 by writing 0x0 or 0xff to address 0x30244a.

VRAM Work Area Width Register (GE_REAL_W)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
VRAM Work Area Width Register (GE_REAL_W)	0x30244c (32 bits)	D31–13	–	reserved	–	–	–	0 when being read.
		D12–0	REAL_WIDTH [12:0]	Rotated work area width	Width = REAL_WIDTH (pixels)	0x20	R	

D[31:13] Reserved

D[12:0] REAL_WIDTH[12:0]: Rotated Work Area Width Bits

Indicates the width of the work area after rotation in number of pixels. (Default: 0x20)

Width = REAL_WIDTH (pixels)

VRAM Work Area Start Address Register (GE_WK_ADDR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
VRAM Work Area Start Address Register (GE_WK_ADDR)	0x302450 (32 bits)	D31–10	VWIN_ADDR[31:10]	Work area start address (1KB boundary address)	Areas 3–5, 7–10, 13–16, and 19–22	0x0	R/W	
		D9–0	VWIN_ADDR[9:0]	Fixed at 0x0 (Cannot be altered.)			R	

D[31:0] VWIN_ADDR[31:0]: Work Area Start Address Bits

Specifies the memory address from which the work area begins. (Default: 0x0)

The work area start address must be a 1K-byte boundary address.

VRAM Work Area Size Register (GE_WK_SIZE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
VRAM Work Area Size Register (GE_WK_SIZE)	0x302454 (32 bits)	D31–28	–	reserved	–	–	–	0 when being read.
		D27–16	VWIN_H [11:0]	Work area height	Height = VWIN_H + 1 (pixels)	0x0	R/W	
		D15–12	–	reserved	–	–	–	0 when being read.
		D11–0	VWIN_W [11:0]	Work area width	Width = VWIN_W + 1 (pixels)	0x1f	R/W	

D[31:28] Reserved

D[27:16] VWIN_H[11:0]: Work Area Height Bits

Specifies the height of the work area in number of pixels. (Default: 0x0)

Height = VWIN_H + 1 (pixels)

D[15:12] Reserved

D[11:0] VWIN_W[11:0]: Work Area Width Bits

Specifies the width of the work area in number of pixels. (Default: 0x1f)

Width = VWIN_W + 1 (pixels)

Notes:

- The work area cannot be configured beyond 4,096 pixels × 4,096 pixels even if an external large capacity VRAM is connected.

- The work area width (data size) must be specified in word (32 bits) increments. If an incorrect value is specified, the GE corrects it to the word (32 bits) boundary value larger and closest to the specified value.

The data size depends on the bpp mode. Furthermore, when the work area is rotated, its width and height must be reset according to the rotation angle. Therefore, VWIN_W[11:0] and VWIN_H[11:0] should be set after setting the bpp mode and the angle of work area rotation.

Table 27.6.4 Work Area Size Specifications

bpp mode	VWIN_W[11:0]	VWIN_H[11:0]
1 bpp	$\text{int}((\text{Width} + 31) / 32) \times 32 - 1$	Height - 1
2 bpp	$\text{int}((\text{Width} + 15) / 16) \times 16 - 1$	Height - 1
4 bpp	$\text{int}((\text{Width} + 7) / 8) \times 8 - 1$	Height - 1
8 bpp	$\text{int}((\text{Width} + 3) / 4) \times 4 - 1$	Height - 1
16 bpp	$\text{int}((\text{Width} + 1) / 2) \times 2 - 1$	Height - 1

The Width and Height mean the maximum logical coordinate values in Figure 27.3.1.2.

Display Configuration Register (GE_DISP_CFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Display Configuration Register (GE_DISP_CFG)	0x302458 (32 bits)	D31–17	–	reserved	–	–	–	0 when being read.
		D16	TF_TYPE	Block transfer type select	1 Pixel to Byte 0 Byte to Byte	0	R/W	
		D15–5	–	reserved	–	–	–	0 when being read.
		D4	SYNC_TYPE	LCDC horizontal/vertical sync type select	1 Vertical 0 Horizontal	0	R/W	
		D3	–	reserved	–	–	–	–
	D2–0	DISP_BPP[2:0]	Color depth (Display data bit per pixel)	DISP_BPP[2:0]	Color depth	0x0	R/W	
				0x7–0x5	reserved			
				0x4	16 bpp			
				0x3	8 bpp			
				0x2	4 bpp			
				0x1	2 bpp			
				0x0	1 bpp			

D[31:17] Reserved

D16 TF_TYPE: Block Transfer Type Select Bit

Selects the transfer data type for block transfer from the work area to USIL.

1 (R/W): Pixel to Byte

0 (R/W): Byte to Byte (default)

Table 27.6.5 Source and Destination Data Types

TF_TYPE	GE Condition	USIL Condition	Source display data	Destination display data
0 (Byte to Byte)	1 bpp	LCD SPI 8-bit mode or LCD parallel mode	P7 P6 P5 P4 P3 P2 P1 P0	P7 P6 P5 P4 P3 P2 P1 P0
	2 bpp		P3 P2 P1 P0	P3 P2 P1 P0
	4 bpp		P1 P0	P1 P0
	8 bpp		P0	P0
0 (Byte to Byte)	8 bpp *	LCD SPI 24-bit mode or LCD parallel mode	P2 P1 P0	P0: R7 R6 R5 R4 R3 R2 R1 R0 G7 G6 G5 G4 G3 G2 G1 G0 B7 B6 B5 B4 B3 B2 B1 B0
			P2 P1 P0	P0: R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
	16 bpp	LCD SPI 16-bit mode or LCD parallel mode	P0: R4 R3 R2 R1 R0 G5 G4 G3 G2 G1 G0 B4 B3 B2 B1 B0	P0: R4 R3 R2 R1 R0 G5 G4 G3 G2 G1 G0 B4 B3 B2 B1 B0
			P0: R4 R3 R2 R1 R0 G5 G4 G3 G2 G1 G0 B4 B3 B2 B1 B0	P0: R4 R3 R2 R1 R0 X G5 G4 G3 G2 G1 G0 B4 B3 B2 B1 B0 X
1 (Pixel to Byte)	1 bpp	LCD SPI 8-bit mode or LCD parallel mode	D0	D0 D0 D0 D0 D0 D0 D0 D0
	2 bpp		D1 D0	D1 D0 D1 D0 D1 D0 D1 D0
	4 bpp		D3 D2 D1 D0	D3 D2 D1 D0 D3 D2 D1 D0
	8 bpp		D7 D6 D5 D4 D3 D2 D1 D0	D7 D6 D5 D4 D3 D2 D1 D0
	16 bpp		R4 R3 R2 R1 R0 G5 G4 G3 G2 G1 G0 B4 B3 B2 B1 B0	R4 R3 R2 G5 G4 G3 B4 B3

* In this mode, GE cannot draw anything, as RGB data is regarded as 3-pixel data not for 1 pixel. However, the BLKCOPY command can be executed normally.

A 5-bpp LCD panel with SPI/parallel I/F can be used via USIL by setting TF_TYPE to 1 (Pixel to Byte) and the color depth to 1 to 8 bpp.

D[15:5] Reserved

D4 SYNC_TYPE: LCDC Horizontal/Vertical Sync Type Select Bit

Selects either horizontal or vertical non-display period as the drawing period when the LCDC synchronization function is enabled.

1 (R/W): Vertical non-display period

0 (R/W): Horizontal non-display period (default)

When the LCDC synchronization function is enabled in a drawing command, the GE starts drawing objects/images (writing to the work area) in sync with the LCD non-display period start timings and drawings are performed in the non-display periods selected using SYNC_TYPE. This makes it possible to eliminate screen flicker. Note, however, that drawing time may exceed the non-display period if it is complicated.

When LCDC synchronization function is disabled, the GE draws objects/images without a delay. In this case, the SYNC_TYPE setting is ineffective.

D3 Reserved

D[2:0] DISP_BPP[2:0]: Color Depth Bits

Selects the color depth (bpp mode) of image data to be written to the work area (VRAM).

Table 27.6.6 Color Depth Settings

DISP_BPP[2:0]	Color depth
0x7 to 0x5	Reserved
0x4	16 bpp (65,536 colors)
0x3	8 bpp (256 colors)
0x2	4 bpp (16 colors/16-level gray scale)
0x1	2 bpp (4 colors/4-level gray scale)
0x0	1 bpp (2 colors/2-level gray scale)

(Default: 0x0)

VRAM Rotation Control Register (GE_ROTATE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
VRAM Rotation Control Register (GE_ROTATE)	0x30245c (32 bits)	D31–2	–	reserved	–	–	–	0 when being read.	
		D1–0	VWIN_ROT[1:0]	VRAM rotation select	VWIN_ROT[1:0] Rotation angle	0x0	R/W		
					0x3	270°			
					0x2	180°			
					0x1	90°			
				0x0	0°				

D[31:2] Reserved**D[1:0] VWIN_ROT[1:0]: VRAM Rotation Select Bits**

Selects an angle of VRAM rotation.

Table 27.6.7 Angle of VRAM Rotation

VWIN_ROT[1:0]	Rotation angle
0x3	270°
0x2	180°
0x1	90°
0x0	0°

(Default: 0x0)

Changing this setting does not affect the contents of the work area. To rotate the image according to the LCD panel, configure the work area and execute the drawing commands again.

Clipping Area Start Position Register (GE_CLIP_ST)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clipping Area Start Position Register (GE_CLIP_ST)	0x302460 (32 bits)	D31–28	–	reserved	–	–	–	0 when being read.
		D27–16	CLIP_UPL_Y[11:0]	Clipping area upper left corner Y coordinate	0 to 4,095	0x0	R/W	
		D15–12	–	reserved	–	–	–	0 when being read.
		D11–0	CLIP_UPL_X[11:0]	Clipping area upper left corner X coordinate	0 to 4,095	0x0	R/W	

D[31:28] Reserved**D[27:16] CLIP_UPL_Y[11:0]: Clipping Area Upper Left Corner Y Coordinate Bits**

Specifies the Y coordinate value of the clipping area upper left corner. (Default: 0x0)

Each drawing command can enable/disable the clipping function as a drawing effect. When a command with clipping enabled is executed, the object/image specified by the command is drawn only within the clipping area that has been configured in advance. Drawing outside the clipping area is masked even if it is within the work area.

To configure a clipping area, write the coordinate values of the upper left and lower right corners to the GE_CLIP_ST and GE_CLIP_END registers, respectively.

Configuring a clipping area using these registers does not enable the clipping function. Enable it in the drawing commands to perform clipping.

If the whole of an object/image drawn by a command is located outside the clipping area when the clipping function is enabled, a clipping area over error occurs (an interrupt can be generated). If only a part of an object/image is beyond the clipping area, the drawing within the clipping area is performed without an error.

- Notes:**
- When configuring a clipping area, the X and Y coordinates of the upper left corner must be set first. Otherwise, proper operations cannot be guaranteed.
 - Do not configure a clipping area that exceeds the work area, as proper drawing results cannot be guaranteed.
 - When the GE_CONFIG command is used to set the GE_CLIP_ST register, the setting value is not written to the register until the GE_CONFIG command for setting the GE_CLIP_END register is executed.

D[15:12] Reserved

D[11:0] CLIP_UPL_X[11:0]: Clipping Area Upper Left Corner X Coordinate Bits

Specifies the X coordinate value of the clipping area upper left corner. (Default: 0x0)

Clipping Area End Position Register (GE_CLIP_END)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clipping Area End Position Register (GE_CLIP_END)	0x302464 (32 bits)	D31–28	–	reserved	–	–	–	0 when being read.
		D27–16	CLIP_LWR_Y[11:0]	Clipping area lower right corner Y coordinate	0 to 4,095	0x0	R/W	
		D15–12	–	reserved	–	–	–	0 when being read.
		D11–0	CLIP_LWR_X[11:0]	Clipping area lower right corner X coordinate	0 to 4,095	0x0	R/W	

D[31:28] Reserved

D[27:16] CLIP_LWR_Y[11:0]: Clipping Area Lower Right Corner Y Coordinate Bits

Specifies the Y coordinate value of the clipping area lower right corner. (Default: 0x0)

D[15:12] Reserved

D[11:0] CLIP_LWR_X[11:0]: Clipping Area Lower Right Corner X Coordinate Bits

Specifies the X coordinate value of the clipping area lower right corner. (Default: 0x0)

Mesh Configuration Register (GE_MESH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Mesh Configuration Register (GE_MESH)	0x302468 (32 bits)	D31–20	–	reserved	–	–	–	0 when being read.
		D19–18	MESH_RW[1:0]	Mesh row width	MESH_RW[1:0] Width 0x3 4 pixels 0x2 reserved 0x1 2 pixels 0x0 1 pixel	0x0	R/W	
		D17–16	MESH_CW[1:0]	Mesh column width	MESH_CW[1:0] Width 0x3 4 pixels 0x2 reserved 0x1 2 pixels 0x0 1 pixel	0x0	R/W	
		D15–0	MESH_COL[15:0]	Mesh color	0x0 to 0xffff	0x0	R/W	

D[31:20] Reserved

D[19:18] MESH_RW[1:0]: Mesh Row Width Bits

Selects the row width of the mesh.

Enabling the mesh effect in a drawing command overlays the mesh pattern configured in this register on the object when written to the work area. The mesh pattern is generated relative to the work area origin (0, 0). Therefore, a continuous mesh pattern will be generated even if two or more objects with mesh specified overlap one another.

Table 27.6.8 Mesh Pattern Settings

MESH_RW[1:0]/MESH_CW[1:0]	Mesh size
0x3	4 pixels
0x2	Reserved
0x1	2 pixels
0x0	1 pixel

(Default: 0x0)

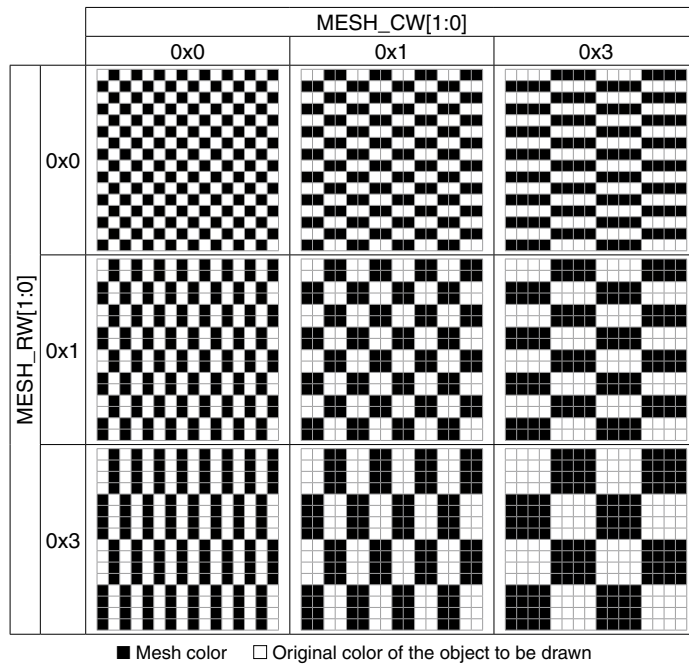


Figure 27.6.2 List of Mesh Patterns

D[17:16] MESH_CW[1:0]: Mesh Column Width Bits

Selects the column width of the mesh.

D[15:0] MESH_COL[15:0]: Mesh Color Bits

Specifies the mesh color. (Default: 0x0)

A color within the effective range for the bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register) can only be specified.

Transparent Color Register (GE_MAGIC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Transparent Color Register (GE_MAGIC)	0x30246c (32 bits)	D31–16	–	reserved	–	–	–	0 when being read.
		D15–0	MAGIC_COL[15:0]	Transparent color	0x0 to 0xffff	0x0	R/W	

D[31:16] Reserved

D[15:0] MAGIC_COL[15:0]: Transparent Color Bits

Specifies the transparent color. (Default: 0x0)

A color within the effective range for the bpp mode (set using DISP_BPP[2:0]/GE_DISP_CFG register) can only be specified.

When transparency is enabled in a drawing command, the pixels with the transparent color specified in objects/images are not drawn on the work area and the current pixel color is left unchanged.

Updated Area Start Position Register (GE_UPDT_ST)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Updated Area Start Position Register (GE_UPDT_ST)	0x302470 (32 bits)	D31–28	–	reserved	–	–	–	0 when being read.
		D27–16	UPDT_UPL_Y[11:0]	Updated area upper left corner Y coordinate	0 to 4,095	0xff	R/W	Cleared by writing any data.
		D15–12	–	reserved	–	–	–	0 when being read.
		D11–0	UPDT_UPL_X[11:0]	Updated area upper left corner X coordinate	0 to 4,095	0xff	R/W	Cleared by writing any data.

D[31:28] Reserved

D[27:16] UPDT_UPL_Y[11:0]: Updated Area Upper Left Corner Y Coordinate Bits

Indicates the Y coordinate value of the updated area upper left corner. (Default: 0xffff)

The area that has been rewritten by a drawing command can be checked using the GE_UPDT_ST and GE_UPDT_END registers. These registers retain the upper left and lower right coordinate values of the rectangular region that contains the objects/images that have been drawn.

The GE_UPDT_ST and GE_UPDT_END registers are cleared by writing any value.

D[15:12] Reserved

D[11:0] UPDT_UPL_X[11:0]: Updated Area Upper Left Corner X Coordinate Bits

Indicates the X coordinate value of the updated area upper left corner. (Default: 0xfff)

- Notes:**
- The coordinates of the updated area can be obtained in the following cases:
 - At least one opaque dot of the specified object is located in the drawing area.
 - As for a transparent object in the drawing area, a copy command or a block transfer command is executed, or the writing effect of rewrite or mesh specifies the transparent color.
 - The coordinates specified for drawing are logical values without consideration of the rotation angle set by VWIN_ROT[1:0]/GE_ROTATE register. The GE converts the logical coordinates into physical values according to the rotation information before it writes to the VRAM. The update area coordinates are physical values after being rotated, therefore the update area coordinates may differ from those specified by the commands.

Updated Area End Position Register (GE_UPDT_END)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Updated Area End Position Register (GE_UPDT_END)	0x302474 (32 bits)	D31–28	–	reserved	–	–	–	0 when being read.
		D27–16	UPDT_LWR_Y[11:0]	Updated area lower right corner Y coordinate	0 to 4,095	0x0	R/W	Cleared by writing any data.
		D15–12	–	reserved	–	–	–	0 when being read.
		D11–0	UPDT_LWR_X[11:0]	Updated area lower right corner X coordinate	0 to 4,095	0x0	R/W	Cleared by writing any data.

D[31:28] Reserved

D[27:16] UPDT_LWR_Y[11:0]: Updated Area Lower Right Corner Y Coordinate Bits

Indicates the Y coordinate value of the updated area lower right corner. (Default: 0x0)

D[15:12] Reserved

D[11:0] UPDT_LWR_X[11:0]: Updated Area Lower Right Corner X Coordinate Bits

Indicates the X coordinate value of the updated area lower right corner. (Default: 0x0)

Palette 1 (GE_PALETTE1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Palette 1 (GE_PALETTE1)	0x302800 0x3028ff (8 bits)	D7–0	–	1/2/4/8-bit color data	0x0 to 0xff	X	R/W	

Specifying Palette 1 in a drawing command substitutes the color defined in the palette for the drawing color specified in the command or original color in the image data when drawing in 1/2/4/8 bpp mode.

Palette 1 is a 256-byte table and is located from address 0x302800 to 0x3028ff. The entries of 256 bytes correspond to original color data 0 to 255. Write the color data by which the original color will be substituted to each entry.

16 bpp mode does not use Palette 1 even if it is specified.

CCT1 4-bit Entries (GE_CCT1_4BIT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CCT1 4-bit Entries (GE_CCT1_4BIT)	0x302910 0x30291f (8 bits)	D7–0	–	CCT1 data (User defined 4 to 8 bpp conversion data)	0x0 to 0xff	X	R/W	

CCT1 (Color Conversion Table 1) is located at 22 bytes from address 0x302910. This table is used for converting color depth when the color conversion table in the header of the image data to be decompressed/drawn is disabled (bcTable = 0). Conversion data must be written to the addresses shown above in advance.

Addresses 0x302910 to 0x30291f is the 4 to 8 bpp conversion data area.

Table 27.6.9 CCT1 Configuration

Table address	4 bpp data (4 to 8 bpp conversion data)	2 bpp data (2 to 8 or 2 to 4 bpp conversion data)	1 bpp data (1 to 8, 1 to 4 or 1 to 2 bpp conversion data)	
0x302910	8-bit color data 0			
0x302911	8-bit color data 1			
0x302912	8-bit color data 2			
0x302913	8-bit color data 3			
0x302914	8-bit color data 4			
0x302915	8-bit color data 5			
0x302916	8-bit color data 6			
0x302917	8-bit color data 7			
0x302918	8-bit color data 8			
0x302919	8-bit color data 9			
0x30291a	8-bit color data 10			
0x30291b	8-bit color data 11			
0x30291c	8-bit color data 12			
0x30291d	8-bit color data 13			
0x30291e	8-bit color data 14			
0x30291f	8-bit color data 15			
0x302920		8 or 4-bit color data 0		
0x302921		8 or 4-bit color data 1		
0x302922		8 or 4-bit color data 2		
0x302923		8 or 4-bit color data 3		
0x302924				8, 4, or 2-bit color data 0
0x302925				8, 4, or 2-bit color data 1

CCT1 2-bit Entries (GE_CCT1_2BIT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CCT1 2-bit Entries (GE_CCT1_2BIT)	0x302920 0x302923 (8 bits)	D7-0	-	CCT1 data (User defined 2 to 4/8 bpp conversion data)	0x0 to 0xff	X	R/W	

Write 2 to 4 or 8 bpp conversion data to the 4 bytes of CCT1 from address 0x302920 to address 0x302923.

CCT1 1-bit Entries (GE_CCT1_1BIT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CCT1 1-bit Entries (GE_CCT1_1BIT)	0x302924 0x302925 (8 bits)	D7-0	-	CCT1 data (User defined 1 to 2/4/8 bpp conversion data)	0x0 to 0xff	X	R/W	

Write 1 to 2, 4 or 8 bpp conversion data to the 2 bytes of CCT1 from address 0x302924 to address 0x302925.

27.7 LCDC Settings

The table below lists the GE work area settings and related LCDC main window (virtual display) settings.

Table 27.7.1 GE Work Area and LCDC Main Window Settings

Setting	Equation		Remarks
	VRAM rotation = 0°, 180°	VRAM rotation = 90°, 270°	
GE	Work area width	$VWIN_W + 1$	$VWIN_H + 1$
	Work area height	$VWIN_H + 1$	$VWIN_W + 1$
	Work area start address	$VWIN_ADDR$	$VWIN_ADDR$
LCDC	Main window start address	$VWIN_ADDR + \frac{((VWIN_W + 1) \times Y0 + X0) \times \text{bpp}}{8}$	$VWIN_ADDR + \frac{((VWIN_H + 1) \times Y0 + X0) \times \text{bpp}}{8}$
	Main screen address offset	$\frac{(VWIN_W + 1) \times \text{bpp}}{32}$	$\frac{(VWIN_H + 1) \times \text{bpp}}{32}$

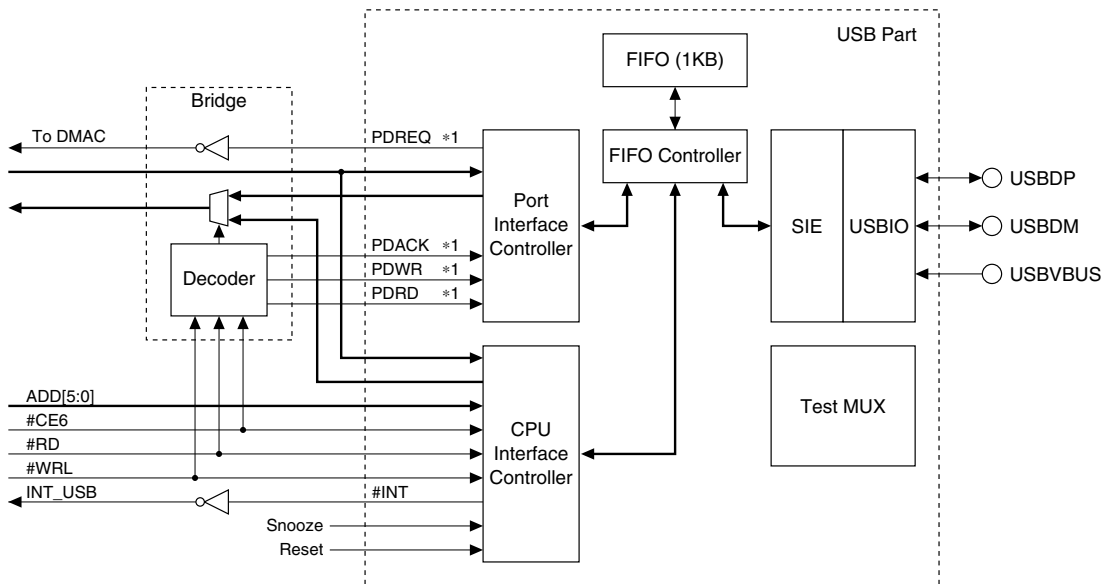
28 USB Function Controller (USB)

28.1 USB Function Controller Overview

The S1C33L26 includes a USB function controller that supports 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 28.1.1 shows the block diagram of the USB function controller.



*1 The PDREQ, PDACK, PDWR and PDRD signals level must be configured as "Active High".

Figure 28.1.1 USB Function Controller Block Diagram

Serial Interface Engine (SIE)

The SIE manages transactions and generates packets. It also controls bus events such as suspend, resume and reset operations.

FIFO

This is a 1KB buffer for endpoints.

FIFO Controller

This controller performs FIFO SRAM address management (user-programmable), timing generation, arbitration and more.

Port Interface Controller

This controller performs asynchronous handshakes.

CPU Interface Controller

This controller controls timings of the CPU interface and enables register access.

Test MUX

Switches the operational mode (test mode) using the Input signal.

28.2 Pins for the USB Interface

Table 28.2.1 list the pins used for the USB interface.

Table 28.2.1 USB Interface Pins

Pin name	I/O	Qty	Function
USBDP	I/O	1	USB D+ pin
USBDM	I/O	1	USB D- pin
USBVBUS	I	1	USB VBUS pin (Allows input of 5 V.)

28.3 USB Operating Clocks

The USB function controller operates with USBCLK supplied from the CMU.

USBREGCLK is also used for accessing the USB control registers.

By default, these two clocks are not supplied to the USB function controller. Therefore the clock supply must be enabled before running the USB function controller. For more information on clock control, see the “Clock Management Unit (CMU)” chapter.

28.4 Settings in Other Modules

Before using the USB function controller, the following modules/register must be programmed.

Clock management unit (CMU): • Control the clock supply to the USB function controller.

Misc register (MISC):

- Set the number of wait cycles to be inserted when the USB control register is accessed.
- Enable or disable the snooze control by the USB function controller.
- Enable or disable the USB interrupts.

Interrupt controller (ITC): • Set the USB interrupt level.

DMA controller (DMAC): • Program the control table for the DMAC channel to be triggered from the USB function controller.

For more information, see the respective chapters.

Note: The DMA transfer address for the USB function controller must be located in Area 6 (0x300d00 to 0x300dff, 256 bytes). Make sure that the transfer address does not exceed the address range from 0x300d00 to 0x300dff by the address increment operation during DMA transfer.

28.5 Functional Description

This section describes the functionality of the USB function controller.

In the subsequent sections, the register names follow the notational convention below:

* When a register for one address is referred to:

Register name + register.

Example: “MainInt register”

* When a discrete bit is referred to:

Register name. bit name + bit, or bit name + bit.

Example: “MainIntStat.RcvEP0SETUP bit”, or “ForceNAK bit of the EP0ControlOUT register”

* When a register present for a specific end-point is referred to:

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”

28.5.1 USB Control

Endpoints

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 28.5.1.1 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 28.5.1.1 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 28.5.1.2 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 28.5.1.2 Setting Items for Endpoints EPa, EPb, EPc and EPd

Item	Register/bit	Description
Transaction direction	EP x { $x=a,b,c,d$ }Config.INxOUT	Sets the transfer direction for each endpoint.
Max. packet size	EP x { $x=a,b,c,d$ }MaxSize_H, EP x { $x=a,b,c,d$ }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	EP x { $x=a,b,c,d$ }Config.EndPointNumber	Sets each endpoint number to any desired value between 0x1 and 0xf.
Toggle mode	EP x { $x=a,b,c,d$ }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	EP x { $x=a,b,c,d$ }Config.EnEndPoint	Enables each endpoint. Set it up when the USB-defined interface that uses the relevant endpoint is enabled.
FIFO region	EP x { $x=a,b,c,d$ }StartAdrs_H, EP x { $x=a,b,c,d$ }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 28.5.1.3 lists control items and statuses related to transaction control on the EP0 endpoint.

Table 28.5.1.3 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.OUT.ForceNAK bit whenever an OUT transaction is completed.
SETUP reception status	MainIntStat.RcvEPOSETUP	Indicates that a SETUP transaction is executed.
Transaction status	EP0IntStat.IN_TranACK, EP0IntStat.OUT_TranACK, EP0IntStat.IN_TranNAK, EP0IntStat.OUT_TranNAK, EP0IntStat.IN_TranErr, EP0IntStat.OUT_TranErr	Indicates the result of the transaction.

Table 28.5.1.4 lists control items and statuses related to transaction processing on the EPa, EPb, EPc, and EPd endpoints.

Table 28.5.1.4 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_TranACK, EPx{x=a,b,c,d}IntStat.OUT_TranACK, EPx{x=a,b,c,d}IntStat.IN_TranNAK, EPx{x=a,b,c,d}IntStat.OUT_TranNAK, EPx{x=a,b,c,d}IntStat.IN_TranErr, EPx{x=a,b,c,d}IntStat.OUT_TranErr	Indicates the result of the transaction.

SETUP transaction

The SETUP transaction addressed to the EP0 endpoint of the macro's own node is automatically executed. (The USB function must be enabled for this to happen.)

When a SETUP transaction is issued, all the contents of the data packet (8 bytes) are stored in the registers EP0Setup_0 through EP0Setup_7, followed by an ACK response. Meanwhile, a RevEP0SETUP 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 28.5.1.1 illustrates how the SETUP transaction is executed.

- (a) The host issues a SETUP token addressed to the EP0 endpoint of this node.
- (b) Next, the host sends an 8-byte long data packet. The macro writes these data in the EP0Setup_0 through EP0Setup_7 registers.
- (c) The macro automatically returns an ACK response. In addition, it sets registers to be automatically set up and issues a status to the firmware.

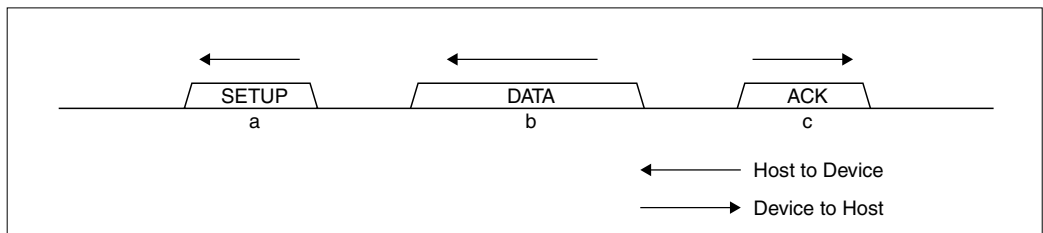


Figure 28.5.1.1 SETUP Transaction

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_TranACK status of the relevant endpoint ($EP_x\{x=0,a,b,c,d\}IntStat.OUT_TranACK$ bit). Furthermore, the FIFO is updated to acknowledge the data reception and to secure a space for the data.

In OUT transactions on the EPa, EPb, EPc, and EPd endpoints, reception of all short-packet data causes an OUT_ShortACK status ($EP_x\{x=a,b,c,d\}IntStat.OUT_ShortACK$ 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_TranErr status ($EP_x\{x=0,a,b,c,d\}IntStat.OUT_TranErr$ bit) is issued. Accordingly, the FIFO is not updated.

If not all data are received in an OUT transaction, a NAK response is returned to the transaction and the OUT_TranNAK status ($EP_x\{x=a,b,c,d\}IntStat.OUT_TranNAK$ bit) is issued. Accordingly, the FIFO is not updated.

Figure 28.5.1.2 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.

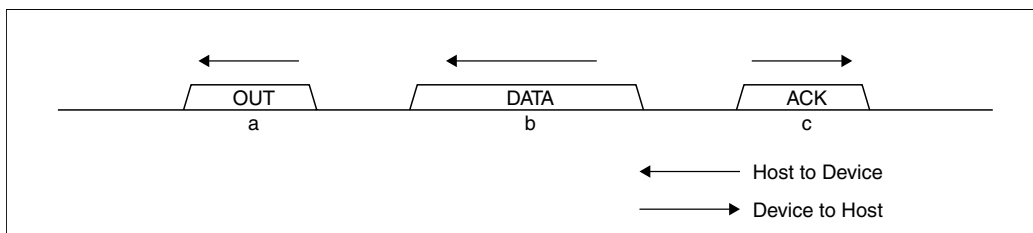


Figure 28.5.1.2 OUT Transaction

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 EP0Control.IN.EnShortPkt bit and the EP_x{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 EP0Control.IN.ForceNAK bit is set after the IN transaction that transmits the short-packet data is closed.

After an ACK response is received in the IN transaction that has returned the data, the transaction is closed, followed by issuance of an IN_TranACK status (EP_x{x=0,a,b,c,d}IntStat.IN_TranACK bit). Also, the FIFO is updated to acknowledge completion of the data transmission and to free the space.

If an ACK response is not received in the IN transaction that has returned the data, the transaction is considered as a failure, followed by issuance of an IN_TranErr status (EP_x{x=0,a,b,c,d}IntStat.IN_TranErr bit). Accordingly, the FIFO is not updated, or no space is freed.

In on an IN endpoint, if no maximum packet size data exist in the FIFO and no permission is granted for short-packet transmission, the IN transaction receives a NAK response and an IN_TranNAK status (EP_x{x=0,a,b,c,d}IntStat.IN_TranNAK bit) is issued to the firmware. Accordingly, the FIFO is not updated, or no space is freed.

Figure 28.5.1.3 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.

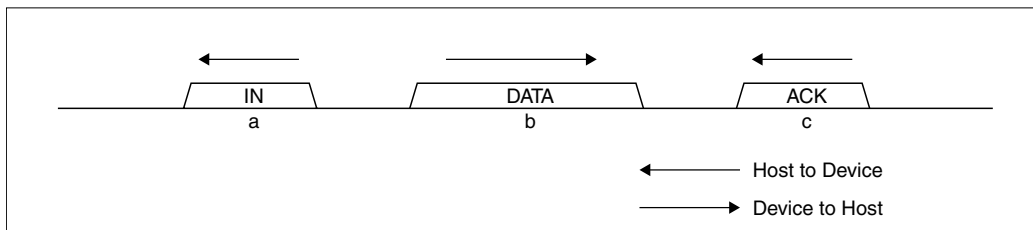


Figure 28.5.1.3 IN Transaction

Control transfer

Control transfer on the EP0 endpoint is controlled as a combination of a number of discrete transactions. Figure 28.5.1.4 illustrates how control transfer is executed for an OUT data stage.

- (a) The host starts control transfer in a SETUP transaction. The device’s firmware analyzes the request contents to prepare for responding to a data stage.
- (b) The host issues an OUT transaction and executes a data stage, and the device receives data.
- (c) The host issues an IN transaction and executes a status stage, and the device returns a zero-length data packet.

Control transfer without a data stage is executed as in this example but without the data stage.

Transition to a status stage is triggered by an issuance of a transaction by the host whose direction is opposite to that of the data stage. Have your firmware monitor an IN_TrانNAK status (EP0IntStat.IN_TrانNAK bit) as a trigger to transit to a status stage from a data stage.

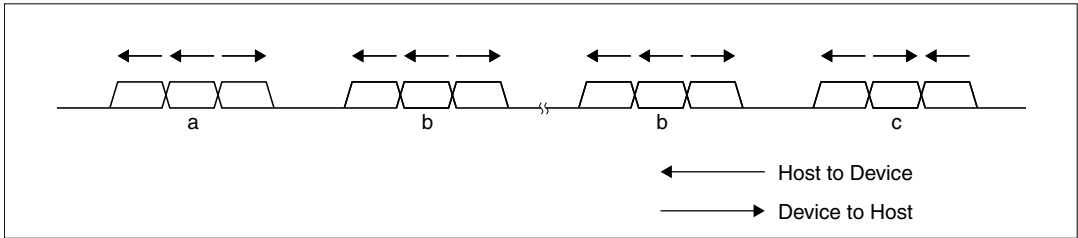


Figure 28.5.1.4 Control Transfer Having an OUT Data Stage

Figure 28.5.1.5 illustrates how control transfer is executed for an IN data stage.

(a) The host starts control transfer in a SETUP transaction. The device's firmware analyzes the request contents to prepare for responding to a data stage.

(b) The host issues an IN transaction and executes a data stage, and the device transmits data.

(c) The host issues an OUT transaction and executes a status stage, and the device returns an ACK response.

Transition to a status stage is triggered by an issuance of a transaction by the host whose direction is opposite to that of the data stage. Have your firmware monitor an OUT_TrانNAK status (EP0IntStat.OUT_TrانNAK bit) as a trigger to transit to a status stage from a data stage.

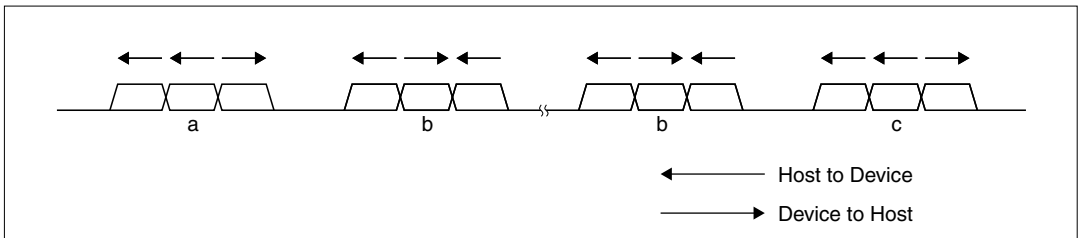


Figure 28.5.1.5 Control Transfer Having an IN Data Stage

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 canceled 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 (see the "Data flow control" section) 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 28.5.1.6 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.

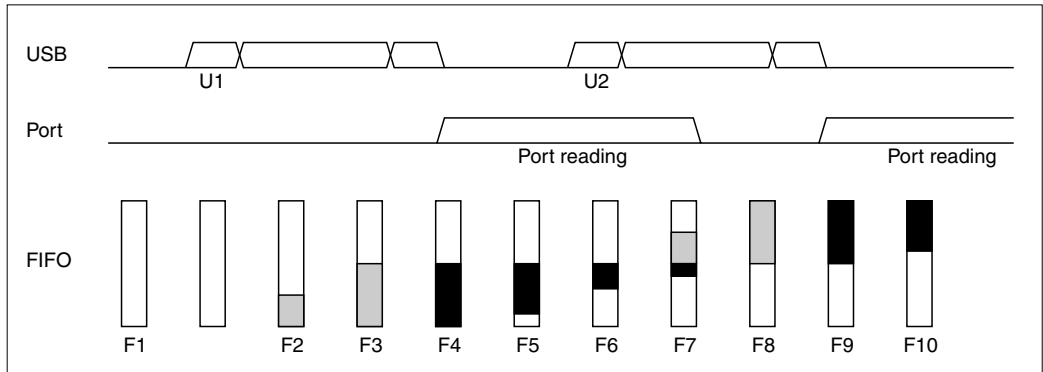


Figure 28.5.1.6 Example of Data Flow in OUT Transfer

- (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 EP_x{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 28.5.1.7 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.

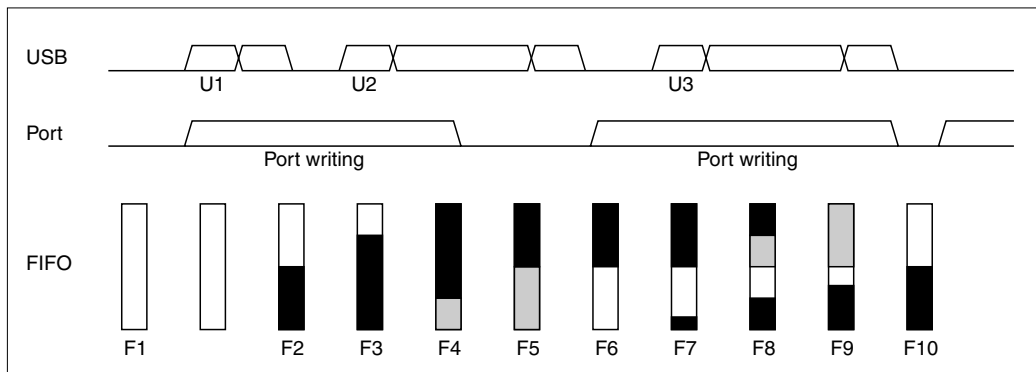


Figure 28.5.1.7 Example of Data Flow in IN Transfer

- (U1) In the first IN transaction, a 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.

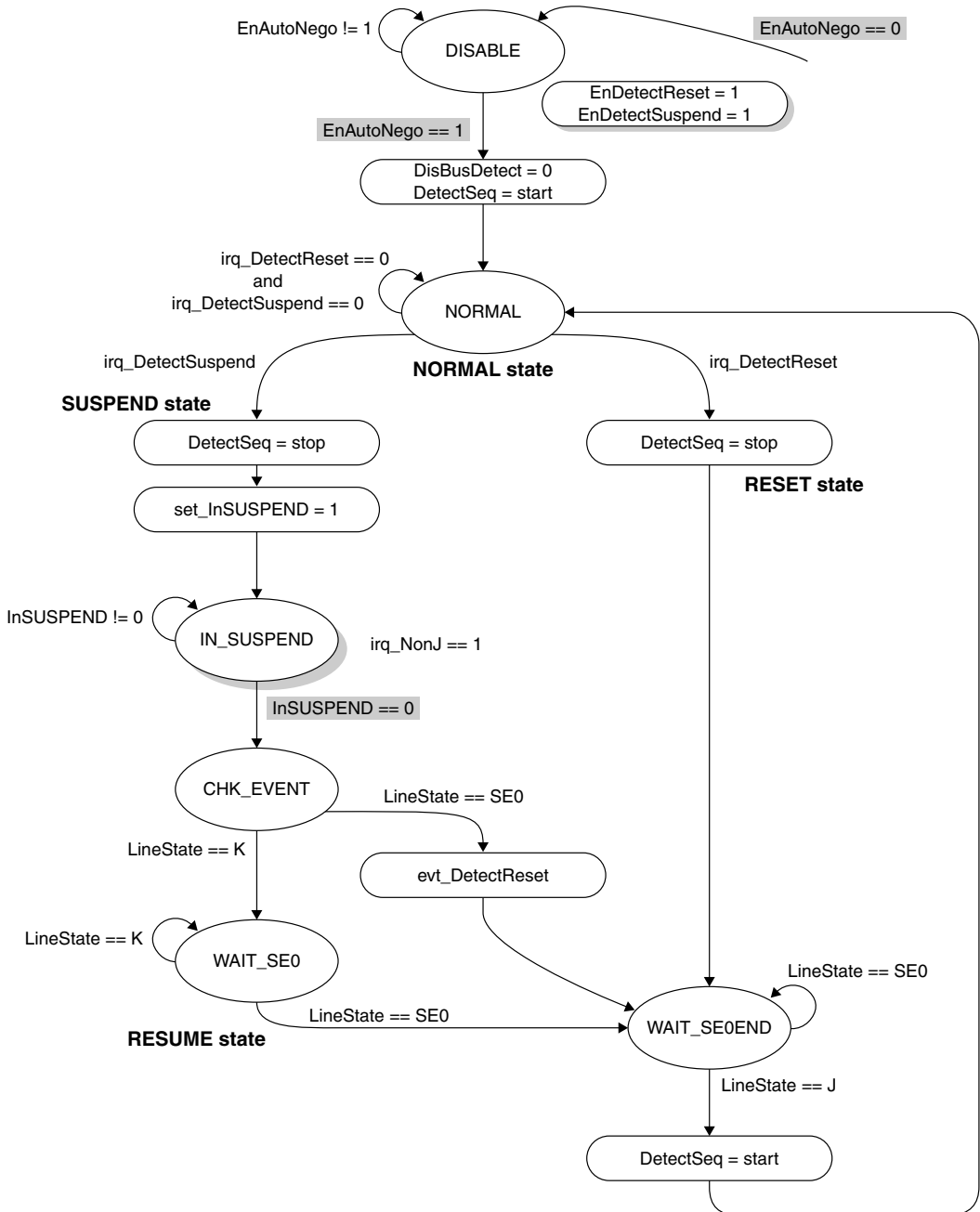


Figure 28.5.1.8 Auto-negotiator

(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/MISC_USB register to 1 to start processing Snooze before reaching T₄. As for self-powered products, however, the firmware does not have to perform Snooze. (Figure 28.5.1.9 shows the operation when Snooze is performed.)

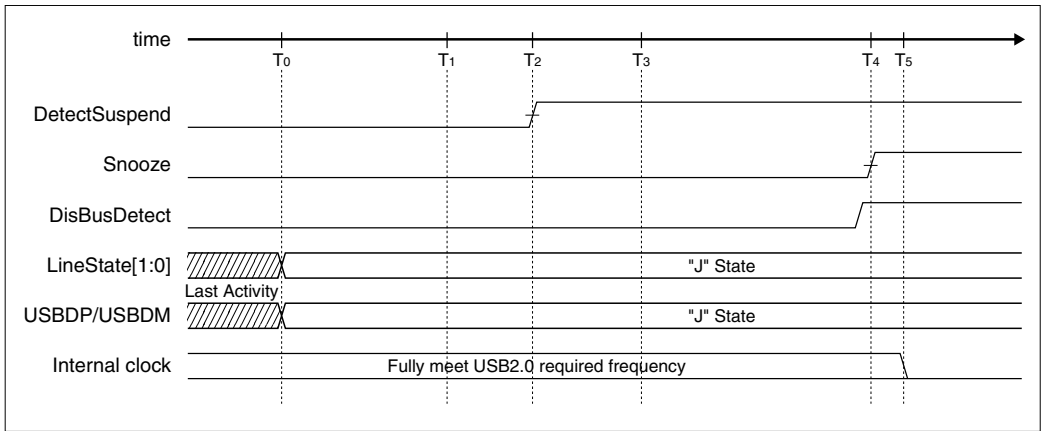


Figure 28.5.1.9 Suspend Timing (FS mode)

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 (T1).
- (2) At T2, 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.

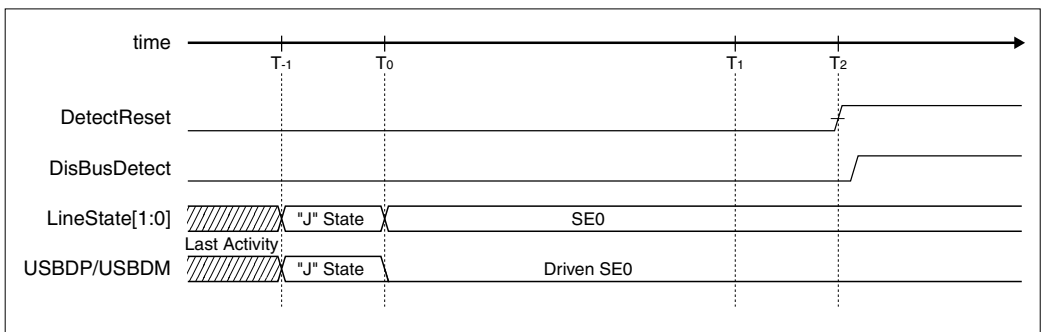


Figure 28.5.1.10 Reset Timing (FS mode)

Issuing resume

This section describes how to enable automatic resume to be triggered by some cause when remote wakeup 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 S1C33L26 supports Snooze state. This section describes the operation for issuing Resume when the oscillation circuit is in operation (USBCLK_EN/CMU_CLKCTL register = 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 bits. This is to cause this macro return from Snooze for automatic wakeup.
- (2) Set the USB_Control.SendWakeup bit and send out the Resume signal.

- (3) The macro sets XcvrControl.OpMode[1:0] to “Disable Bit Stuffing and NRZI encoding” and prepares for transmission of “All 0” data.
- (4) The macro starts data transmission and sends out “FS K” (the Resume signal) to a downstream port.
- (5) The downstream port detects this Resume signal and returns “FS K” (the Resume signal) onto the bus.
- (6) Clear the USB_Control.SendWakeup bit and suspend Resume signal send-out. After that, 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 (USBCLK_EN = 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 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.
- (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 (USBCLK_EN = 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 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 USBCLK_EN to start supplying the USB clock (T₁).
- (4) Clear USBSNZ (T₂).
- (5) The downstream port sends out Reset (T₄).

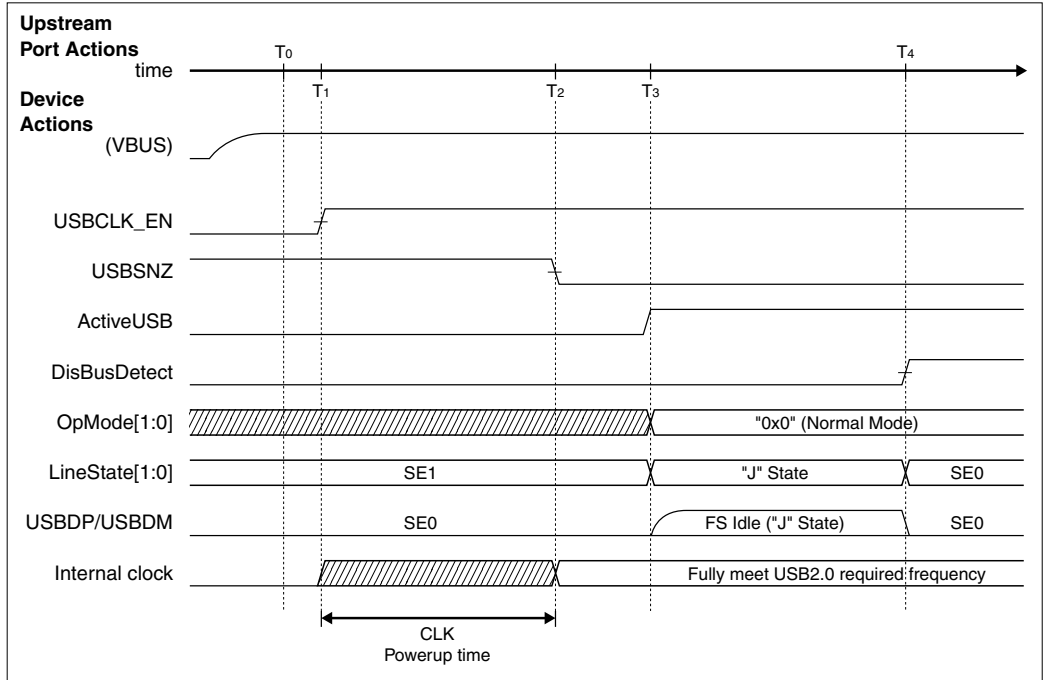


Figure 28.5.1.11 Device Attach Timing

Table 28.5.1.5 Device Attach Timing Values

Timing parameter	Description	Value
T ₀	VBUS is enabled.	0 (Reference)
T ₁	Set USBCLK_EN 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 0x0 (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

28.5.2 FIFO Management

FIFO memory map

This section describes the memory map for the FIFO region.

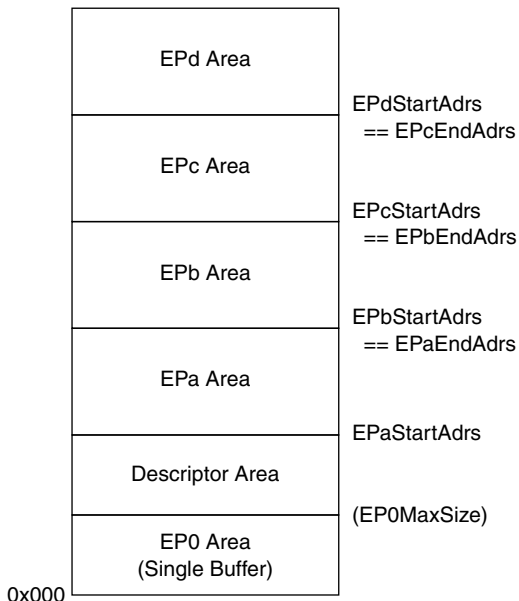


Figure 28.5.2.1 FIFO Memory Map

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.

28.5.3 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. Note that “DMA” in the descriptions refers to the DMA circuit in the USB macro and “DMAC” refers to the DMA controller module in the S1C33L26.

Register setting

Table 28.5.3.1 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 is transferring data.

Table 28.5.3.1 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 0x00000000. To cancel (negate) the DMA request (PDREQ), 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, the USB macro requests data transfer by asserting PDREQ 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 reject data transfer.

If any data is set to the DMA_Latency.DMA_Latency[3:0] bit other than 0x0, 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 0x00000000. To cancel (negate) the DMA request (PDREQ), provide 1 to the DMA_Control.DMA_Stop bit. Note that writing 1 to the DMA_Control.DMA_Stop bit does not stop the DMAC. So to terminate data transfer, first terminate the DMAC (master) and then terminate the macro's DMA transfer.

Note: The S1C33L26 DMAC can only be triggered to start data transfer by the Rising Edge of PDREQ. After that no DMAC trigger will be issued while PDREQ stays active (high level) in multi-word DMA transfer mode. The subsequent DMAC trigger will be issued at the next PDREQ Rising Edge. Therefore, when using the USB macro in multi-word DMA transfer mode, configure the DMAC in successive transfer mode and set the DMAC transfer counter to the same value set in the DMA_Remain_H and DMA_Remain_L registers.

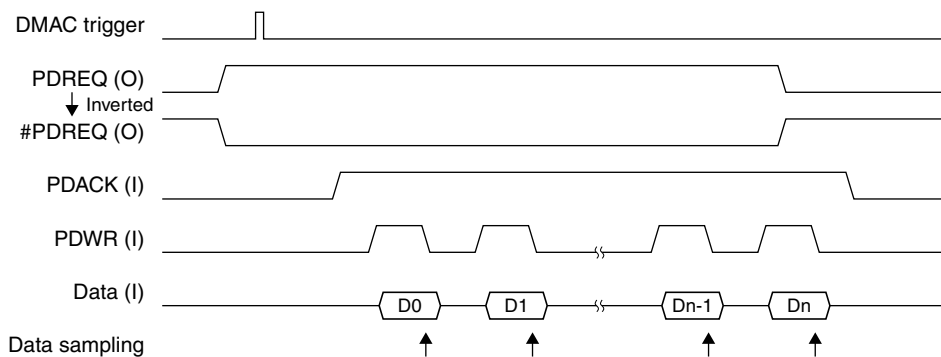


Figure 28.5.3.1 Transfer Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Writing

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.

Note: In the S1C33L26, the USB DMA data transfer count is determined according to the DMAC transfer counter setting. Negating PDREQ by the USB macro does not affect the transfer count. So in RcvLimit mode, the DMAC continues data transfer until the DMAC transfer counter reaches 0 even after the macro negates PDREQ. Therefore, make sure that the DMAC transfer counter is set properly.

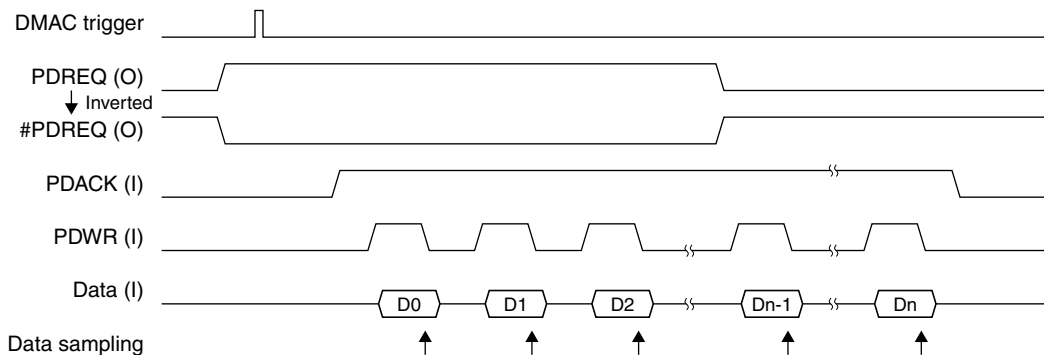


Figure 28.5.3.2 Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Writing (RcvLimit mode)

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, the USB macro requests data transfer by asserting PDREQ if any data exist at the connected endpoint. Turning PDACK to active starts outputting transferred data to the data bus. Have the DMAC (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 reject data transfer.

If any data is set to the DMA_Latency.DMA_Latency[3:0] bit other than 0x0, this mode negates PDREQ once after completing transfer of 4-byte data, and does not assert PDREQ as long as 130 ns × N (N = DMA_Latency.DMA_Latency[3:0]).

If the DMA is set to the Countdown mode with DMA_Config_1.CountMode = 1, the DMA completes data transfer when the DMA_Count_HH, HL, LH and LL registers reach 0x00000000. To cancel (negate) the DMA request (PDREQ), provide 1 to the DMA_Control.DMA_Stop bit. Note that writing 1 to the DMA_Control.DMA_Stop bit does not stop the DMAC. So to terminate data transfer, first terminate the DMAC (master) and then terminate the macro's DMA transfer.

Note: The S1C33L26 DMAC can only be triggered to start data transfer by the Rising Edge of PDREQ. After that no DMAC trigger will be issued while PDREQ stays active (high level) in multi-word DMA transfer mode. The subsequent DMAC trigger will be issued at the next PDREQ Rising Edge. Therefore, when using the USB macro in multi-word DMA transfer mode, configure the DMAC in successive transfer mode and set the DMAC transfer counter to the same value set in the DMA_Remain_H and DMA_Remain_L registers.

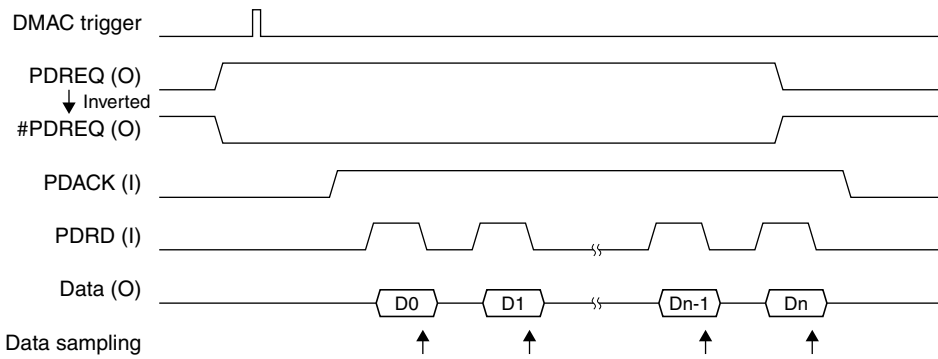


Figure 28.5.3.3 Transfer Waveforms in Asynchronous Multi-Word DMA Transfer Mode - Reading

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, the USB macro requests data transfer by asserting PDREQ 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. 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 0x0, 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 0x00000000. To cancel (negate) the DMA request (PDREQ), provide 1 to the DMA_Control.DMA_Stop bit. Note that writing 1 to the DMA_Control.DMA_Stop bit does not stop the DMAC. So to terminate data transfer, first terminate the DMAC (master) and then terminate the macro's DMA transfer.

Note: The S1C33L26 DMAC can only be triggered to start data transfer by the Rising Edge of PDREQ. The subsequent DMAC trigger will be issued at the next PDREQ Rising Edge. When the DMAC transfer counter reaches 0, DMA transfer will not be started even if a DMAC trigger is issued. Therefore, when using the USB macro in single-word DMA transfer mode, configure the DMAC in single transfer mode and set the DMAC transfer counter to a value equal to or less than that set in the DMA_Remain_H and DMA_Remain_L registers.

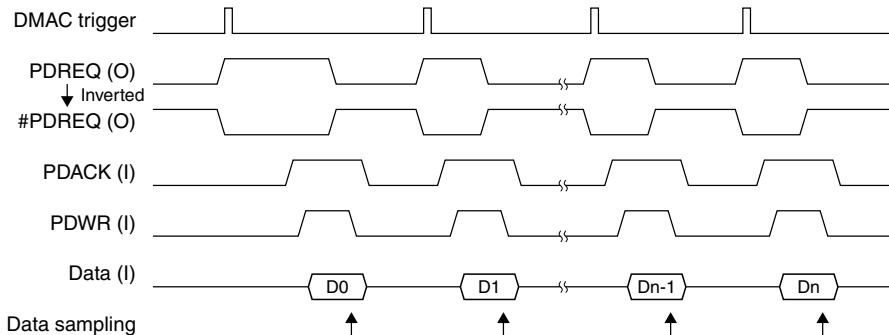


Figure 28.5.3.4 Transfer Waveforms in Asynchronous Single-Word DMA Transfer Mode - Writing

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, the USB macro requests data transfer by asserting PDREQ if any data exist at the connected endpoint. Turning PDACK to active starts outputting transferred data to the data bus. Have the DMAC (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. 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 0x0, 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 0x00000000. To cancel (negate) the DMA request (PDREQ), provide 1 to the DMA_Control.DMA_Stop bit. Note that writing 1 to the DMA_Control.DMA_Stop bit does not stop the DMAC. So to terminate data transfer, first terminate the DMAC (master) and then terminate the macro's DMA transfer.

Note: The S1C33L26 DMAC can only be triggered to start data transfer by the Rising Edge of PDREQ. The subsequent DMAC trigger will be issued at the next PDREQ Rising Edge. When the DMAC transfer counter reaches 0, DMA transfer will not be started even if a DMAC trigger is issued. Therefore, when using the USB macro in single-word DMA transfer mode, configure the DMAC in single transfer mode and set the DMAC transfer counter to a value equal to or less than that set in the DMA_Remain_H and DMA_Remain_L registers.

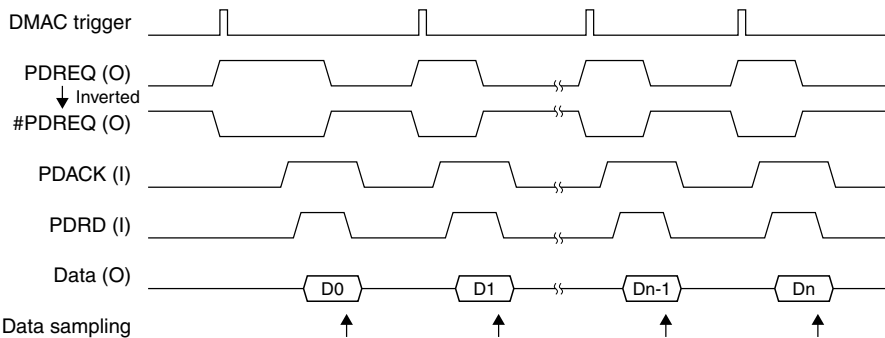


Figure 28.5.3.5 Transfer Waveforms in Asynchronous Single-Word DMA Transfer Mode - Reading

28.5.4 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/MISC_USB register, 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.

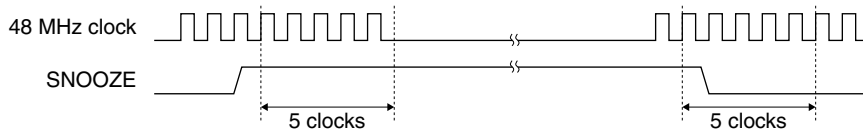


Figure 28.5.4.1 Snooze Sequence

Snooze mode should be set or canceled by the following procedure:

Setting snooze mode

- (1) Write 0x96 to the MISC_PROTECT register to disable write protection for the Misc registers.
- (2) Set USBSNZ in the MISC_USB register to 1 to enable the snooze control.
- (3) Write a value other than 0x96 to the MISC_PROTECT register to enable write protection for the Misc registers.
- (4) Write 0x96 to the CMU_PROTECT register to disable write protection for the CMU registers.
- (5) Set USBCLK_EN in the CMU_CLKCTL register to 0 to stop supplying the USB clock.
- (6) Write a value other than 0x96 to the CMU_PROTECT register to enable write protection for the CMU registers.

Canceling snooze mode

- (1) Write 0x96 to the CMU_PROTECT register to disable write protection for the CMU registers.
- (2) Set USBCLK_EN in the CMU_CLKCTL register to 1 to start supplying the USB clock.
- (3) Write a value other than 0x96 to the CMU_PROTECT register to enable write protection for the CMU registers.
- (4) Write 0x96 to the MISC_PROTECT register to disable write protection for the Misc registers.
- (5) Set USBSNZ in the MISC_USB register to disable the snooze control.
- (6) Write a value other than 0x96 to the MISC_PROTECT register to enable write protection for the Misc registers.

28.6 Registers

28.6.1 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
0x300c00	MainIntStat	R/W	0x00	<i>SIE_IntStat</i>	EPIntStat	DMA_IntStat	FIFO_IntStat	–	–	EP0IntStat	RcvEP0SETUP
0x300c01	SIE_IntStat	R/W	0x00	<i>VBUS_Changed</i>	<i>NonJ</i>	DetectReset	DetectSuspend	RcvSOF	DetectJ	–	SetAddressCmp
0x300c02	EPIntStat	R	0x00	–	–	–	–	EP0IntStat	EPIntStat	EPIntStat	EPIntStat
0x300c03	DMA_IntStat	R/W	0x00	–	–	–	–	–	–	DMA_CountUp	DMA_Cmp
0x300c04	FIFO_IntStat	R/W	0x00	DescriptorCmp	–	–	–	–	–	FIFO_IN_Cmp	FIFO_OUT_Cmp
0x300c05											
0x300c06											
0x300c07	EP0IntStat	R/W	0x00	–	–	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x300c08	EPIntStat	R/W	0x00	–	OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x300c09	EPbIntStat	R/W	0x00	–	OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x300c0a	EPcIntStat	R/W	0x00	–	OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x300c0b	EPdIntStat	R/W	0x00	–	OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x300c0c											
0x300c0d											
0x300c0e											
0x300c0f											

28 USB FUNCTION CONTROLLER (USB)

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
0x300c10	MainIntEnb	R/W	0x00	EnSIE_IntStat	EnEPIntStat	EnDMA_IntStat	EnFIFO_IntStat	-	-	EnEPIntStat	EnRcvEPOSETUP
0x300c11	SIE_IntEnb	R/W	0x00	EnVBUS_Changed	EnNonJ	EnDetectReset	EnDetectSuspend	EnRcvSOF	EnDetectJ	-	EnSetAddressCmp
0x300c12	EPIntEnb	R/W	0x00	-	-	-	-	EnEPIntStat	EnEPIntStat	EnEPIntStat	EnEPIntStat
0x300c13	DMA_IntEnb	R/W	0x00	-	-	-	-	-	-	EnDMA_CountUp	EnDMA_Cmp
0x300c14	FIFO_IntEnb	R/W	0x00	EnDescriptorCmp	-	-	-	-	-	EnFIFO_IN_Cmp	EnFIFO_OUT_Cmp
0x300c15											
0x300c16											
0x300c17	EPOIntEnb	R/W	0x00	-	-	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x300c18	EPaIntEnb	R/W	0x00	-	EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x300c19	EPbIntEnb	R/W	0x00	-	EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x300c1a	EPcIntEnb	R/W	0x00	-	EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x300c1b	EPdIntEnb	R/W	0x00	-	EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x300c1c											
0x300c1d											
0x300c1e											
0x300c1f											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
0x300c20	RevisionNum	R	0x12	Revision Num[7:0]									
0x300c21	USB_Control	R/W	0x00	DisBusDetect	EnAutoNego	InSUSPEND	StartDetectJ	SendWakeup	-	-	ActiveUSB		
0x300c22	USB_Status	R	0xXX	VBUS	1(FS)	-	-	-	-	-	LineState[1:0]		
0x300c23	XcvrControl	R/W	0x01	RpuEnb	-	-	-	-	-	-	OpMode[1:0]		
0x300c24	USB_Test	R/W	0x00	EnUSB_Test	-	-	-	Test_SE0_NAK	Test_J	Test_K	Test_Packet		
0x300c25	EPnControl	W	0x00	AllForceNAK	EPnForceSTALL	AllFIFO_Clr	-	-	-	-	EPnFIFO_Clr		
0x300c26	EPnFIFO_Clr	W	0x00	-	-	-	-	EPnFIFO_Clr	EPcFIFO_Clr	EPbFIFO_Clr	EPaFIFO_Clr		
0x300c27													
0x300c28													
0x300c29													
0x300c2a													
0x300c2b													
0x300c2c													
0x300c2d													
0x300c2e	FrameNumber_H	R	0x80	FnInvalid	-	-	-	-	-	-	FrameNumber[10:8]		
0x300c2f	FrameNumber_L	R	0x00	FrameNumber[7:0]									

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0	
0x300c30	EPOSetup_0	R	0x00	EPOSetup_0[7:0]								
0x300c31	EPOSetup_1	R	0x00	EPOSetup_1[7:0]								
0x300c32	EPOSetup_2	R	0x00	EPOSetup_2[7:0]								
0x300c33	EPOSetup_3	R	0x00	EPOSetup_3[7:0]								
0x300c34	EPOSetup_4	R	0x00	EPOSetup_4[7:0]								
0x300c35	EPOSetup_5	R	0x00	EPOSetup_5[7:0]								
0x300c36	EPOSetup_6	R	0x00	EPOSetup_6[7:0]								
0x300c37	EPOSetup_7	R	0x00	EPOSetup_7[7:0]								
0x300c38	USB_Address	R/W	0x00	AutoSetAddress	USB_Address[6:0]							
0x300c39	EPnControl	R/W	0x00	InXOUT	-	-	-	-	-	-	ReplyDescriptor	
0x300c3a	EPnControlIN	R/W	0x00	-	EnShortPkt	-	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL	
0x300c3b	EPnControlOUT	R/W	0x00	AutoForceNAK	-	-	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL	
0x300c3c												
0x300c3d												
0x300c3e												
0x300c3f	EPnMaxSize	R/W	0x08	-	EPnMaxSize[6:3]				-	-	-	

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0
0x300c40	EPaControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x300c41	EPbControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x300c42	EPcControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x300c43	EPdControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x300c44											
0x300c45											
0x300c46											
0x300c47											
0x300c48											
0x300c49											
0x300c4a											
0x300c4b											
0x300c4c											
0x300c4d											
0x300c4e											
0x300c4f											

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
0x300c50	EPaMaxSize_H	R/W	0x00	-	-	-	-	-	-	EPaMaxSize[9:8]			
0x300c51	EPaMaxSize_L	R/W	0x00	EPaMaxSize[7:0]									
0x300c52	EPaConfig_0	R/W	0x00	INxOUT	ToggleMode	EnEndPoint	-	-	EndPointNumber[3:0]				
0x300c53	EPaConfig_1	R/W	0x00	ISO	ISO_CRCmode	-	-	-	-	-	-		
0x300c54	EPbMaxSize_H	R/W	0x00	-	-	-	-	-	-	EPbMaxSize[9:8]			
0x300c55	EPbMaxSize_L	R/W	0x00	EPbMaxSize[7:0]									
0x300c56	EPbConfig_0	R/W	0x00	INxOUT	ToggleMode	EnEndPoint	-	-	EndPointNumber[3:0]				
0x300c57	EPbConfig_1	R/W	0x00	ISO	ISO_CRCmode	-	-	-	-	-	-		
0x300c58	EPcMaxSize_H	R/W	0x00	-	-	-	-	-	-	EPcMaxSize[9:8]			
0x300c59	EPcMaxSize_L	R/W	0x00	EPcMaxSize[7:0]									
0x300c5a	EPcConfig_0	R/W	0x00	INxOUT	ToggleMode	EnEndPoint	-	-	EndPointNumber[3:0]				
0x300c5b	EPcConfig_1	R/W	0x00	ISO	ISO_CRCmode	-	-	-	-	-	-		
0x300c5c	EPdMaxSize_H	R/W	0x00	-	-	-	-	-	-	EPdMaxSize[9:8]			
0x300c5d	EPdMaxSize_L	R/W	0x00	EPdMaxSize[7:0]									
0x300c5e	EPdConfig_0	R/W	0x00	INxOUT	ToggleMode	EnEndPoint	-	-	EndPointNumber[3:0]				
0x300c5f	EPdConfig_1	R/W	0x00	ISO	ISO_CRCmode	-	-	-	-	-	-		

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
0x300c70	EPaStartAdrs_H	R/W	0x00	-	-	-	-	-	EPaStartAdrs[11:8]				
0x300c71	EPaStartAdrs_L	R/W	0x00	EPaStartAdrs[7:2]									
0x300c72	EPbStartAdrs_H	R/W	0x00	-	-	-	-	-	EPbStartAdrs[11:8]				
0x300c73	EPbStartAdrs_L	R/W	0x00	EPbStartAdrs[7:2]									
0x300c74	EPcStartAdrs_H	R/W	0x00	-	-	-	-	-	EPcStartAdrs[11:8]				
0x300c75	EPcStartAdrs_L	R/W	0x00	EPcStartAdrs[7:2]									
0x300c76	EPdStartAdrs_H	R/W	0x00	-	-	-	-	-	EPdStartAdrs[11:8]				
0x300c77	EPdStartAdrs_L	R/W	0x00	EPdStartAdrs[7:2]									
0x300c78													
0x300c79													
0x300c7a													
0x300c7b													
0x300c7c													
0x300c7d													
0x300c7e													
0x300c7f													

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
0x300c80	CPU_JoinRd	R/W	0x00	-	-	-	-	JoinEPdRd	JoinEPcRd	JoinEPbRd	JoinEPaRd		
0x300c81	CPU_JoinWr	R/W	0x00	-	-	-	-	JoinEPdWr	JoinEPcWr	JoinEPbWr	JoinEPaWr		
0x300c82	EnEPnFIFO_Access	R/W	0x00	-	-	-	-	-	-	EnEPnFIFO_Wr	EnEPnFIFO_Rd		
0x300c83	EPnFIFOforCPU	R/W	0xXX	EPnFIFOData[7:0]									
0x300c84	EPnRdRemain_H	R	0x00	-	-	-	-	-	EPnRdRemain[11:8]				
0x300c85	EPnRdRemain_L	R	0x00	EPnRdRemain[7:0]									
0x300c86	EPnWrRemain_H	R	0x00	-	-	-	-	-	EPnWrRemain[11:8]				
0x300c87	EPnWrRemain_L	R	0x00	EPnWrRemain[7:0]									
0x300c88	DescAdrs_H	R/W	0x00	-	-	-	-	-	DescAdrs[11:8]				
0x300c89	DescAdrs_L	R/W	0x00	DescAdrs[7:0]									
0x300c8a	DescSize_H	R/W	0x00	-	-	-	-	-	-	DescSize[9:8]			
0x300c8b	DescSize_L	R/W	0x00	DescSize[7:0]									
0x300c8c													
0x300c8d													
0x300c8e													
0x300c8f	DescDoor	R/W	0x00	DescMode[7:0]									

Address	Register name	R/W	Init	D7	D6	D5	D4	D3	D2	D1	D0		
0x300c90	DMA_FIFO_Control	R/W	0x00	FIFO_Running	AutoEnShort	-	-	-	-	-	-		
0x300c91	DMA_Join	R/W	0x00	-	-	-	-	JoinEPdDMA	JoinEPcDMA	JoinEPbDMA	JoinEPaDMA		
0x300c92	DMA_Control	R/W	0x00	DMA_Running	PDREQ	PDACK	-	CounterClr	-	DMA_Stop	DMA_Go		
0x300c93													
0x300c94	DMA_Config_0	R/W	0x00	ActivePort	-	-	-	PDREQ_Level	PDACK_Level	PDRDWR_Level	-		
0x300c95	DMA_Config_1	R/W	0x00	RcvLimitMode	-	-	-	SingleWord	-	-	CountMode		
0x300c96													
0x300c97	DMA_Latency	R/W	0x00	-	-	-	-	-	DMA_Latency[3:0]				
0x300c98	DMA_Remain_H	R	0x00	-	-	-	-	-	DMA_Remain[11:8]				
0x300c99	DMA_Remain_L	R	0x00	DMA_Remain[7:0]									
0x300c9a													
0x300c9b													
0x300c9c	DMA_Count_HH	R/W	0x00	DMA_Count[31:24]									
0x300c9d	DMA_Count_HL	R/W	0x00	DMA_Count[23:16]									
0x300c9e	DMA_Count_LH	R/W	0x00	DMA_Count[15:8]									
0x300c9f	DMA_Count_LL	R/W	0x00	DMA_Count[7:0]									

28.6.2 Detailed Description of Registers

MainIntStat (Main Interrupt Status)

Register name	Address	Bit	Name		Setting	Init.	R/W	Remarks		
MainIntStat (Main interrupt status)	0x300c00 (8 bits)	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.

SIE_IntStat (SIE Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
SIE_IntStat (SIE interrupt status)	0x300c01 (8 bits)	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 In-SUSPEND bit of the USB_Control register is set to 1. This bit is valid during snooze as well.

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 MISC_USB register 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.

EPIntStat (EP Interrupt Status)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks		
EPIntStat (EP interrupt status)	0x300c02 (8 bits)	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.

DMA_IntStat (DMA Interrupt Status)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks		
DMA_IntStat (DMA interrupt status)	0x300c03 (8 bits)	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.

FIFO_IntStat (FIFO Interrupt Status)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks		
FIFO_IntStat (FIFO interrupt status)	0x300c04 (8 bits)	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_TransNAK 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.

EP0IntStat (EP0 Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EP0IntStat (EP0 interrupt status)	0x300c07 (8 bits)	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)

This register displays the interrupt status of the endpoint EP0.

D[7:6] Reserved**D5 IN_TransACK**

Shows a cause of interrupt directly.

Set to 1 when ACK is received in the IN transaction.

D4 OUT_TransACK

Shows a cause of interrupt directly.

Set to 1 when ACK is replied in the OUT transaction.

D3 IN_TransNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the IN transaction.

D2 OUT_TransNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the OUT transaction.

D1 IN_TransErr

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_TransErr

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.

EPaintStat (EPa Interrupt Status)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPaintStat (EPa interrupt status)	0x300c08 (8 bits)	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.

EPbIntStat (EPb Interrupt Status)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPbIntStat (EPb interrupt status)	0x300c09 (8 bits)	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_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.

EPcIntStat (EPc Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPcIntStat (EPc interrupt status)	0x300c0a (8 bits)	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.

28 USB FUNCTION CONTROLLER (USB)

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_TrnErr

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.

EPdIntStat (EPd Interrupt Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPdIntStat (EPd interrupt status)	0x300c0b (8 bits)	D7	–		–	–	–	0 when being read.	
		D6	OUT_ShortACK	1	Out short packet ACK	0	None	0	R(W)
		D5	IN_TrnACK	1	In transaction ACK	0	None	0	R(W)
		D4	OUT_TrnACK	1	Out transaction ACK	0	None	0	R(W)
		D3	IN_TrnNAK	1	In transaction NAK	0	None	0	R(W)
		D2	OUT_TrnNAK	1	Out transaction NAK	0	None	0	R(W)
		D1	IN_TrnErr	1	In transaction error	0	None	0	R(W)
		D0	OUT_TrnErr	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_TrnACK and this bits are set to 1 at the same time.

D5 IN_TrnACK

Shows a cause of interrupt directly.

Set to 1 when ACK is received in the IN transaction.

D4 OUT_TrnACK

Shows a cause of interrupt directly.

Set to 1 when ACK is replied in the OUT transaction.

D3 IN_TrnNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the IN transaction.

D2 OUT_TrnNAK

Shows a cause of interrupt directly.

Set to 1 when NAK is replied in the OUT transaction.

D1 IN_TrnErr

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_TrnErr

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.

MainIntEnb (Main Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
MainIntEnb (Main interrupt enable)	0x300c10 (8 bits)	D7	EnSIE_IntStat	1	Enable	0	Disable			
		D6	EnEPrintStat							
		D5	EnDMA_IntStat							
		D4	EnFIFO_IntStat							
		D3–2	–		–	–	0 when being read.			
		D1	EnEP0IntStat	1	Enable	0	Disable		0	R/W
		D0	EnRcvEP0SETUP							

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.

SIE_IntEnb (SIE Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
SIE_IntEnb (SIE interrupt enable)	0x300c11 (8 bits)	D7	EnVBUS_Changed	1	Enable	0	Disable	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	–					–	–	
		D0	EnSetAddressCmp	1	Enable	0	Disable	0	R/W	

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.

EPrIntEnb (EPr Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EPrIntEnb (EPr interrupt enable)	0x300c12 (8 bits)	D7–4	–	1	Enable	0	Disable	–	–	0 when being read.
		D3	EnEPdIntStat					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.

DMA_IntEnb (DMA Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
DMA_IntEnb (DMA interrupt enable)	0x300c13 (8 bits)	D7–2	–	1	Enable	0	Disable	–	–	0 when being read.
		D1	EnDMA_CountUp					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.

FIFO_IntEnb (FIFO Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
FIFO_IntEnb (FIFO interrupt enable)	0x300c14 (8 bits)	D7	EnDescriptorCmp	1	Enable	0	Disable	0	R/W	
		D6–2	–					–	–	
		D1	EnFIFO_IN_Cmp	1	Enable	0	Disable	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.

EP0IntEnb (EP0 Interrupt Enable)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EP0IntEnb (EP0 interrupt enable)	0x300c17 (8 bits)	D7–6	–	1	Enable	0	Disable	–	–	0 when being read.
		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 EP0IntStat bit of the MainIntStat register with the cause of interrupt of the EP0IntStat register.

EPaIntEnb (EPa Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPaIntEnb (EPa interrupt enable)	0x300c18 (8 bits)	D7	–	1 Enable	0 Disable	–	0 when being read.
		D6	EnOUT_ShortACK			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.

EPbIntEnb (EPb Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPbIntEnb (EPb interrupt enable)	0x300c19 (8 bits)	D7	–	1 Enable	0 Disable	–	0 when being read.
		D6	EnOUT_ShortACK			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.

EPcIntEnb (EPc Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPcIntEnb (EPc interrupt enable)	0x300c1a (8 bits)	D7	–	1 Enable	0 Disable	–	0 when being read.
		D6	EnOUT_ShortACK			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.

EPdIntEnb (EPd Interrupt Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPdIntEnb (EPd interrupt enable)	0x300c1b (8 bits)	D7	–	1 Enable	0 Disable	–	0 when being read.
		D6	EnOUT_ShortACK			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.

RevisionNum (Revision Number)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
RevisionNum (Revision number)	0x300c20 (8 bits)	D7–0	RevisionNum[7:0]	Revision number (0x12)	0x12	R	

This register shows the revision number of the USB function controller. This register is valid during snooze as well.

USB_Control (USB Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
USB_Control (USB control)	0x300c21 (8 bits)	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	Deactivate 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 μ s 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.

USB_Status (USB Status)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
USB_Status (USB status)	0x300c22 (8 bits)	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.
		D1-0	LineState[1:0]	LineState[1:0]		DP/DM		X	R	
				0x3	SE1	0x2	K	0x1	J	0x0

This register displays the status related to the USB.

This register is valid during snooze as well.

28 USB FUNCTION CONTROLLER (USB)

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[1:0]	DP/DM
0x3	SE1
0x2	K
0x1	J
0x0	SE0

XcvrControl (Xcvr Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
XcvrControl (Xcvr control)	0x300c23 (8 bits)	D7	RpuEnb	1	Enable pull-up	0	Disable pull-up	0	R/W	
		D6-2	-	-	-	-	-	-	-	0 when being read.
		D1-0	OpMode[1:0]	OpMode[1:0]	Operation mode		0x1	R/W		
				0x3	reserved					
0x2	Disable bitstuffing and NRZI encoding									
0x1	Non-driving									
0x0	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[1:0]	Operation mode
0x3	Reserved
0x2	Disable bitstuffing and NRZI encoding
0x1	Non-driving
0x0	Normal operation

* When the USB cable is pulled off, it is recommended to set this register to 0x1.

USB_Test (USB Test)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
USB_Test (USB test)	0x300c24 (8 bits)	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 0x2 (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 0x2 (Disable Bitstuffing and NRZI encoding).

D0 Test_Packet

By setting this bit to 1, the Test_Packet test mode can start.

Since this test mode uses the endpoint EPc, set the followings.

- (1) Set the MaxPacketSize of the endpoint EPc to 64 or more, the direction of transfer to IN and the EndPointNumber to 0xf to make the endpoint be ready to use. And allocate the FIFO of the endpoint EPc for 64 bytes or more.
- (2) Do not overlap the above setting with the settings of the endpoints EPa and EPb.
Or clear the EPaConfig_0.EnEndPoint bit and EPbConfig_0.EnEndPoint bit.
- (3) Clear the FIFO of the EPc and write data for the following test packet into this FIFO.
- (4) Set the EnIN_TranErr of the EPcIntEnb register to 0 (clear this bit).
IN_TranErr status is set to 1 at every time the Test Packet transmission completes.

The data to write into the FIFO in the packet transmission test mode are the following 53 bytes.

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xaa, 0xaa, 0xaa, 0xaa, 0xaa, 0xaa, 0xaa,
0xaa, 0xee, 0xee, 0xee, 0xee, 0xee, 0xee, 0xee,
0xee, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0xbf, 0xdf,
0xef, 0xf7, 0xfb, 0xfd, 0xfc, 0x7e, 0xbf, 0xdf,
0xef, 0xf7, 0xfb, 0xfd, 0x7e
```

Since the SIE adds the PID and CRC to the test packet when sending it, the data to write into the FIFO are from “the data after the DATA 0 PID” to “the data before the CRC16” that are described as the test packet data in the USB standard Rev.2.0. (Note that Test Packet is defined only HS mode in USB specification.)

EPnControl (Endpoint Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks
EPnControl (Endpoint control)	0x300c25 (8 bits)	D7	AllForceNAK	1	Set all ForceNAK	0	W	0 when being read.
		D6	EPrForceSTALL	1	Set EP's ForceSTALL	0	W	
		D5	AllFIFO_Clr	1	Clear all FIFO	0	W	
		D4-1	-	-	-	-	-	
		D0	EP0FIFO_Clr	1	Clear EP0 FIFO	0	W	

This register sets operations of entire endpoints, and display them.

D7 AllForceNAK

Sets the ForceNAK bit of all endpoints to 1.

28 USB FUNCTION CONTROLLER (USB)

D6 EPrForceSTALL

Sets the ForceSTALL bit of EPa, EPb, EPc and EPd endpoints to 1.

D5 AIIFIFO_Clr

Clears 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

Clears the FIFO of the endpoint EP0. This bit is automatically set 0 (to be cleared) after completing the FIFO clear operation.

EPrFIFO_Clr (EPr FIFO Clear)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPrFIFO_Clr (EPr FIFO clear)	0x300c26 (8 bits)	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

Clears 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

Clears 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

Clears 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

Clears 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.

FrameNumber_H (Frame Number HIGH)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
FrameNumber_H (Frame number high)	0x300c2e (8 bits)	D7	FnInvalid	1 Invalid frame number 0 Valid frame number	1	R	
		D6-3	–	–	–	–	0 when being read.
		D2-0	FrameNumber[10:8]	Frame number high	0x0	R	

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.

FrameNumber_L (Frame Number LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
FrameNumber_L (Frame number low)	0x300c2f (8 bits)	D7-0	FrameNumber[7:0]	Frame number low	0x0	R	

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.

EP0Setup_0 (EP0 Setup 0)–EP0Setup_7 (EP0 Setup 7)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EP0Setup_0 (EP0 setup 0)	0x300c30	D7-0	EP0Setup_n[7:0]	Endpoint 0 setup data 0	0x0	R	
EP0Setup_7 (EP0 setup 7)	0x300c37 (8 bits)			Endpoint 0 setup data 7			

Eight-byte data received at the endpoint EP0 setup stage are stored from the EP0Setup_0 sequentially.

EP0Setup_0 register

BmRequestType is set.

EP0Setup_1 register

BRequest is set.

EP0Setup_2 register

The lower order 8 bits in Wvalue are set.

EP0Setup_3 register

The upper order 8 bits in Wvalue are set.

EP0Setup_4 register

The lower order 8 bits in WIndex are set.

EP0Setup_5 register

The upper order 8 bits in WIndex are set.

EP0Setup_6 register

The lower order 8 bits in WLength are set.

EP0Setup_7 register

The upper order 8 bits in WLength are set.

USB_Address (USB Address)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
USB_Address (USB address)	0x300c38 (8 bits)	D7	AutoSetAddress	1 Auto set address	0 Do nothing	0	R/W	
		D6-0	USB_Address[6:0]	USB address		0x0	R/W	

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 EP0Setup_0-7 registers and interpret the request.

- (2) Set the AutoSetAddress bit.
- (3) Set the INxOUT bit of the EP0Control register.
- (4) Clear the ForceNAK bit of the EP0ControlIN register, and set the EnShortPkt bit.
- (5) Wait for the end of the status stage.

The SetAddressCmp bit of the SIE_IntStat register is set to 1.

D[6:0] USB_Address[6:0]

These bits set up the USB address.

The USB address is written automatically by the AutoSetAddress function. Or it can be written.

EP0Control (EP0 Control)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EP0Control (EP0 control)	0x300c39 (8 bits)	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

Sets 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_TranACK bit of the EP0IntStat register is set to 1.

Refer to the section describing operations, for details.

EP0ControlIN (EP0 Control IN)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks		
EP0ControlIN (EP0 control IN)	0x300c3a (8 bits)	D7	–		–	–	0 when being read.		
		D6	EnShortPkt	1 Enable short packet	0 Do nothing	0	R/W		
		D5	–		–	–	–	0 when being read.	
		D4	ToggleStat		Toggle sequence bit		0	R	
		D3	ToggleSet	1 Set toggle sequence bit	0 Do nothing	0	R/W		
		D2	ToggleClr	1 Clear toggle sequence bit	0 Do nothing	0	R/W		
		D1	ForceNAK	1 Force NAK	0 Do nothing	0	R/W		
		D0	ForceSTALL	1 Force STALL	0 Do nothing	0	R/W		

This register sets the operations related to the IN transaction of the endpoint EP0 and displays their status.

D7 **Reserved**

D6 **EnShortPkt**

Setting this bit to 1 enables to send the data within the FIFO that is less than the quantity specified for the MaxPacketSize, as a short packet for the IN transaction of the endpoint EP0. When the IN transaction that transmitted short packets completes, this bit is automatically set to 0 (to be cleared). When a packet of the max packet size is transmitted, this bit is not cleared.

If this bit is set to 1 when the FIFO has no data, a zero-length packet can be transmitted for the IN token from the host. If the data is written into the FIFO that is in the transmission process with the packet to which this bit is set, that data may be included in transmission. Therefore, do not write into the FIFO until the packet transmission completes and this bit is cleared.

D5 **Reserved**

D4 **ToggleStat**

Shows the status of the toggle sequence bit in the IN transaction of the endpoint EP0.

D3 **ToggleSet**

Sets the toggle sequence bit in the IN transaction of the endpoint EP0, to 1.

D2 **ToggleClr**

Sets the toggle sequence bit in the IN transaction of the endpoint EP0, to 0 (clear).

D1 **ForceNAK**

If this bit is set to 1, the NAK response is done for the IN transaction of the endpoint EP0, regardless of the FIFO data quantity.

When the RcvEP0SETUP bit of the MainIntStat register is set to 1 after completion of the setup stage, this bit is set to 1, and this bit cannot be set to 0 (to be cleared) as long as the RcvEP0SETUP bit is 1. When the IN transaction that transmitted short packets completes, this bit is set to 1.

When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

D0 **ForceSTALL**

If this bit is set to 1, the STALL response is done for the IN transaction of the endpoint EP0. This bit has a priority over the setting of the ForceNAK bit.

When the RcvEP0SETUP bit of the MainIntStat register is set to 1 after completion of the setup stage, this bit is set to 0 (to be cleared), and this bit cannot be set to 1 as long as the RcvEP0SETUP bit is 1.

When a transaction has been being done for a certain period of time, the setting of this bit will be enabled from the next transaction.

EP0ControlOUT (EP0 Control OUT)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EP0ControlOUT (EP0 control OUT)	0x300c3b (8 bits)	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 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.

EP0MaxSize (EP0 Max Packet Size)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks
EP0MaxSize (EP0 max packet size)	0x300c3f (8 bits)	D7	-	-	-	-	-	0 when being read.
		D6-3	EP0MaxSize[6:3]	Endpoint EP0 max packet size		0x1	R/W	
		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

EPaControl (EPa Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks			
EPaControl (EPa control)	0x300c40 (8 bits)	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**

Sets the toggle sequence bit of the endpoint EPa to 1.

D2 **ToggleClr**

Sets 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.

EPbControl (EPb Control)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks		
EPbControl (EPb control)	0x300c41 (8 bits)	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

Sets the toggle sequence bit of the endpoint EPb to 1.

D2 ToggleClr

Sets 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.

EPcControl (EPc Control)

Register name	Address	Bit	Name	Setting			Init.	R/W	Remarks		
EPcControl (EPc control)	0x300c42 (8 bits)	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

Sets the toggle sequence bit of the endpoint EPc to 1.

D2 ToggleClr

Sets 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.

EPdControl (EPd Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
EPdControl (EPd control)	0x300c43 (8 bits)	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	0 when being read.
		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	
		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

Sets the toggle sequence bit of the endpoint EPd to 1.

D2 ToggleClr

Sets 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.

EPaMaxSize_H (EPa Max Packet Size HIGH)**EPaMaxSize_L (EPa Max Packet Size LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPaMaxSize_H (EPa max packet size high)	0x300c50 (8 bits)	D7-2	–	–	–	–	0 when being read.
		D1-0	EPaMaxSize[9:8]	Endpoint EPa max packet size	0x0	R/W	
EPaMaxSize_L (EPa max packet size low)	0x300c51 (8 bits)	D7-0	EPaMaxSize[7:0]	Endpoint EPa max packet size	0x0	R/W	

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.

EPaConfig_0 (EPa Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPaConfig_0 (EPa configuration 0)	0x300c52 (8 bits)	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	–	–	–	–	–	
		D3–0	EndPointNumber [3:0]	Endpoint number (0x1 to 0xf)	0x0	R/W		

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

Sets the transfer direction of the endpoint.

D6 ToggleMode

Sets 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[3:0]

Sets an endpoint number between 0x1 and 0xf.

EPaConfig_1 (EPa Configuration 1)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPaConfig_1 (EPa configuration 1)	0x300c53 (8 bits)	D7	ISO	1 ISO	0 Non-ISO	0	R/W	
		D6	ISO_CRCmode	1 CRC mode	0 Normal ISO	0	R/W	
		D5–0	–	–	–	–	–	0 when being read.

This register sets up the endpoint EPa.

Perform the setup so that combination of the EndPointNumber and the INxOUT does not overlap with those of other endpoints.

D7 ISO

Sets 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

EPbMaxSize_H (EPb Max Packet Size HIGH)

EPbMaxSize_L (EPb Max Packet Size LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPbMaxSize_H (EPb max packet size high)	0x300c54 (8 bits)	D7–2	–	–	–	–	0 when being read.
		D1–0	EPbMaxSize[9:8]	Endpoint EPb max packet size	0x0	R/W	
EPbMaxSize_L (EPb max packet size low)	0x300c55 (8 bits)	D7–0	EPbMaxSize[7:0]	Endpoint EPb max packet size	0x0	R/W	

28 USB FUNCTION CONTROLLER (USB)

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.

EPbConfig_0 (EPb Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPbConfig_0 (EPb configuration 0)	0x300c56 (8 bits)	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.
		D3–0	EndPointNumber [3:0]	Endpoint number (0x1 to 0xf)	0x0	R/W		

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

Sets the transfer direction of the endpoint.

D6 ToggleMode

Sets 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[3:0]

Sets an endpoint number between 0x1 and 0xf.

EPbConfig_1 (EPb Configuration 1)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPbConfig_1 (EPb configuration 1)	0x300c57 (8 bits)	D7	ISO	1 ISO	0 Non-ISO	0	R/W	
		D6	ISO_CRCmode	1 CRC mode	0 Normal ISO	0	R/W	
		D5–0	–	–	–	–	–	0 when being read.

This register sets up the endpoint EPb.

Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 ISO

Sets 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

EPcMaxSize_H (EPc Max Packet Size HIGH)

EPcMaxSize_L (EPc Max Packet Size LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPcMaxSize_H (EPc max packet size high)	0x300c58 (8 bits)	D7-2	–	–	–	–	–	0 when being read.
		D1-0	EPcMaxSize[9:8]	Endpoint EPc max packet size	0x0	R/W		
EPcMaxSize_L (EPc max packet size low)	0x300c59 (8 bits)	D7-0	EPcMaxSize[7:0]	Endpoint EPc max packet size	0x0	R/W		

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.

EPcConfig_0 (EPc Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks		
EPcConfig_0 (EPc configuration 0)	0x300c5a (8 bits)	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.
		D3-0	EndPointNumber [3:0]	Endpoint number (0x1 to 0xf)	0x0	R/W			

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

Sets the transfer direction of the endpoint.

D6 ToggleMode

Sets 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[3:0]

Sets an endpoint number between 0x1 and 0xf.

EPcConfig_1 (EPc Configuration 1)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPcConfig_1 (EPc configuration 1)	0x300c5b (8 bits)	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

Sets 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

EPdMaxSize_H (EPd Max Packet Size HIGH) EPdMaxSize_L (EPd Max Packet Size LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPdMaxSize_H (EPd max packet size high)	0x300c5c (8 bits)	D7-2	–	–	–	–	0 when being read.
		D1-0	EPdMaxSize[9:8]	Endpoint EPd max packet size	0x0	R/W	
EPdMaxSize_L (EPd max packet size low)	0x300c5d (8 bits)	D7-0	EPdMaxSize[7:0]	Endpoint EPd max packet size	0x0	R/W	

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.

EPdConfig_0 (EPd Configuration 0)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPdConfig_0 (EPd configuration 0)	0x300c5e (8 bits)	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.
		D3-0	EndPointNumber [3:0]	Endpoint number (0x1 to 0xf)	0x0	R/W		

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

Sets the transfer direction of the endpoint.

D6 ToggleMode

Sets 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[3:0]**

Sets an endpoint number between 0x1 and 0xf.

EPdConfig_1 (EPd Configuration 1)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPdConfig_1 (EPd configuration 1)	0x300c5f (8 bits)	D7	ISO	1 ISO	0 Non-ISO	0	R/W	
		D6	ISO_CRCmode	1 CRC mode	0 Normal ISO	0	R/W	
		D5-0	–	–	–	–	–	0 when being read.

This register sets up the endpoint EPd.

Perform the setup so that combination of the EndpointNumber and the INxOUT does not overlap with those of other endpoints.

D7 ISO

Sets 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**EPaStartAdrs_H (EPa FIFO Start Address HIGH)****EPaStartAdrs_L (EPa FIFO Start Address LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPaStartAdrs_H (EPa FIFO start address high)	0x300c70 (8 bits)	D7-4	-	-	-	-	0 when being read.
		D3-0	EPaStartAdrs[11:8]	Endpoint EPa start address	0x0	R/W	
EPaStartAdrs_L (EPa FIFO start address low)	0x300c71 (8 bits)	D7-2	EPaStartAdrs[7:2]	Endpoint EPa start address	0x0	R/W	
		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.

EPbStartAdrs_H (EPb FIFO Start Address HIGH)**EPbStartAdrs_L (EPb FIFO Start Address LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPbStartAdrs_H (EPb FIFO start address high)	0x300c72 (8 bits)	D7-4	-	-	-	-	0 when being read.
		D3-0	EPbStartAdrs[11:8]	Endpoint EPb start address	0x0	R/W	
EPbStartAdrs_L (EPb FIFO start address low)	0x300c73 (8 bits)	D7-2	EPbStartAdrs[7:2]	Endpoint EPb start address	0x0	R/W	
		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.

EPcStartAdrs_H (EPc FIFO Start Address HIGH)

EPcStartAdrs_L (EPc FIFO Start Address LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPcStartAdrs_H (EPc FIFO start address high)	0x300c74 (8 bits)	D7-4	-	-	-	-	0 when being read.
		D3-0	EPcStartAdrs[11:8]	Endpoint EPc start address	0x0	R/W	
EPcStartAdrs_L (EPc FIFO start address low)	0x300c75 (8 bits)	D7-2	EPcStartAdrs[7:2]	Endpoint EPc start address	0x0	R/W	
		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.

EPdStartAdrs_H (EPd FIFO Start Address HIGH)

EPdStartAdrs_L (EPd FIFO Start Address LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPdStartAdrs_H (EPd FIFO start address high)	0x300c76 (8 bits)	D7-4	-	-	-	-	0 when being read.
		D3-0	EPdStartAdrs[11:8]	Endpoint EPd start address	0x0	R/W	
EPdStartAdrs_L (EPd FIFO start address low)	0x300c77 (8 bits)	D7-2	EPdStartAdrs[7:2]	Endpoint EPd start address	0x0	R/W	
		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.

CPU_JoinRd (CPU Join FIFO Read)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
CPU_JoinRd (CPU join FIFO read)	0x300c80 (8 bits)	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.

CPU_JoinWr (CPU Join FIFO Write)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
CPU_JoinWr (CPU join FIFO write)	0x300c81 (8 bits)	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	

28 USB FUNCTION CONTROLLER (USB)

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.

EnEPnFIFO_Access (EPn FIFO Access Enable)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EnEPnFIFO_Access (Enable EPn FIFO access)	0x300c82 (8 bits)	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.

EPnFIFOforCPU (EPn FIFO for CPU)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnFIFOforCPU (EPn FIFO for CPU)	0x300c83 (8 bits)	D7-0	EPnFIFOData[7:0]	Endpoint n FIFO access from CPU	X	R/W	

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.

EPnRdRemain_H (EPn FIFO Read Remain HIGH) EPnRdRemain_L (EPn FIFO Read Remain LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnRdRemain_H (EPn FIFO read remain high)	0x300c84 (8 bits)	D7-4 D3-0	EPnRdRemain[11:8]	Endpoint n FIFO read remain	- 0x0	- R	0 when being read.
EPnRdRemain_L (EPn FIFO read remain low)	0x300c85 (8 bits)	D7-0	EPnRdRemain[7:0]	Endpoint n FIFO read remain	0x0	R	

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.

EPnWrRemain_H (EPn FIFO Write Remain HIGH) EPnWrRemain_L (EPn FIFO Write Remain LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
EPnWrRemain_H (EPn FIFO write remain high)	0x300c86 (8 bits)	D7-4 D3-0	EPnWrRemain[11:8]	Endpoint n FIFO write remain	- 0x0	- R	0 when being read.
EPnWrRemain_L (EPn FIFO write remain low)	0x300c87 (8 bits)	D7-0	EPnWrRemain[7:0]	Endpoint n FIFO write remain	0x0	R	

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.

DescAdrs_H (Descriptor Address HIGH) DescAdrs_L (Descriptor Address LOW)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescAdrs_H (Descriptor address high)	0x300c88 (8 bits)	D7-4 D3-0	DescAdrs[11:8]	Descriptor address	- 0x0	- R/W	0 when being read.
DescAdrs_L (Descriptor address low)	0x300c89 (8 bits)	D7-0	DescAdrs[7:0]	Descriptor address	0x0	R/W	

DescAdrs[11:0]

Specifies the start address of the FIFO used at the start of Descriptor reply operation, Descriptor write operation and Descriptor read operation in the Descriptor reply function.

The Descriptor Address does not have the function to allocate the FIFO area for the Descriptor reply function. The entire FIFO area ranging from 0x0000 to 0x03ff (1K bytes) can be specified for the Descriptor Address, regardless of the FIFO area setting.

In the Description reply, DescAdrs is updated every time the IN transaction completes at the endpoint EP0, as many times as the number of data transmitted. Refer to the item on the ReplyDescriptor of the 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.

DescSize_H (Descriptor Size HIGH)**DescSize_L (Descriptor Size LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescSize_H (Descriptor size high)	0x300c8a (8 bits)	D7-2	–	Descriptor size	–	–	0 when being read.
		D1-0	DescSize[9:8]				
DescSize_L (Descriptor size low)	0x300c8b (8 bits)	D7-0	DescSize[7:0]	Descriptor size	0x0	R/W	

DescSize[9:0]

Specifies 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.

DescDoor (Descriptor Door)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DescDoor (Descriptor door)	0x300c8f (8 bits)	D7-0	DescMode[7:0]	Descriptor door	0x0	R/W	

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.

DMA_FIFO_Control (DMA FIFO Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
DMA_FIFO_Control (DMA FIFO control)	0x300c90 (8 bits)	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

DMA_Join (DMA Join FIFO)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
DMA_Join (DMA join FIFO)	0x300c91 (8 bits)	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.

DMA_Control (DMA Control)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks		
DMA_Control (DMA control)	0x300c92 (8 bits)	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	

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.

28 USB FUNCTION CONTROLLER (USB)

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 negates the DMA request (PDREQ) signal. After this bit is set to 1, 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.

Note: Setting this bit to 1 does not stop the DMAC. So to terminate data transfer, first terminate the DMAC (master) and then set this bit to 1.

D0 DMA_Go

Setting this bit to 1 starts the DMA transfer.

DMA_Config_0 (DMA Configuration 0)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
DMA_Config_0 (DMA configuration 0)	0x300c94 (8 bits)	D7	ActivePort	1	Activate DMA port	0	Deactivate 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

Sets 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

Sets the PDREQ logic level. Set to 0 (active-high).

D2 PDACK_Level

Sets the PDACK logic level. Set to 0 (active-high).

D1 PDRDWR_Level

Sets the logic levels of the PDRD and PDWR signals. Set to 0 (active-high).

D0 Reserved

DMA_Config_1 (DMA Configuration 1)

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
DMA_Config_1 (DMA configuration 1)	0x300c95 (8 bits)	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.

Note: In the S1C33L26, the USB DMA data transfer count is determined according to the DMAC transfer counter setting. Negating PDREQ by the USB macro does not affect the transfer count. So in RcvLimit mode, the DMAC continues data transfer until the DMAC transfer counter reaches 0 even after the macro negates PDREQ. Therefore, make sure that the DMAC transfer counter is set properly.

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.

- Notes:**
- In multi-word DMA transfer mode, the DMAC can only be triggered to start data transfer by the Rising Edge of PDREQ. After that no DMAC trigger will be issued while PDREQ stays active (high level). The subsequent DMAC trigger will be issued at the next PDREQ Rising Edge. Therefore, when using the USB macro in multi-word DMA transfer mode, configure the DMAC in successive transfer mode and set the DMAC transfer counter to the same value set in the DMA_Remain_H and DMA_Remain_L registers.
 - In single-word DMA transfer mode, the DMAC can only be triggered to start data transfer by the Rising Edge of PDREQ. The subsequent DMAC trigger will be issued at the next PDREQ Rising Edge. When the DMAC transfer counter reaches 0, DMA transfer will not be started even if a DMAC trigger is issued. Therefore, when using the USB macro in single-word DMA transfer mode, configure the DMAC in single transfer mode and set the DMAC transfer counter to a value equal to or less than that set in the DMA_Remain_H and DMA_Remain_L registers.

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 DMAC is stopped. The Transfer Byte Counter (DMA_Count_HH, HL, LH, LL) shows the number of transmissions for reference.

In the Count-down mode, the DMA request (PDREQ) signal is asserted 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.

DMA_Latency (DMA Latency)

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Latency (DMA latency)	0x300c97 (8 bits)	D7-4	–		–	–	0 when being read.
		D3-0	DMA_Latency[3:0]	Latency	0x0	R/W	

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.

DMA_Remain_H (DMA FIFO Remain HIGH)**DMA_Remain_L (DMA FIFO Remain LOW)**

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks
DMA_Remain_H (DMA FIFO remain high)	0x300c98 (8 bits)	D7-4	–		–	–	0 when being read.
		D3-0	DMA_Remain[11:8]	DMA FIFO remain	0x0	R	
DMA_Remain_L (DMA FIFO remain low)	0x300c99 (8 bits)	D7-0	DMA_Remain[7:0]	DMA FIFO remain	0x0	R	

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.

DMA_Count_HH (DMA Transfer Byte Counter HIGH/HIGH)**DMA_Count_HL (DMA Transfer Byte Counter HIGH/LOW)****DMA_Count_LH (DMA Transfer Byte Counter LOW/HIGH)****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)	0x300c9c (8 bits)	D7-0	DMA_Count[31:24]	DMA transfer byte counter	0x0	R/W	
DMA_Count_HL (DMA transfer byte counter high/low)	0x300c9d (8 bits)	D7-0	DMA_Count[23:16]	DMA transfer byte counter	0x0	R/W	
DMA_Count_LH (DMA transfer byte counter low/high)	0x300c9e (8 bits)	D7-0	DMA_Count[15:8]	DMA transfer byte counter	0x0	R/W	
DMA_Count_LL (DMA transfer byte counter low/low)	0x300c9f (8 bits)	D7-0	DMA_Count[7:0]	DMA transfer byte counter	0x0	R/W	

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.

29 Misc Registers (MISC)

- Notes:**
- The Misc registers at addresses 0x300010–0x300018 are write-protected. Before the Misc registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to PROT[7:0]/MISC_PROTECT register. Note that since unnecessary rewrites to the Misc registers could lead to erratic system operation, PROT[7:0] should be set to other than 0x96 unless the Misc registers must be rewritten.
 - PCLK1 must be supplied from the CMU to access the Misc registers.
For details of the clock control bit, see the “Clock Management Unit (CMU)” chapter.

29.1 RTC Wait Control

The MISC_RTCWT register contains RTCWT[2:0] to set the number of wait cycles to be inserted when accessing the RTC registers. The number of wait cycles should be set according to the MCLK clock frequency.

Table 29.1.1 RTCWT[2:0] (RTC Wait Cycle) Settings

RTCWT[2:0]	Number of wait cycles	MCLK frequency
0x7	7 cycles	f _{MCLK} ≤ 60 MHz
0x6	6 cycles	
0x5	5 cycles	
0x4	4 cycles	
0x3	3 cycles	
0x2	2 cycles	
0x1	1 cycle	
0x0	0 cycles	Cannot be set ^(Note)

(Default: 0x7)

Note: The S1C33L26 RTC cannot operate if RTCWT[2:0] is set to 0x0 (0 wait cycles).

29.2 Internal RAM Wait Control

The MISC_RAMWT register contains COREWT and BUSWT to set the number of wait cycles (in MCLK) to be inserted when accessing IRAM (Area 0) and IVRAM (Area 3), respectively.

When the control bit is set to 1 (default), one wait cycle will be inserted when the IRAM or IVRAM is accessed.

When the control bit is set to 0, no wait cycle will be inserted.

If IVRAM is relocated to Area 0, the COREWT setting is ineffective.

Table 29.2.1 COREWT (IRAM Wait Cycle) Settings

COREWT	Number of wait cycles	MCLK frequency
1	1 cycle	f _{MCLK} ≤ 60 MHz
0	0 cycles	

(Default: 1)

Table 29.2.2 BUSWT (IVRAM Wait Cycle) Settings

BUSWT	Number of wait cycles	MCLK frequency
1	1 cycle	f _{MCLK} ≤ 60 MHz
0	0 cycles	

(Default: 1)

29.3 USB Settings

29.3.1 USB Wait Control

The MISC_USB register contains USBWT[2:0] to set the number of wait cycles to be inserted when accessing the USB registers. The number of wait cycles should be set according to the MCLK clock frequency.

Table 29.3.1.1 USBWT[2:0] (USB Wait Cycle) Settings

USBWT[2:0]	Number of wait cycles	MCLK frequency
0x7	7 cycles	f _{MCLK} ≤ 60 MHz
0x6	6 cycles	f _{MCLK} ≤ 56 MHz
0x5	5 cycles	f _{MCLK} ≤ 45 MHz
0x4	4 cycles	f _{MCLK} ≤ 36 MHz
0x3	3 cycles	f _{MCLK} ≤ 24 MHz
0x2	2 cycles	f _{MCLK} ≤ 16 MHz
0x1	1 cycle	f _{MCLK} < 8 MHz
0x0	0 cycles	

(Default: 0x7)

29.3.2 Snooze Control

The MISC_USB register contains USBSNZ that controls Snooze mode for the USB function controller. Setting USBSNZ to 1 enables Snooze mode.

Refer to the “USB Function Controller (USB)” chapter for details on control of the USB function controller.

29.3.3 USB Interrupt Enable

The MISC_USB register contains USBINTEN that enables or disables the USB function controller to generate interrupts. Setting USBINTEN to 1 enables USB interrupts; setting to 0 disables USB interrupts.

29.4 RAM Location

The MISC_RAM_LOC register contains IVRAM_LOC to select the IVRAM location and DSTRAM_CFG to select the DSTRAM location.

IVRAM is located in Area 3 by default (IVRAM_LOC = 1) and is used as the internal VRAM. Setting IVRAM_LOC to 0 relocates IVRAM in Area 0 and it can be used as a general-purpose RAM.

DSTRAM is located in Area 3 by default (DSTRAM_CFG = 0) and is used as a general-purpose RAM. Setting DSTRAM_CFG to 1 relocates it to the LCDC module as LUTRAM (look-up table RAM). In this case LUTRAM cannot be accessed by the CPU.

29.5 Boot Register

The MISC_BOOT register is used to confirm the boot device and configure #CE10 boot conditions.

BOOT[3:0] indicates the boot device that has been specified by the BOOT pin.

Table 29.5.1 BOOT[3:0] Bits

BOOT[3:0]	Boot mode
0b1000	SPI/RS232C boot
0b0100	NOR Flash/external ROM boot
Other	Reserved

The MISC_BOOT register contains another control bit BOOT_ENA that is used for the booting process by the internal boot sequencer.

Note: When programming a Flash memory on the target board, BOOT_ENA must be set to 0. Be sure to avoid changing the boot mode when writing data to the MISC_BOOT register.

For more information on booting, see “Boot” in Appendix.

29.6 Control Register Details

Table 29.6.1 List of Misc Registers

Address	Register name		Function
0x300010	MISC_RTCWT	RTC Wait Control Register	Configure RTC access cycles
0x300012	MISC_USB	USB Configuration Register	Enable USB interrupt and configure access cycle
0x300014	MISC_RAMWT	Internal RAM Wait Control Register	Configure IRAM and IVRAM access cycles
0x300016	MISC_BOOT	Boot Register	Indicate/set boot conditions
0x300018	MISC_RAM_LOC	RAM Location Select Register	Select area for locating RAM
0x300020	MISC_PROTECT	Misc Protect Register	Enable/disable Misc register write protection

The Misc registers are described in detail below. These are 8-bit registers.

Note: The Misc registers at addresses 0x300010–0x300018 are write-protected. Before the Misc registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to PROT[7:0]/MISC_PROTECT register. Note that since unnecessary rewrites to the Misc registers could lead to erratic system operation, PROT[7:0] should be set to other than 0x96 unless the Misc registers must be rewritten.

RTC Wait Control Register (MISC_RTCWT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Wait Control Register (MISC_RTCWT)	0x300010 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	RTCWT[2:0]	RTC register access wait control	0 to 7 cycles	0x7	R/W	Write-protected

D[7:3] Reserved

D[2:0] RTCWT[2:0]: RTC Register Access Wait Control Bits

Sets the number of wait cycles to be inserted when accessing the RTC control register.

Table 29.6.2 RTCWT[2:0] (RTC Wait Cycle) Settings

RTCWT[2:0]	Number of wait cycles	MCLK frequency
0x7	7 cycles	fMCLK ≤ 60 MHz
0x6	6 cycles	
0x5	5 cycles	
0x4	4 cycles	
0x3	3 cycles	
0x2	2 cycles	
0x1	1 cycle	
0x0	0 cycles	Cannot be set (Note)

(Default: 0x7)

The number of wait cycles should be set according to the MCLK clock frequency.

Note: The S1C33L26 RTC cannot operate if RTCWT[2:0] is set to 0x0 (0 wait cycles).

USB Configuration Register (MISC_USB)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USB Configuration Register (MISC_USB)	0x300012 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6	USBINTEN	USB interrupt enable	1 Enable 0 Disable	0	R/W	Write-protected	
		D5	USBSNZ	USB snooze control	1 Enable 0 Disable	0	R/W		
		D4–3	–	reserved	–	–	–	–	0 when being read.
		D2–0	USBWT[2:0]	USB register access wait control	0 to 7 cycles	0x7	R/W	Write-protected	

D7 Reserved

D6 USBINTEN: USB Interrupt Enable Bit

Enables or disables USB interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

For more information on the causes of USB interrupts, see the “USB Function Controller (USB)” chapter.

D5 USBSNZ: USB Snooze Control Bit

Enables or disables the USB to enter snooze mode.

1 (R/W): Enabled

0 (R/W): Disabled (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 the “Snooze” section in the “USB Function Controller (USB)” chapter.

D[4:3] Reserved

D[2:0] USBWT[2:0]: USB Register Access Wait Control Bits

Sets the number of wait cycles to be inserted when accessing the USB control register.

Table 29.6.3 USBWT[2:0] (USB Wait Cycle) Settings

USBWT[2:0]	Number of wait cycles	MCLK frequency
0x7	7 cycles	f _{MCLK} ≤ 60 MHz
0x6	6 cycles	f _{MCLK} ≤ 56 MHz
0x5	5 cycles	f _{MCLK} ≤ 45 MHz
0x4	4 cycles	f _{MCLK} ≤ 36 MHz
0x3	3 cycles	f _{MCLK} ≤ 24 MHz
0x2	2 cycles	f _{MCLK} ≤ 16 MHz
0x1	1 cycle	f _{MCLK} < 8 MHz
0x0	0 cycles	

(Default: 0x7)

The number of wait cycles should be set according to the MCLK clock frequency.

Internal RAM Wait Control Register (MISC_RAMWT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal RAM Wait Control Register (MISC_RAMWT)	0x300014 (8 bits)	D7–2	–	reserved		–	–	0 when being read.
		D1	COREWT	IRAM (12KB) access wait control	1 1 cycle 0 0 cycles	1	R/W	Write-protected
		D0	BUSWT	IVRAM (20KB) access wait control	1 1 cycle 0 0 cycles	1	R/W	

D[7:2] Reserved

D1 COREWT: IRAM (12KB) Access Wait Control Bit

Sets the number of wait cycles to be inserted when accessing IRAM (Area 0).

1 (R/W): 1 cycle (default)

0 (R/W): 0 cycles (no wait inserted)

Table 29.6.4 COREWT (IRAM Wait Cycle) Settings

COREWT	Number of wait cycles	MCLK frequency
1	1 cycle	f _{MCLK} ≤ 60 MHz
0	0 cycles	

D0 BUSWT: IVRAM (20KB) Access Wait Control Bit

Sets the number of wait cycles to be inserted when accessing IVRAM (Area 3).

1 (R/W): 1 cycle (default)

0 (R/W): 0 cycles (no wait inserted)

If IVRAM is relocated to Area 0, the COREWT setting is ineffective.

Table 29.6.5 BUSWT (IVRAM Wait Cycle) Settings

BUSWT	Number of wait cycles	MCLK frequency
1	1 cycle	f _{MCLK} ≤ 60 MHz
0	0 cycles	

Boot Register (MISC_BOOT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
BOOT Register (MISC_BOOT)	0x300016 (8 bits)	D7-4	BOOT[3:0]	Boot mode indicator	BOOT[3:0] 0b1000 SPI/RS232C 0b0100 NOR/ROM Other reserved	*	R	* Depends on the BOOT pin status at initial reset
		D3-2	–	reserved	–	–	–	0 when being read.
		D1	BOOT_ENA	#CE10 area boot enable	1 Internal 0 External	1	R/W	Write-protected
		D0	–	reserved	–	–	–	0 when being read.

D[7:4] BOOT[3:0]: Boot Mode Indicator Bits

Indicates the boot device that has been specified by the BOOT pin.

Table 29.6.6 BOOT[3:0] Bits

BOOT[3:0]	Boot mode
0b1000	SPI/RS232C boot
0b0100	NOR Flash/external ROM boot
Other	Reserved

BOOT[3:0] is set by the system boot sequencer. Do not alter these bit values from the user routine.

D[3:2] Reserved

D1 BOOT_ENA: #CE10 Area Boot Enable Bit

Enables fetching the RESET vector from the #CE10 external area (0xc00000).

1 (R/W): Internal boot (default)

0 (R/W): External boot

BOOT_ENA is set by the system boot sequencer. When programming a Flash memory using ICD33, BOOT_ENA must be set to 0.

D0 Reserved

RAM Location Select Register (MISC_RAM_LOC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RAM Location Select Register (MISC_RAM_LOC)	0x300018 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	DSTRAM_CFG	DSTRAM configuration	1 LUTRAM 0 DSTRAM	0	R/W	Write-protected
		D3-1	–	reserved	–	–	–	0 when being read.
		D0	IVRAM_LOC	IVRAM location select	1 Area 3 0 Area 0	1	R/W	Write-protected

D[7:5] Reserved

D4 DSTRAM_CFG: DSTRAM Configuration Bit

Selects whether DSTRAM in Area 3 is used as DSTRAM or LUTRAM.

1 (R/W): LUTRAM

0 (R/W): DSTRAM (default)

By default (DSTRAM_CFG = 0), DSTRAM is located in Area 3 and is used as a general-purpose RAM or DMA control table memory.

Setting DSTRAM_CFG to 1 configures DSTRAM to LUTRAM to be used as the color look-up tables for the LCDC. When using the color look-up tables, set DSTRAM_CFG to 1 and LUTPASS/LCDC_DISPMD register to 0. For more information on the color look-up tables, see the “LCD Controller (LCDC)” chapter.

Note: When DSTRAM is switched to LUTRAM, locate the DMAC control table in IVRAM (Area 3) or an external RAM.

D[3:1] Reserved

D0 IVRAM_LOC: IVRAM Location Select Bit

Selects the 20KB IVRAM location.

1 (R/W): Area 3 (default)

0 (R/W): Area 0

By default (IVRAM_LOC = 1), IVRAM is located in Area 3 and is used as the internal VRAM. Setting IVRAM_LOC to 0 relocate IVRAM in Area 0 and it can be used as a general-purpose RAM. In this case, 20KB IVRAM follows 12KB IRAM.

Note: When IVRAM is switched to Area 0 or Area 3, the contents of IVRAM may be destroyed. Therefore, write data to IVRAM after the location is changed.

Misc Protect Register (MISC_PROTECT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Misc Protect Register (MISC_PROTECT)	0x300020 (8 bits)	D7-0	PROT[7:0]	Misc register write-protect flag	Writing 10010110 (0x96) removes the write protection of the Misc registers (0x300010-0x300018). Writing another value set the write protection.	0x0	R/W	

D[7:0] PROT[7:0]: Misc Register Write-Protect Flag Bits

Enables or disables write protection of the Misc registers (0x300010-0x300018).

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 0x300018, write data 0x96 to PROT[7:0] to disable write protection. If PROT[7:0] 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 the register will not be altered even though the instruction may have been executed without an error. Once PROT[7:0] 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, PROT[7:0] should be set to other than 0x96 to prevent accidental writing to the Misc registers.

30 Divider (DIV)

The S1C33L26 has an embedded coprocessor that provides a signed/unsigned 16 ÷ 16-bit division function. This section explains how to use the divider.

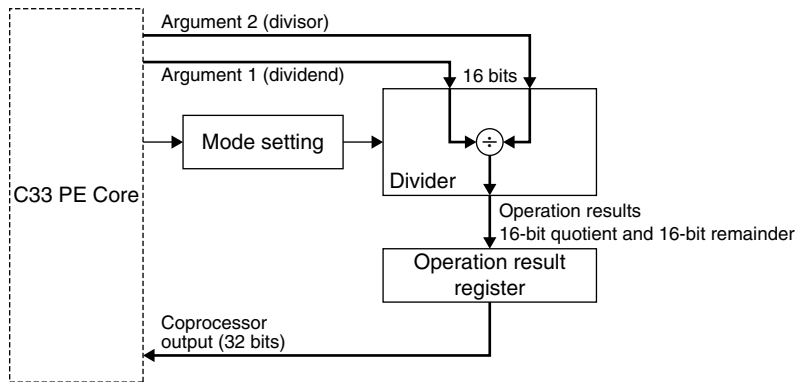


Figure 30.1 Divider Block Diagram

Operation mode

The coprocessor operates according to the operation mode specified by the application program. As listed in Table 30.1, the coprocessor supports two operations.

The operation mode can be specified with a 6-bit data by writing it to the mode setting register in the divider. Use a “do.c” instruction for this writing.

do.c imm6 imm6[5:0] is written to the mode setting register.

Table 30.1 Mode Settings

Setting value (D[5:0])	Operation mode
0x8	Unsigned division mode Performs unsigned division.
0x9	Signed division mode Performs signed division.
Other	Reserved

Division

The division function performs “A (16 bits) = B (16 bits) ÷ C (16 bits), D (16 bits) = remainder.”

To perform a division, set the operation mode to 0x8 (unsigned division) or 0x9 (signed division). Then send the 16-bit dividend (B) and 16-bit divisor (C) to the divider using a “ld.c imm4, %rs” instruction. The quotient (A) and the remainder (D) will be stored in the low-order 16 bits and the high-order 16 bits of the operation result register, respectively. The quotient and remainder can be loaded to a CPU register using a “ld.c %rd, imm4” instruction.

Example:

```
do.c 0x8 ; Sets the operation mode (unsigned division mode).
ld.c 0x0, %r0 ; Sends the dividend set in %r0.
ld.c 0x1, %r1 ; Sends the divisor set in %r1. The divider performs “%r0 ÷ %r1.”
ld.c %r0, 0x0 ; Loads the result to %r0. (%r0[31:16] = remainder, %r0[15:0] = quotient)
; 0x0 (imm4) is a dummy parameter (any value can be specified).
```

Notes:

- Uncertain quotient and remainder will result if the divisor is set to 0.

- Since the DIV module does not support the C, V, Z, and N flags, the “ld.cf” instruction cannot be used. Uncertain values may be set to the C, V, Z, and N flags in PSR of the C33 PE Core if used.
- A -32,768 quotient and a 0 remainder will result when the dividend is set to -32,768 and the divisor is set -1.

31 Electrical Characteristics

31.1 Absolute Maximum Rating

(V_{SS} = 0V)

Item	Symbol	Condition	Rated value	Unit
Internal logic and 1.8 V system I/O power supply voltage *1	LV _{DD}	–	V _{SS} - 0.3 to 2.5	V
RTC power supply voltage *1	RTC _V DD	–	V _{SS} - 0.3 to 2.5	V
PLL power supply voltage *1	PLL _V DD	–	V _{SS} - 0.3 to 2.5	V
3.3 V system I/O power supply voltage *1	HV _{DD}	–	V _{SS} - 0.3 to 4.0	V
ADC power supply voltage *1	AV _{DD}	–	V _{SS} - 0.3 to 4.0	V
3.3 V system input voltage	HV _I	–	V _{SS} - 0.3 to HV _{DD} + 0.5	V
1.8 V system input voltage	LV _I	–	V _{SS} - 0.3 to LV _{DD} + 0.5	V
Analog input voltage	AV _I	–	V _{SS} - 0.3 to AV _{DD} + 0.5	V
3.3 V system output voltage	HV _O	–	V _{SS} - 0.3 to HV _{DD} + 0.5	V
1.8 V system output voltage	LV _O	–	V _{SS} - 0.3 to LV _{DD} + 0.5	V
High level output current	IOH	1 pin	-10	mA
		Total of all pins	-40	mA
Low level output current	IOL	1 pin	10	mA
		Total of all pins	40	mA
Storage temperature	Tstg	–	-65 to 150	°C

*1) HV_{DD}/AV_{DD} ≥ LV_{DD}/RTC_VDD/PLL_VDDLV_{DD} = RTC_VDD = PLL_VDD*2) The maximum input voltage range of the #STBY pin is V_{SS} - 0.3 V to 4.0 V.

31.2 Recommended Operating Conditions

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Power supply voltage (High) *1	HV _{DD} AV _{DD}	When USB is not used	2.70	3.30	3.60	V
		When USB is used	3.00	3.30	3.60	V
Power supply voltage (Low) *1	LV _{DD} RTC _V DD PLL _V DD	Crystal oscillator or external clock input	1.65	1.80	1.95	V
		Ceramic oscillator	1.70	1.80	1.90	V
Input voltage	HV _I LV _I AV _I	–	-0.3	–	HV _{DD} + 0.3	V
		–	-0.3	–	LV _{DD} + 0.3	V
		–	-0.3	–	AV _{DD} + 0.3	V
Ambient temperature	Ta	Crystal oscillator or external clock input	-40	25	85	°C
		Ceramic oscillator	0	25	70	°C
CPU operating frequency	fCPU	–	–	–	60	MHz
Bus operating frequency	fBUS	–	–	–	60	MHz
MCLKI oscillation frequency	fOSC3	–	–	–	48	MHz
RTCCLKI oscillation frequency	fOSC1	–	–	32.768	–	kHz
Input rise time (normal input)	tr1	–	–	–	50	ns
Input fall time (normal input)	tf1	–	–	–	50	ns
Input rise time (Schmitt input)	tr2	–	–	–	5	ms
Input fall time (Schmitt input)	tf2	–	–	–	5	ms

*1) HV_{DD}/AV_{DD} ≥ LV_{DD}/RTC_VDD/PLL_VDDLV_{DD} = RTC_VDD = PLL_VDD*2) The recommended input voltage range of the #STBY pin is V_{SS} - 0.3 V to 3.6 V.

31.3 DC Characteristics

HV_{DD} = AV_{DD} = 3.0 to 3.6 V

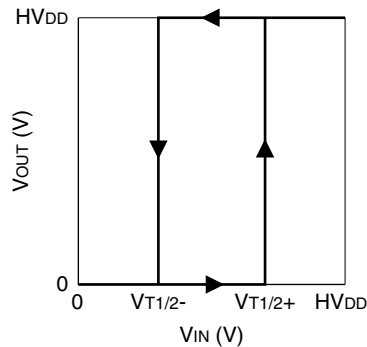
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Input leakage current	I _{LI}	HV _{DD} = 3.6V, LV _{DD} = 1.95V, HV _{IN} = HV _{DD} , LV _{IN} = LV _{DD} , V _{IL} = 0V	-5	-	5	μA
Off-state leakage current	I _{OZ}	HV _{DD} = 3.6V, LV _{DD} = 1.95V, HV _{IN} = HV _{DD} , LV _{IN} = LV _{DD} , V _{IL} = 0V	-5	-	5	μA
HV_{DD}, AV_{DD} system I/O						
High level output voltage (TYPE1)	V _{OH1H}	HV _{DD} = 3.0V, I _{OH} = -2mA	HV _{DD} - 0.4	-	-	V
Low level output voltage (TYPE1)	V _{OL1H}	HV _{DD} = 3.0V, I _{OL} = 2mA	-	-	0.4	V
High level output voltage (TYPE2)	V _{OH2H}	HV _{DD} = 3.0V, I _{OH} = -4mA	HV _{DD} - 0.4	-	-	V
Low level output voltage (TYPE2)	V _{OL2H}	HV _{DD} = 3.0V, I _{OL} = 4mA	-	-	0.4	V
High level output voltage (TYPE3)	V _{OH3H}	HV _{DD} = 3.0V, I _{OH} = -8mA	HV _{DD} - 0.4	-	-	V
Low level output voltage (TYPE3)	V _{OL3H}	HV _{DD} = 3.0V, I _{OL} = 8mA	-	-	0.4	V
High level input voltage (LVTTTL)	V _{IH1H}	HV _{DD} = 3.6V	2	-	HV _{DD} + 0.3	V
Low level input voltage (LVTTTL)	V _{IL1H}	HV _{DD} = 3.0V	-0.3	-	0.8	V
High level input voltage (LVCMOS)	V _{IH2H}	HV _{DD} = 3.6V	2.2	-	HV _{DD} + 0.3	V
Low level input voltage (LVCMOS)	V _{IL2H}	HV _{DD} = 3.0V	-	-	0.8	V
Positive trigger input voltage (LVCMOS Schmitt)	V _{T1+}	HV _{DD} = 3.6V, LV _{DD} = 1.95V	1.4	-	2.7	V
Negative trigger input voltage (LVCMOS Schmitt)	V _{T1-}	HV _{DD} = 3.0V, LV _{DD} = 1.65V	0.6	-	1.8	V
Hysteresis voltage (LVCMOS Schmitt)	ΔV ₁	HV _{DD} = 3.0V, LV _{DD} = 1.65V	0.3	-	-	V
Pull-up resistor (TYPE1)	R _{PLU1H}	V _I = 0V	25	50	120	kΩ
Pull-down resistor (TYPE1)	R _{PLD1H}	V _I = HV _{DD}	25	50	120	kΩ
Pull-up resistor (TYPE2)	R _{PLU2H}	V _I = 0V	50	100	240	kΩ
Pull-down resistor (TYPE2)	R _{PLD2H}	V _I = HV _{DD}	50	100	240	kΩ
High level holding current (Bus hold latch)	I _{IBHH}	HV _{DD} = 3.0V, V _I = 2.0V	-	-	-20	μA
Low level holding current (Bus hold latch)	I _{IBHL}	HV _{DD} = 3.0V, V _I = 0.8V	-	-	17	μA
High level inverting current (Bus hold latch)	I _{IBHH0}	HV _{DD} = 3.6V, V _I = 0.8V	-350	-	-	μA
Low level inverting current (Bus hold latch)	I _{IBHL0}	HV _{DD} = 3.6V, V _I = 2.0V	300	-	-	μA
LV_{DD}, RTCV_{DD} system I/O						
High level output voltage (TYPE1)	V _{OH1L}	LV _{DD} = 1.65V, I _{OH} = -1mA	LV _{DD} - 0.4	-	-	V
Low level output voltage (TYPE1)	V _{OL1L}	LV _{DD} = 1.65V, I _{OL} = 1mA	-	-	0.4	V
High level input voltage (LVCMOS)	V _{IH1L}	LV _{DD} = 1.95V	1.27	-	LV _{DD} + 0.3	V
Low level input voltage (LVCMOS)	V _{IL1L}	LV _{DD} = 1.65V	-0.3	-	0.57	V
Positive trigger input voltage (LVCMOS Schmitt)	V _{T2+}	HV _{DD} = 3.6V, LV _{DD} = 1.95V	0.6	-	1.4	V
Negative trigger input voltage (LVCMOS Schmitt)	V _{T2-}	HV _{DD} = 3.0V, LV _{DD} = 1.65V	0.3	-	1.1	V
Hysteresis voltage (LVCMOS Schmitt)	ΔV ₂	HV _{DD} = 3.0V, LV _{DD} = 1.65V	0.02	-	-	V
Pull-down resistor (TYPE2)	R _{PLD2L}	V _I = LV _{DD}	48	120	300	kΩ
Input pin capacitance	C _I	f = 1MHz, HV _{DD} = 0V	-	-	8	pF
Output pin capacitance	C _O	f = 1MHz, HV _{DD} = 0V	-	-	8	pF
I/O pin capacitance	C _{IO}	f = 1MHz, HV _{DD} = 0V	-	-	8	pF

HV_{DD} = AV_{DD} = 2.7 to 3.6 V

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Input leakage current	I _{LI}	HV _{DD} = 3.6V, LV _{DD} = 1.95V, HV _{IN} = HV _{DD} , LV _{IN} = LV _{DD} , V _{IL} = 0V	-5	-	5	μA
Off-state leakage current	I _{OZ}	HV _{DD} = 3.6V, LV _{DD} = 1.95V, HV _{IN} = HV _{DD} , LV _{IN} = LV _{DD} , V _{IL} = 0V	-5	-	5	μA
HV_{DD}, AV_{DD} system I/O						
High level output voltage (TYPE1)	V _{OH1H}	HV _{DD} = 2.7V, I _{OH} = -1.8mA	HV _{DD} - 0.4	-	-	V
Low level output voltage (TYPE1)	V _{OL1H}	HV _{DD} = 2.7V, I _{OL} = 1.8mA	-	-	0.4	V
High level output voltage (TYPE2)	V _{OH2H}	HV _{DD} = 2.7V, I _{OH} = -3.6mA	HV _{DD} - 0.4	-	-	V
Low level output voltage (TYPE2)	V _{OL2H}	HV _{DD} = 2.7V, I _{OL} = 3.6mA	-	-	0.4	V
High level output voltage (TYPE3)	V _{OH3H}	HV _{DD} = 2.7V, I _{OH} = -7.2mA	HV _{DD} - 0.4	-	-	V
Low level output voltage (TYPE3)	V _{OL3H}	HV _{DD} = 2.7V, I _{OL} = 7.2mA	-	-	0.4	V
High level input voltage (LVTTTL)	V _{IH1H}	HV _{DD} = 3.6V	2	-	HV _{DD} + 0.3	V
Low level input voltage (LVTTTL)	V _{IL1H}	HV _{DD} = 2.7V	-0.3	-	0.7	V
High level input voltage (LVCMOS)	V _{IH2H}	HV _{DD} = 3.6V	2.2	-	HV _{DD} + 0.3	V
Low level input voltage (LVCMOS)	V _{IL2H}	HV _{DD} = 2.7V	-	-	0.7	V
Positive trigger input voltage (LVCMOS Schmitt)	V _{T1+}	HV _{DD} = 3.6V, LV _{DD} = 1.95V	1.4	-	2.7	V

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Negative trigger input voltage (LVCMOS Schmitt)	V_{T1-}	$HV_{DD} = 2.7V, LV_{DD} = 1.65V$	0.5	–	1.8	V
Hysteresis voltage (LVCMOS Schmitt)	ΔV_1	$HV_{DD} = 2.7V, LV_{DD} = 1.65V$	0.27	–	–	V
Pull-up resistor (TYPE1)	R_{PLU1H}	$V_I = 0V$	24	50	150	k Ω
Pull-down resistor (TYPE1)	R_{PLD1H}	$V_I = HV_{DD}$	24	50	150	k Ω
Pull-up resistor (TYPE2)	R_{PLU2H}	$V_I = 0V$	48	100	300	k Ω
Pull-down resistor (TYPE2)	R_{PLD2H}	$V_I = HV_{DD}$	48	100	300	k Ω
High level holding current (Bus hold latch)	I_{HBH}	$HV_{DD} = 2.7V, V_I = 2.0V$	–	–	-15.7	μA
Low level holding current (Bus hold latch)	I_{BHL}	$HV_{DD} = 2.7V, V_I = 0.7V$	–	–	12.2	μA
High level inverting current (Bus hold latch)	I_{BHH0}	$HV_{DD} = 3.6V, V_I = 0.7V$	-350	–	–	μA
Low level inverting current (Bus hold latch)	I_{BHL0}	$HV_{DD} = 3.6V, V_I = 2.0V$	300	–	–	μA
LV_{DD}, RTCV_{DD} system I/O						
High level output voltage (TYPE1)	V_{OH1L}	$LV_{DD} = 1.65V, I_{OH} = -1mA$	$LV_{DD} - 0.4$	–	–	V
Low level output voltage (TYPE1)	V_{OL1L}	$LV_{DD} = 1.65V, I_{OL} = 1mA$	–	–	0.4	V
High level input voltage (LVCMOS)	V_{IH1L}	$LV_{DD} = 1.95V$	1.27	–	$LV_{DD} + 0.3$	V
Low level input voltage (LVCMOS)	V_{IL1L}	$LV_{DD} = 1.65V$	-0.3	–	0.57	V
Positive trigger input voltage (LVCMOS Schmitt)	V_{T2+}	$HV_{DD} = 3.6V, LV_{DD} = 1.95V$	0.6	–	1.4	V
Negative trigger input voltage (LVCMOS Schmitt)	V_{T2-}	$HV_{DD} = 2.7V, LV_{DD} = 1.65V$	0.3	–	1.1	V
Hysteresis voltage (LVCMOS Schmitt)	ΔV_2	$HV_{DD} = 2.7V, LV_{DD} = 1.65V$	0.02	–	–	V
Pull-down resistor (TYPE2)	R_{PLD2L}	$V_I = LV_{DD}$	48	120	300	k Ω
Input pin capacitance	C_I	$f = 1MHz, HV_{DD} = 0V$	–	–	8	pF
Output pin capacitance	C_O	$f = 1MHz, HV_{DD} = 0V$	–	–	8	pF
I/O pin capacitance	C_{IO}	$f = 1MHz, HV_{DD} = 0V$	–	–	8	pF

Schmitt input voltage



31.4 Current Consumption

Operating current

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Power source
Battery backup current	IBKUP	OSC1: Off, RTC: Stop LV _{DD} /HV _{DD} /AV _{DD} : Off #STBY = Low	–	0.04	–	μA	RTCV _{DD}
		OSC1: 32kHz, RTC: Run LV _{DD} /HV _{DD} /AV _{DD} : Off #STBY = Low	–	1.3	–	μA	
Current consumption during execution (CPU/CCU/IRAM) SYSCLK = OSC1, CPU clock = SYSCLK	IDD1	CPU clock = 32kHz *3, *4	–	15	–	μA	LV _{DD}
Current consumption during execution (CPU/CCU/IRAM) SYSCLK = OSC3, CPU clock = SYSCLK	IDD2	CPU clock = 5MHz *1, *3	–	2.3	–	mA	LV _{DD}
		CPU clock = 16MHz *1, *3	–	7.3	–	mA	
		CPU clock = 24MHz *1, *3	–	11	–	mA	
		CPU clock = 33MHz *1, *3	–	15	–	mA	
Current consumption during execution (CPU/CCU/IRAM) OSC3 = 48MHz, SYSCLK = PLL out (60MHz), CPU clock = SYSCLK	IDD3	CPU clock = 48MHz *1, *3	–	22	–	mA	LV _{DD}
		CPU clock = 60MHz *1	–	28	–	mA	

31 ELECTRICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Power source
Current consumption during execution in double frequency mode (CPU/CCU/IRAM) SYSCLK = OSC3, CPU clock = 1/2 SYSCLK	IDD4	SYSCLK = 36MHz *2, *3 CPU clock = 18MHz	–	7	–	mA	LVDD
		SYSCLK = 48MHz *2, *3 CPU clock = 24MHz	–	9	–	mA	
Current consumption during execution in double frequency mode (CPU/CCU/IRAM) SYSCLK = PLL out, CPU clock = 1/2 SYSCLK	IDD5	SYSCLK = 72MHz *2 CPU clock = 36MHz	–	14	–	mA	LVDD
Current consumption during execution (CPU/CCU/IRAM) OSC3 = 48MHz, SYSCLK = OSC3, MCLK = SYSCLK	IDD6	CPU clock = 1/2 MCLK *1, *3	–	15	–	mA	LVDD
		CPU clock = 1/4 MCLK *1, *3	–	11	–	mA	
		CPU clock = 1/8 MCLK *1, *3	–	9	–	mA	
Current consumption in HALT mode (CPU/CCU/IRAM) SYSCLK = OSC1, CPU clock = SYSCLK	IHALT1	CPU clock = 32kHz *3, *4	–	3	–	µA	LVDD
Current consumption in HALT mode (CPU/CCU/IRAM) SYSCLK = OSC3, CPU clock = SYSCLK	IHALT2	CPU clock = 5MHz *1, *3	–	0.5	–	mA	LVDD
		CPU clock = 16MHz *1, *3	–	1.4	–	mA	
		CPU clock = 24MHz *1, *3	–	2.1	–	mA	
		CPU clock = 33MHz *1, *3	–	2.9	–	mA	
Current consumption in HALT mode (CPU/CCU/IRAM) OSC3 = 48MHz, SYSCLK = PLL out (60MHz), CPU clock = SYSCLK	IHALT3	CPU clock = 60MHz *1	–	6	–	mA	LVDD
Current consumption in SLEEP mode SYSCLK = OSC3, CPU clock = SYSCLK	ISLP	OSC3: Off, OSC1: Off	–	1.6	–	µA	LVDD
RTC current consumption	IRTC	RTC: Run, RTCCLK = 32kHz PCLK2 = 48MHz, #STBY = Low	–	1.3	–	µA	RTCVDD
		RTC: Run, RTCCLK = 32kHz PCLK2 = 10MHz, #STBY = High	–	70	–	µA	
		RTC: Run, RTCCLK = 32kHz PCLK2 = 33MHz, #STBY = High	–	210	–	µA	
		RTC: Run, RTCCLK = 32kHz PCLK2 = 48MHz, #STBY = High	–	300	–	µA	
		RTC: Run, RTCCLK = 32kHz PCLK2 = 60MHz, #STBY = High	–	410	–	µA	
USB current consumption during idle	IUSB	USB: Snooze mode USBCLK = 48MHz	–	3.7	–	mA	LVDD
LCDC current consumption during idle	ILCDC	LCDC: Power save mode LCLK = 48MHz	–	0.1	–	mA	LVDD
		LCDC: Normal mode LCLK = 48MHz	–	0.2	–	mA	
BCLK and SDCLK group current consumption during idle (SRAMC, SDRAMC)	IBUS	BCLK, SDCLK = 48MHz	–	4.1	–	mA	LVDD
PCLK1 group current consumption during idle (PSC Ch.0, T8 Ch.0/2/4/6, USI, FSIO Ch.0, T16A5 Ch0/1, I2S, T16P, MISC)	IPCLK1	PCLK1 = 48MHz	–	2.6	–	mA	LVDD
PCLK2 group current consumption during idle (PSC Ch.1, T8 Ch.1/3/5/7, USIL, FSIO Ch.1, WDT, GPIO, BBRAM, ITC, REMC, CCU/SRAMC/SDRAMC/LCDC/RTC registers)	IPCLK2	PCLK2 = 48MHz	–	4.8	–	mA	LVDD
GE current consumption during idle	IGE	When GCLK only is supplied GCLK = 48MHz	–	5.5	–	mA	LVDD
ADC10 current consumption	IADC10	On	–	260	–	µA	AVDD
		Off	–	0.5	–	µA	
PLL current consumption	IPLL	48MHz input → 60MHz output	–	2	–	mA	PLLVD
		Off	–	0.8	–	µA	

*1) • The program is executed in IRAM in Area 0.

- The OSC1 and peripheral module clocks are all disabled (off).
- The SSCG circuit is disabled (off).
- A GPIO port is used as the SDCLK port.
- The GPIO ports are all configured to input with pulled up (no floating input).
- The current consumption is measured by executing a test program that consists of 51% load instructions, 21% arithmetic operation instructions, 10% branch instructions and 18% ext instructions.

- *2) • The program is executed in an external SDRAM (K4S561632E).
 - The peripheral module clocks except OSC1, BCLK, and SDCLK are all disabled (off). (CMU_CLKCTL register = 0x18)
 - The SSCG circuit is disabled (off).
 - The GPIO ports are all configured to input with pulled up (no floating input).
 - The current consumption is measured by executing a test program that consists of 51% load instructions, 21% arithmetic operation instructions, 10% branch instructions and 18% ext instructions.
- *3) The PLL circuit is disabled (off).
- *4) • The program is executed in IRAM in Area 0.
 - The OSC3 and peripheral module clocks are all disabled (off).
 - The SSCG circuit is disabled (off).
 - A GPIO port is used as the SDCLK port.
 - The GPIO ports are all configured to input with pulled up (no floating input).
 - The current consumption is measured by executing a test program that consists of 51% load instructions, 21% arithmetic operation instructions, 10% branch instructions and 18% ext instructions.

31.5 A/D Converter Characteristics

Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, HV_{DD} = AV_{DD} = 2.7 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C, ADST[2:0] = 0x7

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	—		—	10	—	bits
Conversion time *1	—		10	—	1250	μs
Zero scale error	E _{ZS}		-2	—	2	LSB
Full scale error	E _{FS}		-2	—	2	LSB
Integral linearity error	EL		-3	—	3	LSB
Differential linearity error	Ed		-3	—	3	LSB
Permissible signal source impedance	—		—	—	5	kΩ
Analog input capacitance	—		—	—	45	pF

*1) Condition for Min. value: A/D clock = 2MHz. Condition for Max. value: A/D clock = 16kHz.

A/D conversion error

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

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

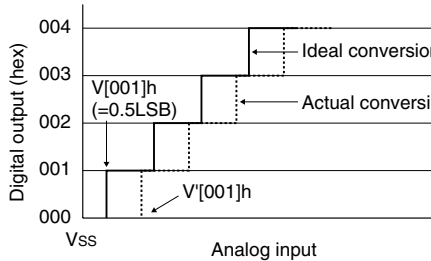
V[3FF]h = Ideal voltage at full-scale point (=1022.5LSB)

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

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

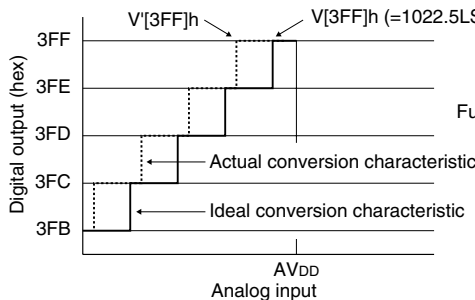
$$1\text{LSB}' = \frac{V'[3FF]h - V'[001]h}{2^{10} - 2}$$

Zero scale error



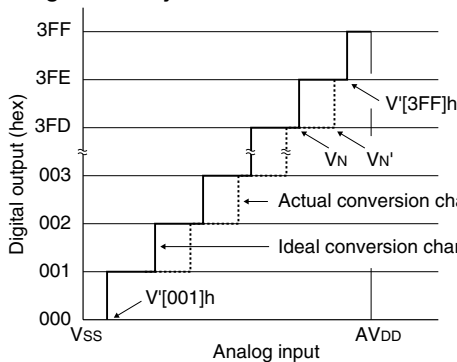
$$\text{Zero scale error } E_{ZS} = \frac{(V'[001]h - 0.5\text{LSB}') - (V[001]h - 0.5\text{LSB})}{1\text{LSB}} \text{ [LSB]}$$

Full scale error



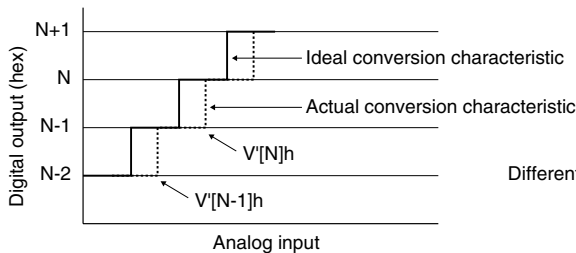
$$\text{Full scale error } E_{FS} = \frac{(V'[3FF]h + 0.5\text{LSB}') - (V[3FF]h + 0.5\text{LSB})}{1\text{LSB}} \text{ [LSB]}$$

■ Integral linearity error



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

■ Differential linearity error



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

31.6 Oscillation Characteristics

Oscillation characteristics vary depending on conditions such as components used (resonator, Rf, Rd, CG, CD) and board pattern. Use the following characteristics as reference values. In particular, when a ceramic or crystal resonator is used, evaluate the components adequately under real operating conditions by mounting them on the board before the external register (Rf, Rd) and capacitor (CG, CD) values are finally decided.

OSC1 crystal oscillation

Unless otherwise specified: LVDD = RTCVDD = 1.65 to 1.95V, VSS = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	tSTA1	*1	–	–	3	s

OSC3 crystal oscillation

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

Unless otherwise specified: LVDD = RTCVDD = 1.65 to 1.95V, VSS = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	tSTA3	*1	–	–	25	ms

OSC3 ceramic oscillation

Unless otherwise specified: LVDD = RTCVDD = 1.65 to 1.95V, VSS = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	tSTA3	*1	–	–	25	ms

*1) When the recommended parts shown in the “Basic External Wiring Diagram” chapter are used

31.7 PLL Characteristics

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

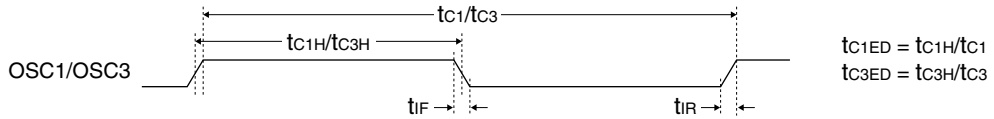
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Input frequency *1	f _{PLLIN}		5	–	48	MHz
Output frequency *2	f _{PLLOUT}		20	–	72	MHz
Output stabilization time	t _{PLL}		–	–	200	μs

*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, ×3, ×4, ×5, ×6, ×7, ×8, ×9, ×10, ×11, ×12, ×13, ×14, ×15, ×16

31.8 AC Characteristics

31.8.1 External Clock Input Characteristics



OSC1 external clock

Unless otherwise specified: LV_{DD} = RTCV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
OSC1 external clock cycle time	t _{c1}	–	30.51	–	μs
OSC1 external clock input duty	t _{c1ED}	45	–	55	%
OSC1 external clock input rise time	t _{rF}	–	–	5	ns
OSC1 external clock input fall time	t _{fR}	–	–	5	ns

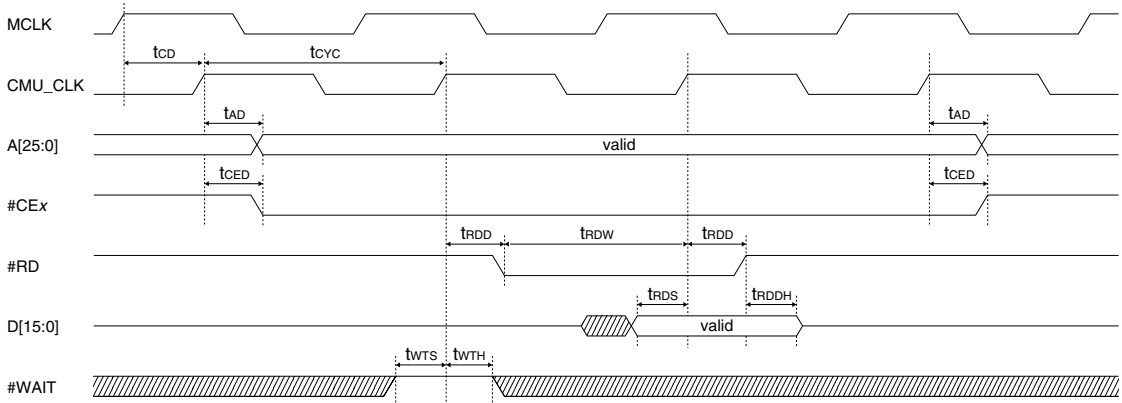
OSC3 external clock

Unless otherwise specified: LV_{DD} = RTCV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

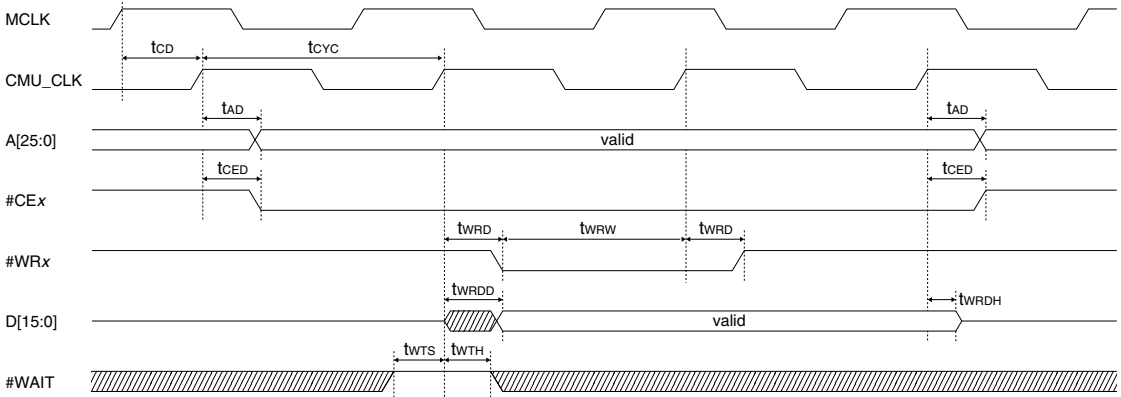
Item	Symbol	Min.	Typ.	Max.	Unit
OSC3 external clock cycle time	t _{c3}	20.83	–	500	ns
OSC3 external clock input duty	t _{c3ED}	45	–	55	%
OSC3 external clock input rise time	t _{rF}	–	–	5	ns
OSC3 external clock input fall time	t _{fR}	–	–	5	ns

31.8.2 SRAMC AC Characteristics

SRAM read cycle



SRAM write cycle



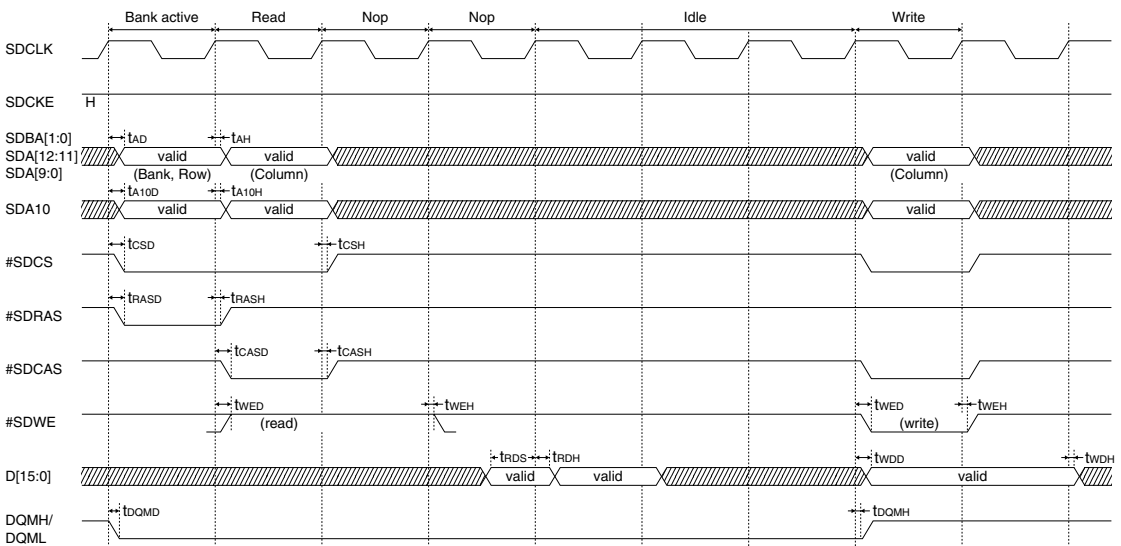
Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, HV_{DD} = 2.7 to 3.6V, V_{SS} = 0V, external load = 50pF, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
CMU_CLK output delay time	t_{CD}	–	–	22	ns
Address delay time	t_{AD}	–	–	13	ns
#CE _x delay time	t_{CED}	–	–	13	ns
Write delay time	t_{WRD}	–	–	13	ns
Write data delay time	t_{WRDD}	–	–	13	ns
Write data hold time	t_{WRDH}	0	–	–	ns
Read delay time	t_{RDD}	–	–	13	ns
Read data setup time	t_{RDS}	13	–	–	ns
Read data hold time	t_{RDDH}	0	–	–	ns
Write signal pulse width	t_{WRW}	$t_{cyc}(1 + WC) - 13$	–	–	ns
Read signal pulse width	t_{RDW}	$t_{cyc}(1 + WC) - 13$	–	–	ns
#WAIT setup time	t_{WTS}	12	–	–	ns
#WAIT hold time	t_{WTH}	0	–	–	ns

WC: Number of wait cycles

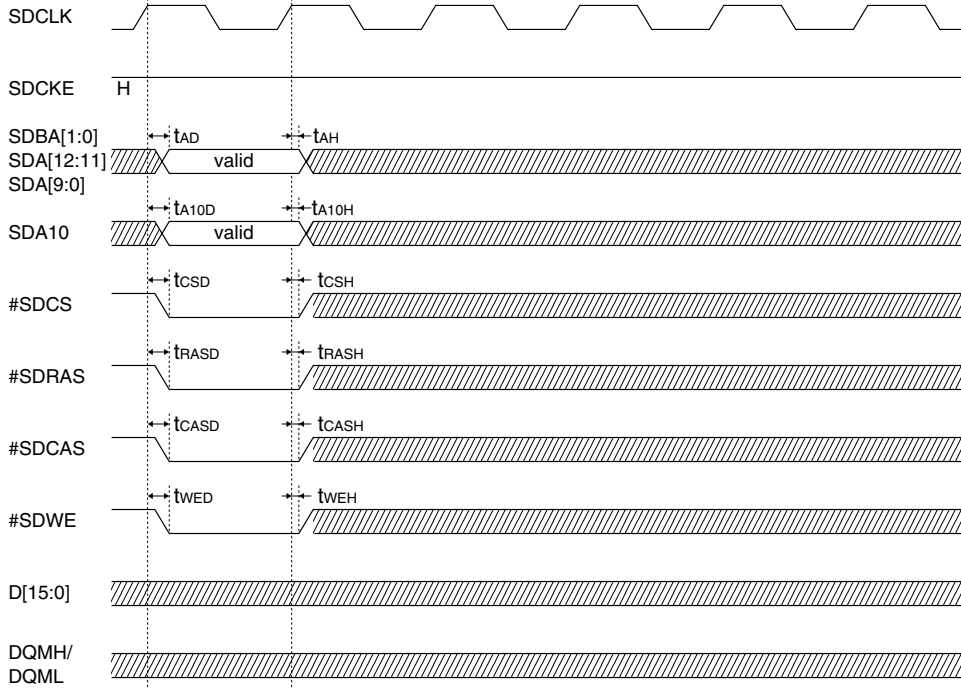
31.8.3 SDRAMC AC Characteristics

SDRAM access cycle

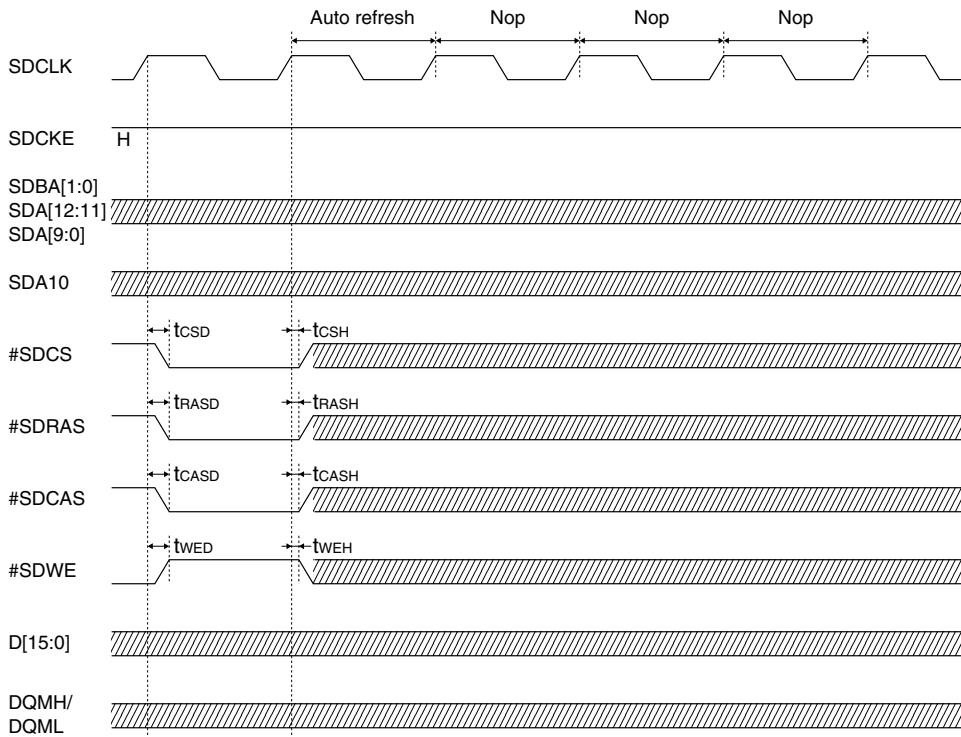


* Read: CAS latency = 2, burst length = 2 Write: single write

SDRAM mode-register-set cycle

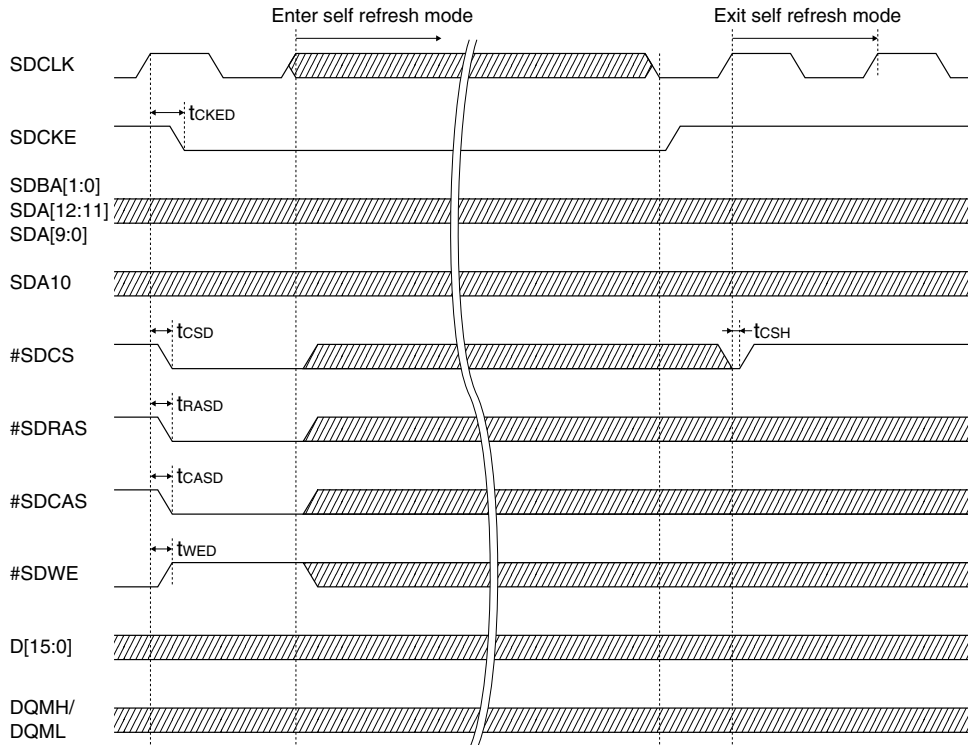


SDRAM auto-refresh cycle



* 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.

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

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

External load conditions: Address bus/data bus = 50pF, SDCLK/control signals = 20pF

SDRAM: Setup time = 3.5ns, Hold time = 1ns, Access time = 6.5ns max.

Item	Symbol	Min.	Typ.	Max.	Unit
Address delay time	tAD	–	–	12.1	ns
Address hold time	tAH	1.3	–	–	ns
SDA10 signal delay time	tA10D	–	–	12.1	ns
SDA10 signal hold time	tA10H	1.3	–	–	ns
#SDCS signal delay time	tCSD	–	–	12.1	ns
#SDCS signal hold time	tCSH	1.3	–	–	ns
#SDRAS signal delay time	tRASD	–	–	12.1	ns
#SDRAS signal hold time	tRASH	1.3	–	–	ns
#SDCAS signal delay time	tCASD	–	–	12.1	ns
#SDCAS signal hold time	tCASH	1.3	–	–	ns
DQMH, DQML signal delay time	tDQMD	–	–	12.1	ns
DQMH, DQML signal hold time	tDQMH	1.3	–	–	ns
SDCKE signal delay time	tCKED	–	–	12.1	ns
SDCKE signal hold time	tCKEH	1.3	–	–	ns
#SDWE signal delay time	tWED	–	–	12.1	ns
#SDWE signal hold time	tWEH	1.3	–	–	ns
Read data setup time	tRDS	6.3	–	–	ns
Read data hold time	tRDH	0	–	–	ns
Write data delay time	tWDD	–	–	12.1	ns
Write data hold time	tWDH	1.3	–	–	ns

Double frequency mode (SDCLK = 2 × MCLK, 72MHz Max.)

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

External load conditions: Address bus/data bus = 30pF, SDCLK/control signals = 20pF

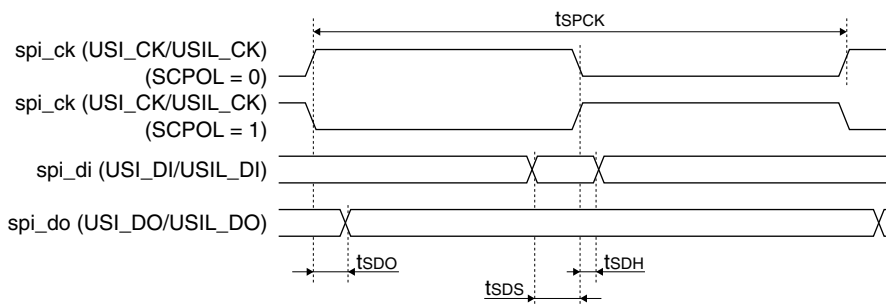
SDRAM: Setup time = 2.5ns, Hold time = 1ns, Access time = 6.5ns max.

Item	Symbol	Min.	Typ.	Max.	Unit
Address delay time	t _{AD}	–	–	10.3	ns
Address hold time	t _{AH}	1.3	–	–	ns
SDA10 signal delay time	t _{A10D}	–	–	10.3	ns
SDA10 signal hold time	t _{A10H}	1.3	–	–	ns
#SDCS signal delay time	t _{CSD}	–	–	10.3	ns
#SDCS signal hold time	t _{CSH}	1.3	–	–	ns
#SDRAS signal delay time	t _{TRASD}	–	–	10.3	ns
#SDRAS signal hold time	t _{RASH}	1.3	–	–	ns
#SDCAS signal delay time	t _{CASD}	–	–	10.3	ns
#SDCAS signal hold time	t _{CASH}	1.3	–	–	ns
DQMH, DQML signal delay time	t _{DQMD}	–	–	10.3	ns
DQMH, DQML signal hold time	t _{DQMH}	1.3	–	–	ns
SDCKE signal delay time	t _{CKED}	–	–	10.3	ns
SDCKE signal hold time	t _{CKEH}	1.3	–	–	ns
#SDWE signal delay time	t _{WED}	–	–	10.3	ns
#SDWE signal hold time	t _{WEH}	1.3	–	–	ns
Read data setup time	t _{RDS}	6.3	–	–	ns
Read data hold time	t _{RDH}	0	–	–	ns
Write data delay time	t _{WDD}	–	–	10.3	ns
Write data hold time	t _{WDH}	1.3	–	–	ns

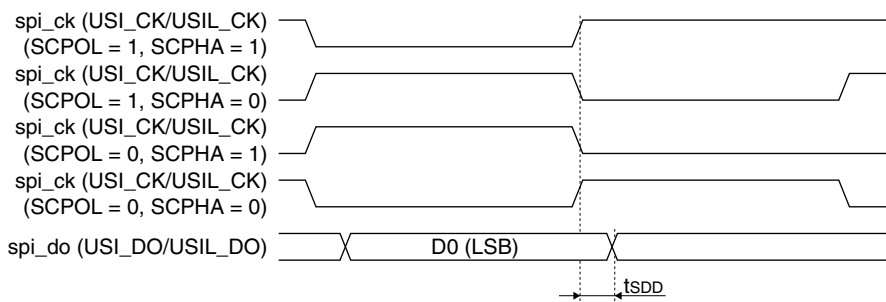
Note: All the signals change at the rising edge of the SDRAM clock.

31.8.4 USI/USIL AC Characteristics

SPI master/slave mode (USI/USIL)



Last bit output timing in slave mode



SPI master mode (normal mode)

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

Item	Symbol	Min.	Typ.	Max.	Unit
spi_ck cycle time	t _{SPCK}	85 + t _{PCLK}	–	–	ns
spi_di setup time	t _{SDS}	85 + t _{PCLK}	–	–	ns
spi_di hold time	t _{SDH}	0	–	–	ns
spi_do output delay time	t _{SDO}	–	–	20	ns

31 ELECTRICAL CHARACTERISTICS

SPI master mode (fast mode)

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

Item	Symbol	Min.	Typ.	Max.	Unit
spi_ck cycle time	tSPCK	85	–	–	ns
spi_di setup time	tSDS	85	–	–	ns
spi_di hold time	tSDH	0	–	–	ns
spi_do output delay time	tSDO	–	–	10	ns

SPI slave mode

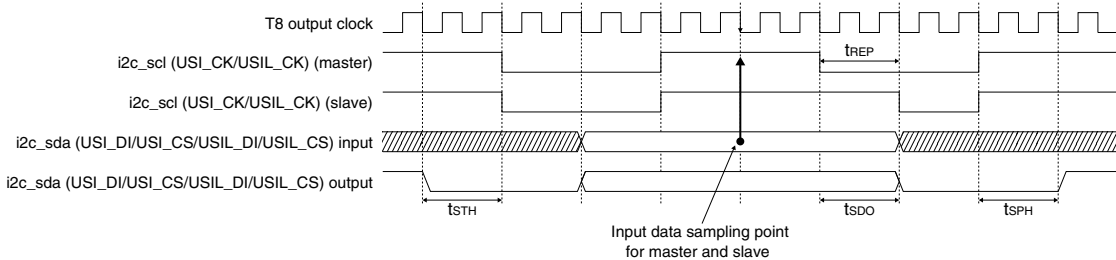
Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, HV_{DD} = 2.7 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
spi_ck cycle time	tSPCK	*2	–	–	ns
spi_di setup time	tSDS	10 + tPCLK	–	–	ns
spi_di hold time	tSDH	10	–	–	ns
spi_do output delay time	tSDO	–	–	80	ns
spi_do last bit delay time	tSDD	2*tPCLK	–	–	ns

*1) tPCLK: PCLK1 or PCLK2 (peripheral module clock supplied from the CMU) clock cycle time

*2) tSPCK(min.) = 80ns if tPCLK ≤ 60ns or tSPCK(min.) = “20 + tPCLK” ns if tPCLK > 60ns

I²C master/slave mode (USI/USIL)



I²C master mode

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

Item	Symbol	Min.	Typ.	Max.	Unit
i2c_scl cycle time	tsCL	2500	–	–	ns
i2c_sda output delay time	tSDO	–	–	2*tT8	ns
Start condition hold time	tSTH	4*tT8	–	–	ns
Stop condition hold time	tSPH	3*tT8	–	–	ns

I²C slave mode

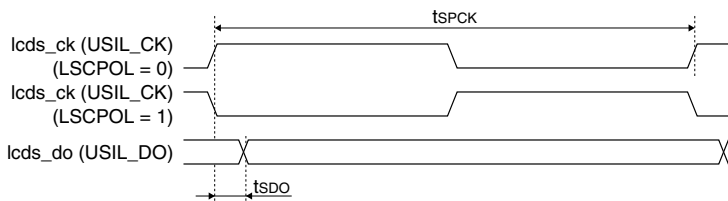
Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, HV_{DD} = 2.7 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
i2c_scl cycle time	tsCL	2500	–	–	ns
i2c_scl input clock response delay time	tREP	2*tT8	–	–	ns
i2c_sda output delay time	tSDO	–	–	2*tT8	ns
Start condition hold time	tSTH	7*tPCLK	–	–	ns
Stop condition hold time	tSPH	7*tPCLK	–	–	ns

tPCLK: PCLK1 or PCLK2 (peripheral module clock supplied from the CMU) clock cycle time

tT8 = T8 output clock cycle time

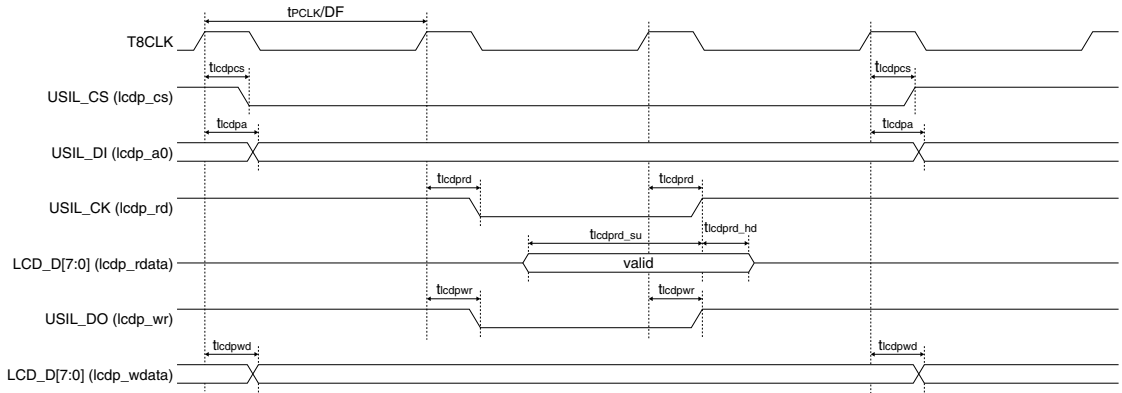
LCD SPI mode (USIL)



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

Item	Symbol	Min.	Typ.	Max.	Unit
spi_ck cycle time	tSPCK	85 + tPCLK	–	–	ns
spi_do output delay time	tSDO	–	–	20	ns

LCD parallel mode (USIL)

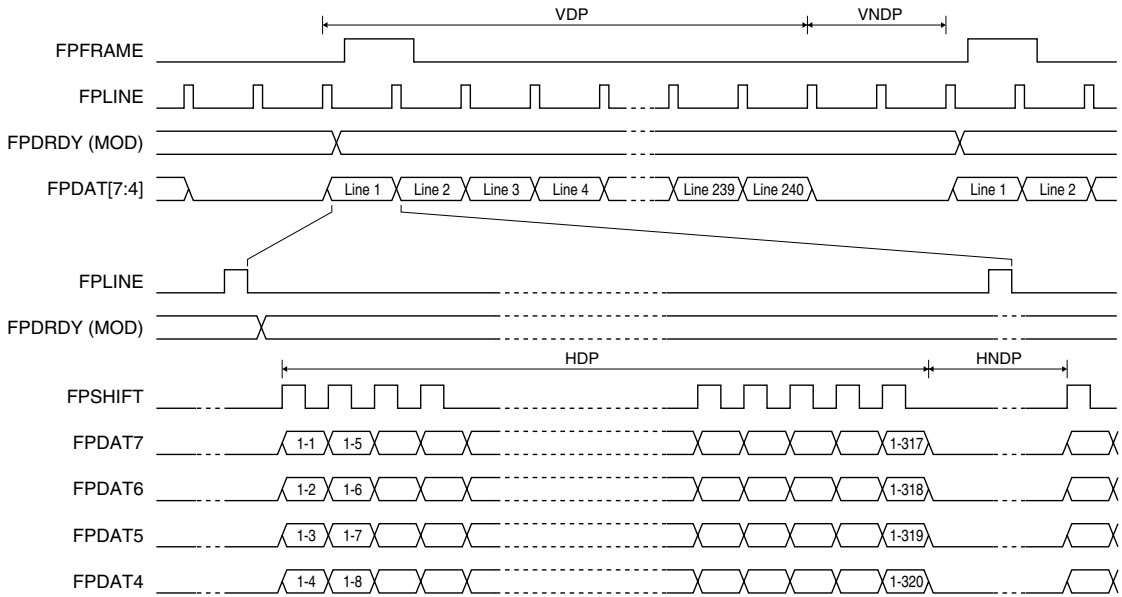


Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, HV_{DD} = 2.7 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C, output load = 50pF

Item	Symbol	Min.	Typ.	Max.	Unit
lcdp_cs output delay time	tcdpcs	–	–	19	ns
lcdp_a0 output delay time	tcdpa	–	–	19	ns
lcdp_rd output delay time	tcdprd	–	–	19	ns
lcdp_rdata setup time	tcdprdsu	17.5	–	–	ns
lcdp_rdata hold time	tcdprdh	0	–	–	ns
lcdp_wr output delay time	tcdpwr	–	–	19	ns
lcdp_wdata output delay time	tcdpwd	–	–	19	ns

31.8.5 LCDC AC Characteristics

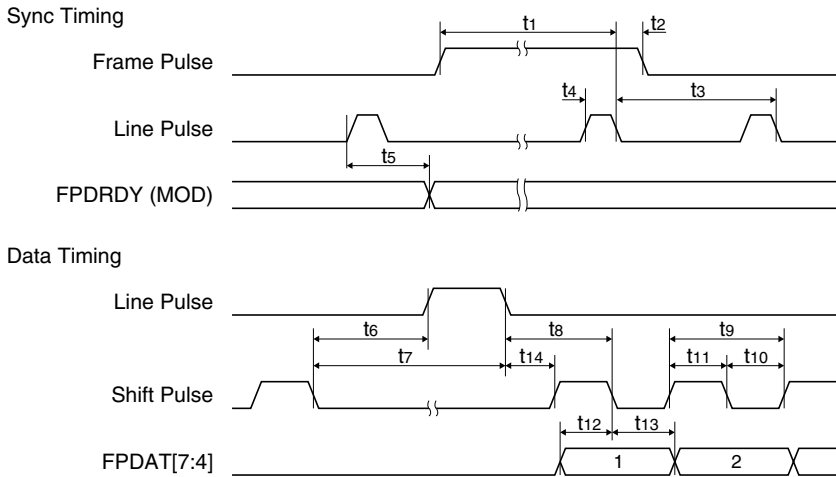
4-bit single monochrome panel timing



* Diagram drawn with 2 FPLINE vertical blank period
 Example timing for a 320 × 240 panel
 For this timing diagram FPSMASK is set to 1

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

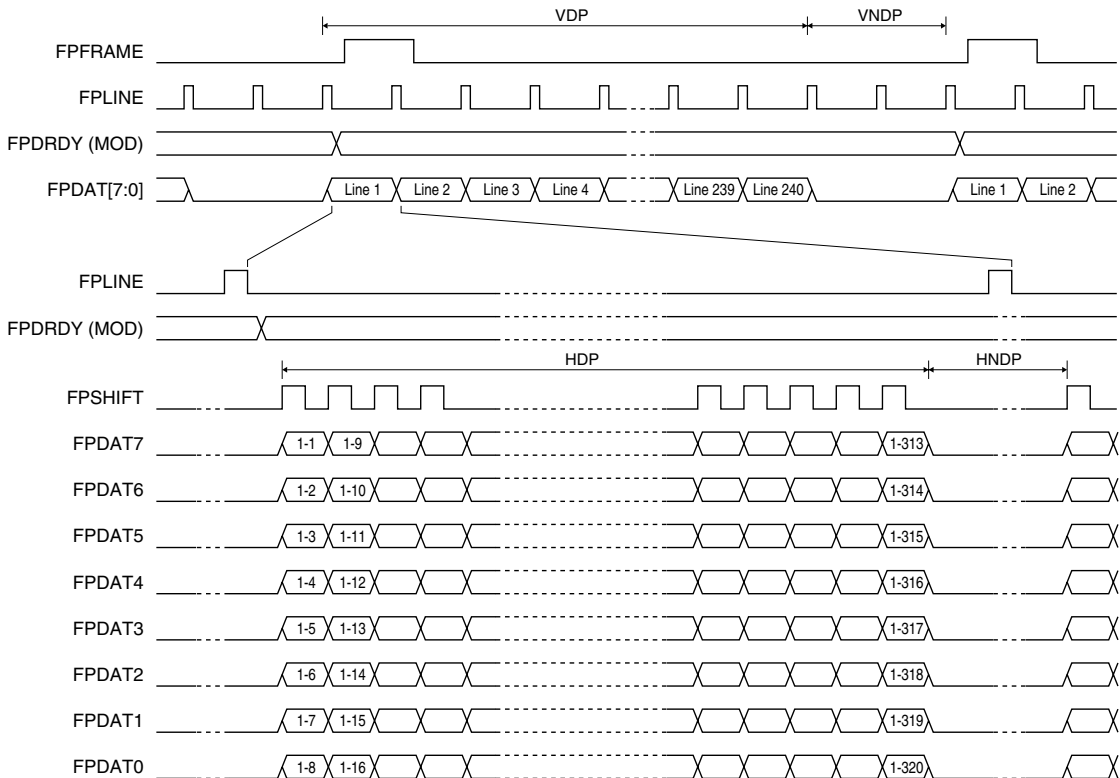
31 ELECTRICAL CHARACTERISTICS



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. $t_{3min} = HDP + HNDP$ (Ts)
 4. $t_{6min} = HNDP + 2$ (Ts)
 5. $t_{7min} = HNDP + 11$ (Ts)

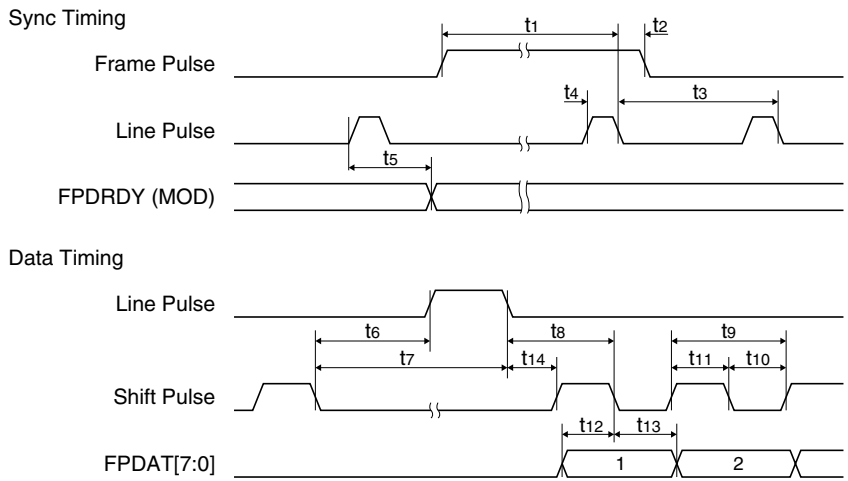
8-bit single monochrome panel timing



* Diagram drawn with 2 FPLINE vertical blank period
 Example timing for a 320 × 240 panel
 For this timing diagram FPSMASK is set to 1

HDP (Horizontal Display Period) = $(\text{HDPCNT}[6:0] + 1) \times 8 \text{ (Ts)}$
 HNDP (Horizontal Non-Display Period) = $(\text{HTCNT}[6:0] - \text{HDPCNT}[6:0]) \times 8 \text{ (Ts)}$
 VDP (Vertical Display Period) = $\text{VDCNT}[9:0] + 1 \text{ (lines)}$
 VNDP (Vertical Non-Display Period) = $\text{VTCNT}[9:0] - \text{VDCNT}[9:0] \text{ (lines)}$

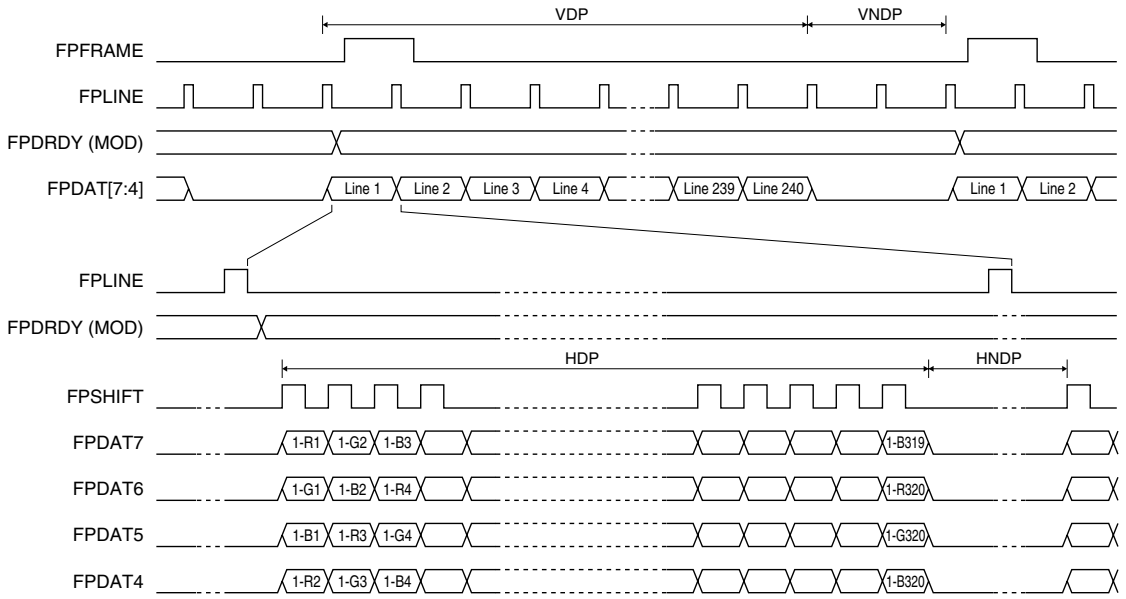
31 ELECTRICAL CHARACTERISTICS



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. $t_{3min} = HDP + HNDP$ (Ts)
 4. $t_{6min} = HNDP + 4$ (Ts)
 5. $t_{7min} = HNDP + 13$ (Ts)

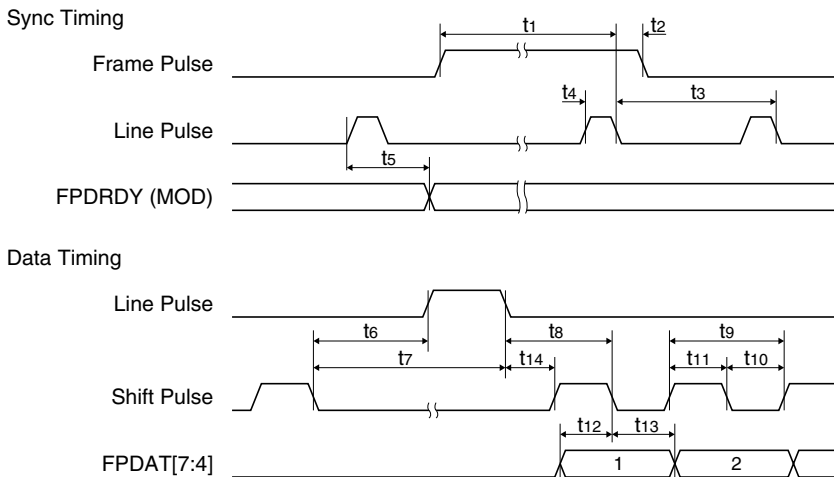
4-bit single color panel timing



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

- HDP (Horizontal Display Period) = $(\text{HDPCNT}[6:0] + 1) \times 8$ (Ts)
- HNRP (Horizontal Non-Display Period) = $(\text{HTCNT}[6:0] - \text{HDPCNT}[6:0]) \times 8$ (Ts)
- VDP (Vertical Display Period) = $\text{VDPCNT}[9:0] + 1$ (lines)
- VNDP (Vertical Non-Display Period) = $\text{VTCNT}[9:0] - \text{VDPCNT}[9:0]$ (lines)

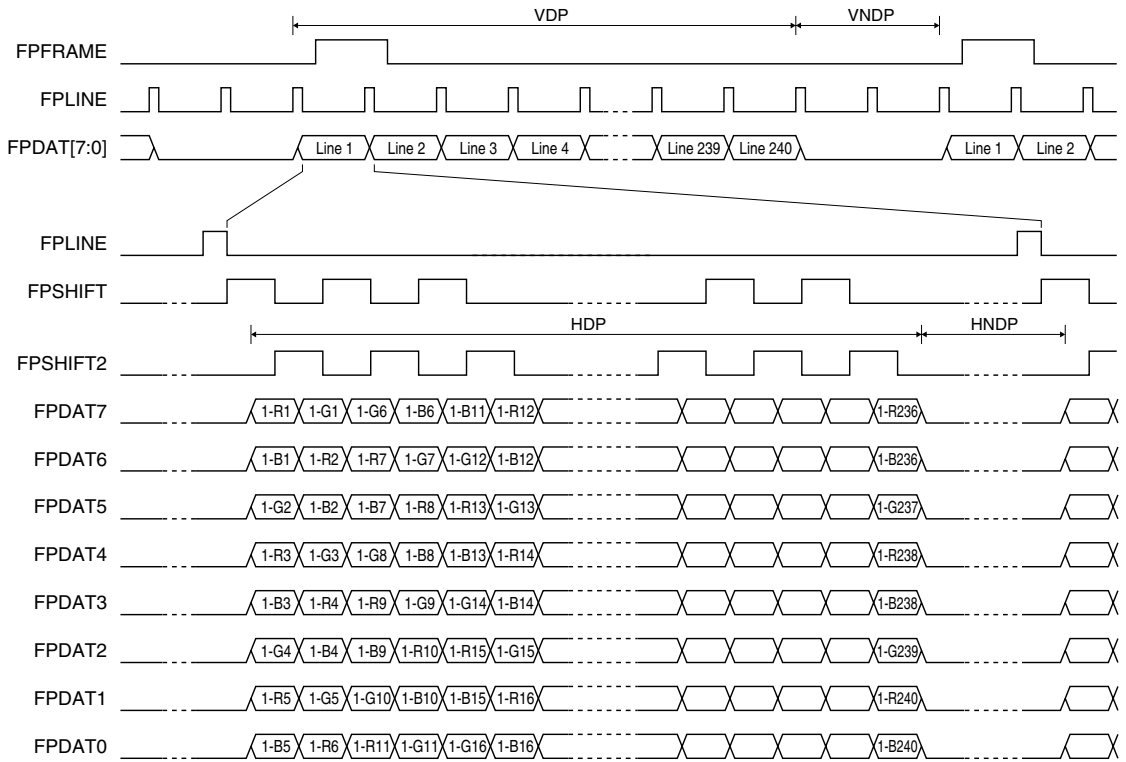
31 ELECTRICAL CHARACTERISTICS



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. $t_{3min} = HDP + HNDP$ (Ts)
 4. $t_{6min} = HNDP + 1.5$ (Ts)
 5. $t_{7min} = HNDP + 10.5$ (Ts)

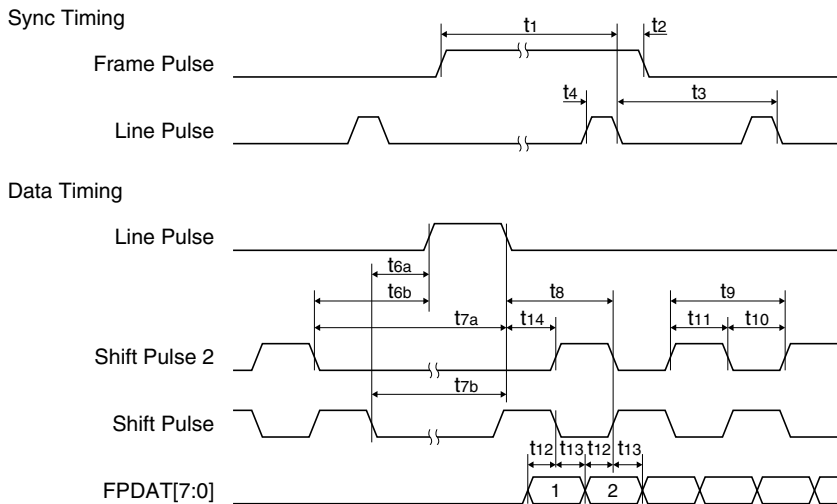
8-bit single color panel timing (Format 1)



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

- HDP (Horizontal Display Period) = (HDPCNT[6:0] + 1) × 8 (Ts)
- HNDP (Horizontal Non-Display Period) = (HTCNT[6:0] - HDPCNT[6:0]) × 8 (Ts)
- VDP (Vertical Display Period) = VDPCNT[9:0] + 1 (lines)
- VNDP (Vertical Non-Display Period) = VTCNT[9:0] - VDPCNT[9:0] (lines)

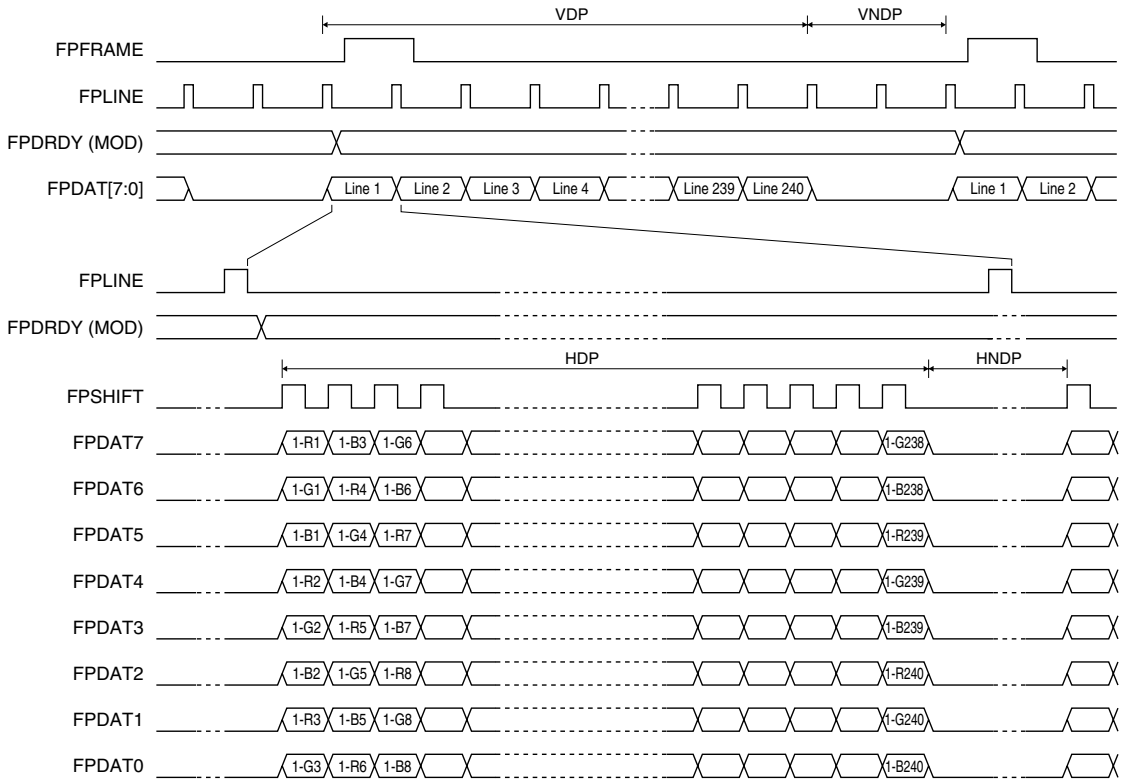
31 ELECTRICAL CHARACTERISTICS



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. $t_{3min} = HDP + HNDP$ (Ts)
 4. $t_{6amin} = HNDP + t_{13} - t_{10} + 1$ (Ts)
 5. $t_{6bmin} = HNDP + t_{13} + 1$ (Ts)
 6. $t_{7amin} = HNDP + 11$ (Ts)
 7. $t_{7bmin} = HNDP + 11 - t_{10}$ (Ts)

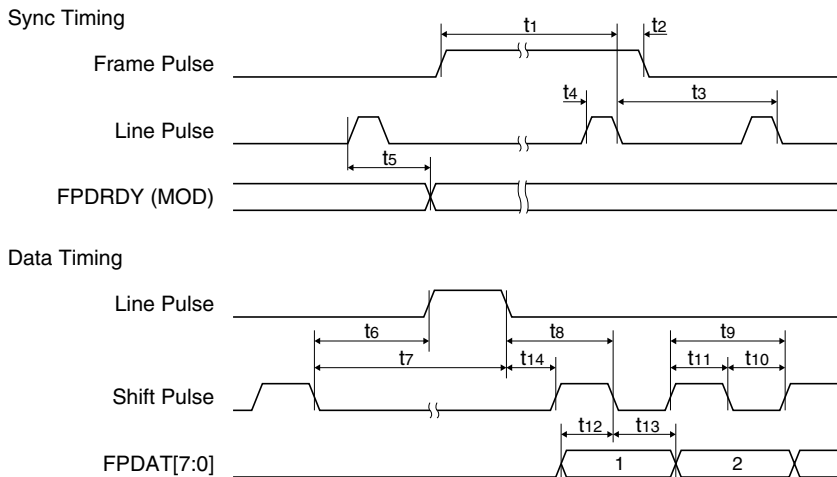
8-bit single color panel timing (Format 2)



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

HDP (Horizontal Display Period) = $(HDCNT[6:0] + 1) \times 8$ (Ts)
 $HNDP$ (Horizontal Non-Display Period) = $(HTCNT[6:0] - HDCNT[6:0]) \times 8$ (Ts)
 VDP (Vertical Display Period) = $VDCNT[9:0] + 1$ (lines)
 $VNDP$ (Vertical Non-Display Period) = $VTCNT[9:0] - VDCNT[9:0]$ (lines)

31 ELECTRICAL CHARACTERISTICS

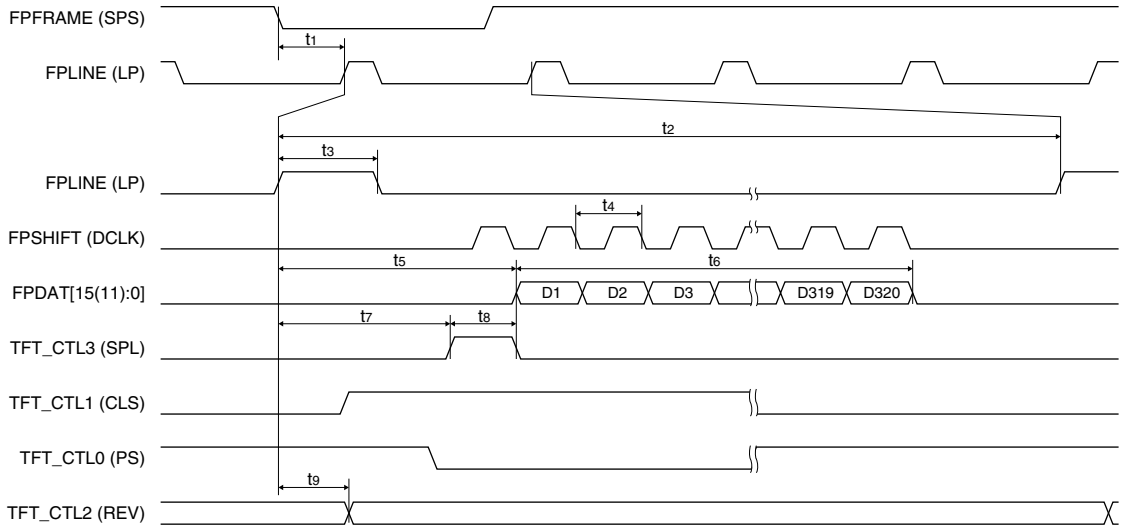


Symbol	Parameter	Min.	Typ.	Max.	Unit
t_1	Frame Pulse setup to Line Pulse falling edge	note 2	–	–	(note 1)
t_2	Frame Pulse hold from Line Pulse falling edge	9	–	–	Ts
t_3	Line Pulse period	note 3	–	–	–
t_4	Line Pulse width	9	–	–	Ts
t_5	MOD delay from Line Pulse rising edge	1	–	–	Ts
t_6	Shift Pulse falling 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. $t_{3min} = HDP + HNDP$ (Ts)
 4. $t_{6min} = HNDP + 1$ (Ts)
 5. $t_{7min} = HNDP + 10$ (Ts)

Generic HR-TFT panel timing

(1) Generic HR-TFT panel horizontal timing

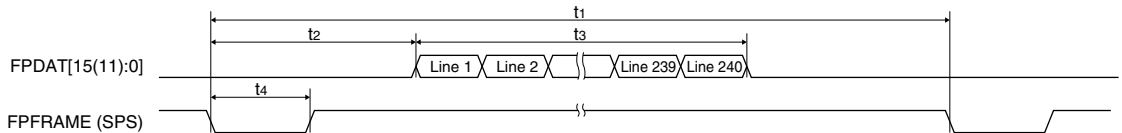


* Example timing for a 320 × 240 panel

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. T_s = pixel clock period
 2. t_{1typ} = $FPLINE_ST[9:0] + 1$ (T_s)
 3. t_{2typ} = $(HTCNT[6:0] + 1) \times 8$ (T_s)
 4. t_{3typ} = $FPLINE_WD[6:0] + 1$ (T_s)
 5. t_{5typ} = $HDPCNT[9:0] + 1$ (T_s)
 6. t_{6typ} = $(HDPCNT[6:0] + 1) \times 8$ (T_s)

(2) Generic HR-TFT panel vertical timing



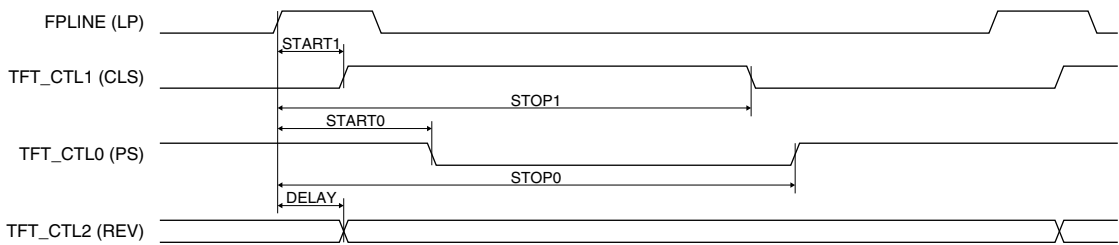
* Example timing for a 320 × 240 panel

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)
 2. t_{2typ} = $VDPSCNT[9:0]$ (Lines)
 3. t_{3typ} = $VDCNT[9:0] + 1$ (Lines)

31 ELECTRICAL CHARACTERISTICS

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



* When FPLINE_ST[9:0] = 0x0

START1 = CTL1ST[7:0] (Ts)

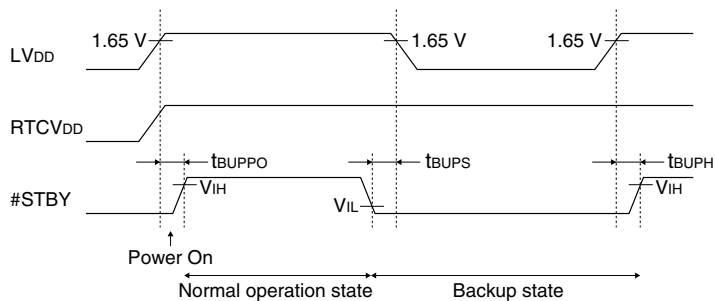
STOP1 = CTL1STP[7:0] + 1 (Ts)

START0 = CTL0ST[7:0] (Ts)

STOP0 = CTL0STP[7:0] + 1 (Ts)

DELAY = CTL2DLY[7:0] (Ts)

31.8.6 #STBY AC Characteristics



Unless otherwise specified: LVDD = RTCVDD = PLLVDD = 1.65 to 1.95V, HVDD = AVDD = 2.7 to 3.6V, VSS = 0V

Item	Symbol	Min.	Typ.	Max.	Unit
Power shutdown time when entering backup status *1	tbUPS	1	–	–	ms
Power stabilization time when exiting from backup status *2	tbUPH	1	–	–	ms

*1) After setting the #STBY pin to low to place the IC into backup state, keep LVDD at 1.65 V or more for at least tbUPS.

*2) Before setting the #STBY pin to high to cancel backup state, wait at least tbUPH after LVDD goes 1.65 V or higher.

31.9 USB DC and AC Characteristics

Input levels

Unless otherwise specified: LVDD = 1.65 to 1.95V, HVDD = 3.0 to 3.6V, VSS = 0V, Ta = 0 to 70°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
VBUS input *1	V _{BUS}		4.40	–	5.25	V
High (driven) *2	V _{IH}		2.0	–	–	V
High (floating) *2	V _{IHZ}		2.7	–	3.6	V
Low *2	V _{IL}		–	–	0.8	V
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

*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: LVDD = 1.65 to 1.95V, HVDD = 3.0 to 3.6V, VSS = 0V, Ta = 0 to 70°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Low *3	V _{OL}		0	–	0.3	V
High (driven) *3	V _{OH}		2.8	–	3.6	V
Output signal crossover voltage *4	V _{CRS}		1.3	–	2.0	V

*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: LV_{DD} = 1.65 to 1.95V, HV_{DD} = 3.0 to 3.6V, V_{SS} = 0V, Ta = 0 to 70°C

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

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

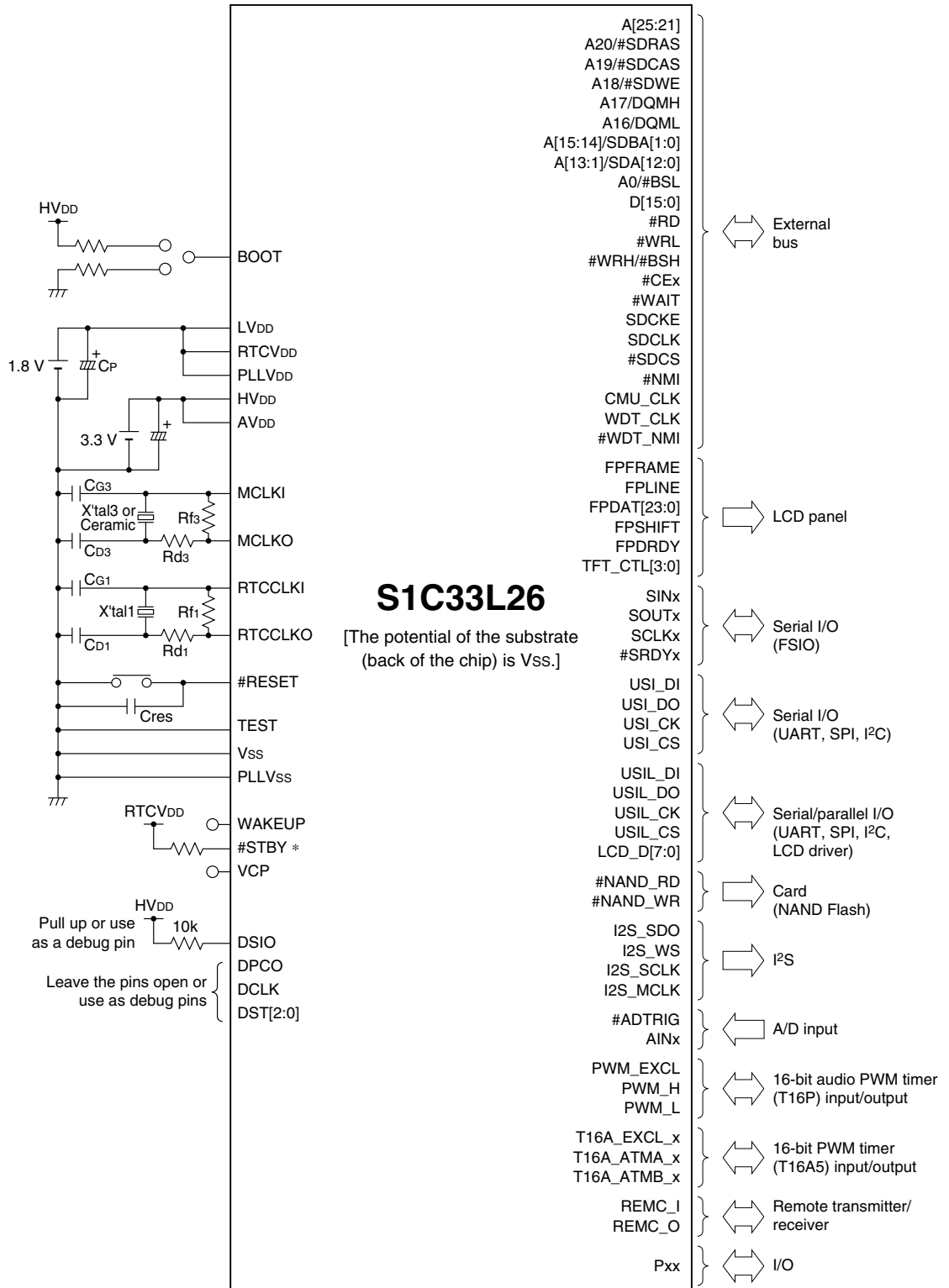
Driver characteristics

Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, HV_{DD} = 3.0 to 3.6V, V_{SS} = 0V, Ta = 0 to 70°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Rise time *6	T _{FR}		4	–	20	ns
Fall time *6	T _{FF}		4	–	20	ns
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	–	–	kΩ
VBUS resistor ratio		R1 : R2	1 : 2 (nominal)			–

*6) Refer to Figures 7-8 and 7-9 in the USB2.0 Specification for the conditions.

32 Basic External Connection Diagram



* The #STBY pin should be fixed at the RTCVDd level if it is not used for power control.

Recommended values for external parts

External parts for the OSC1 oscillator circuit

Symbol	Resonator	Recommended manufacturer	Frequency [Hz]	Product number	Recommended values				Recommended operating condition Temperature range [°C]
					C _{D1} [pF]	C _{G1} [pF]	R _{f1} [Ω]	R _{d1} [Ω]	
X'tal1	Crystal	Epson Toyocom Corporation	32.768k	*1	–	–	–	–	–
		(Reference values)	32.768k	–	10	10	10M	0	-40 to 85

*1 Please contact the recommended manufacturer.

External parts for the OSC3 oscillator circuit

Symbol	Resonator	Recommended manufacturer	Frequency [Hz]	Product number	Recommended values *2				Recommended operating condition Temperature range [°C]
					C _{D3} [pF]	C _{G3} [pF]	R _{f3} [Ω]	R _{d3} [Ω]	
X'tal3	Crystal	Epson Toyocom Corporation	1M to 48M	*1	–	–	–	–	–
		(Reference values)	1M to 48M	–	15	15	1M	0	-40 to 85
Ceramic	Ceramic	Murata Manufacturing Co., Ltd.	1M	CSBFB1M00J58-R1 [SMD]	330	330	1M	680	-20 to 80
			1M	CSBLA1M00J58-B0 [leaded]	330	330	1M	680	-20 to 80
			4M	CSTCR4M00G55-R0 [leaded]	(39)	(39)	1M	470	-20 to 80
			4M	CSTLS4M00G56-B0 [leaded]	(47)	(47)	1M	330	-20 to 80
			10M	CSTCE10M0G55-R0 [SMD]	(33)	(33)	1M	220	-20 to 80
			10M	CSTLS10M0G56-B0 [leaded]	(47)	(47)	1M	220	-20 to 80
			20M	CSTCE20M0V53-R0 [SMD]	(15)	(15)	1M	0	-20 to 80
			20M	CSTCG20M0V53-R0 [small SMD]	(15)	(15)	1M	0	-20 to 80
			40M	CSTCW40M0X51-R0 [SMD]	(6)	(6)	1M	0	-20 to 80
			48M	CSTCZ48MOX12R*-R [SMD]	(10)	(10)	1M	0	-30 to 85

*1 Please contact the recommended manufacturer.

*2 The C_{D3} and C_{G3} values enclosed with () are the built-in capacitances of the resonator.

Other

Symbol	Name	Recommended value
CP	Capacitor for power supply	3.3 μF
Cres	Capacitor for #RESET pin	0.47 μF

- Notes:**
- The values in the above table are shown only for reference and not guaranteed.
 - Crystal and ceramic resonators are extremely sensitive to influence of external components and printed-circuit boards. Before using a resonator, please contact the manufacturer for further information on conditions of use.

Appendix A List of I/O Registers

Peripheral	Address	Register name		Function
MISC registers (MISC) (8-bit device)	0x300010	MISC_RTCWT	RTC Wait Control Register	Configure RTC access cycles
	0x300012	MISC_USB	USB Configuration Register	Enable USB interrupt and configure access cycle
	0x300014	MISC_RAMWT	Internal RAM Wait Control Register	Configure IRAM and IVRAM access cycles
	0x300016	MISC_BOOT	Boot Register	Indicate/set boot conditions
	0x300018	MISC_RAM_LOC	RAM Location Select Register	Select area for locating RAM
	0x300020	MISC_PROTECT	Misc Protect Register	Enable/disable Misc register write protection
Clock management unit (CMU) (8-bit device)	0x300100	CMU_OSCSEL	Clock Source Select Register	Select system clock source
	0x300101	CMU_OSCCTL	Oscillation Control Register	Control oscillators
	0x300103	CMU_LCLKDIV	LCDC Clock Division Ratio Select Register	Set LCLK frequency
	0x300104	CMU_CLKCTL	Clock Control Register	Control clock supply to peripheral/bus modules
	0x300105	CMU_SYSCCLKDIV	System Clock Division Ratio Select Register	Set system clock frequency
	0x300106	CMU_CMUCLK	CMU_CLK Select Register	Select CMU_CLK output clock
	0x300107	CMU_PLLINDIV	PLL Input Clock Division Ratio Select Register	Set PLL input clock frequency
	0x300108	CMU_PLLCTL0	PLL Control Register 0	Set PLL multiplication rate and enable PLL
	0x300109	CMU_PLLCTL1	PLL Control Register 1	Set PLL parameters
	0x30010a	CMU_PLLCTL2	PLL Control Register 2	
	0x30010c	CMU_SSCG0	SSCG Macro Control Register 0	Enable SSCG
	0x30010d	CMU_SSCG1	SSCG Macro Control Register 1	Set SSCG parameters
	0x300110	CMU_PROTECT	CMU Write Protect Register	Enable/disable CMU register write protection
	Interrupt controller (ITC) (8-bit device)	0x300210	ITC_FPT03_LV	FPT0–3 Interrupt Level Register
0x300211		ITC_FPT47_LV	FPT4–7 Interrupt Level Register	Set FPT4–7 interrupt levels
0x300212		ITC_FPT8B_LV	FPT8–B Interrupt Level Register	Set FPT8–B interrupt levels
0x300213		ITC_FPTCF_LV	FPTC–F Interrupt Level Register	Set FPTC–F interrupt levels
0x300214		ITC_DMA02_LV	DMAC Ch.0 & 2 Interrupt Level Register	Set DMAC Ch.0 and 2 interrupt levels
0x300215		ITC_DMA13_LV	DMAC Ch.1 & 3 Interrupt Level Register	Set DMAC Ch.1 and 3 interrupt levels
0x300216		ITC_DMA46_LV	DMAC Ch.4 & 6 Interrupt Level Register	Set DMAC Ch.4 and 6 interrupt levels
0x300217		ITC_DMA57_LV	DMAC Ch.5 & 7 Interrupt Level Register	Set DMAC Ch.5 and 7 interrupt levels
0x300218		ITC_T16P_LV	T16P Interrupt Level Register	Set T16P interrupt level
0x300219		ITC_T16A0_LV	T16A5 Ch.0 Interrupt Level Register	Set T16A5 Ch.0 interrupt level
0x30021a		ITC_T16A1_LV	T16A5 Ch.1 Interrupt Level Register	Set T16A5 Ch.1 interrupt level
0x30021b		ITC_LCDC_LV	LCDC Interrupt Level Register	Set LCDC interrupt level
0x30021d		ITC_T804_LV	T8 Ch.0 & 4 Interrupt Level Register	Set T8 Ch.0 and 4 interrupt levels
0x30021e		ITC_T815_LV	T8 Ch.1 & 5 Interrupt Level Register	Set T8 Ch.1 and 5 interrupt levels
0x30021f		ITC_T826_LV	T8 Ch.2 & 6 Interrupt Level Register	Set T8 Ch.2 and 6 interrupt levels
0x300220		ITC_T837_LV	T8 Ch.3 & 7 Interrupt Level Register	Set T8 Ch.3 and 7 interrupt levels
0x300221		ITC_USI_LV	USI Interrupt Level Register	Set USI interrupt level
0x300222		ITC_FSIO0_LV	FSIO Ch.0 Interrupt Level Register	Set FSIO Ch.0 interrupt level
0x300223		ITC_ADC10_LV	ADC10 Interrupt Level Register	Set ADC10 interrupt level
0x300224		ITC_RTC_LV	RTC Interrupt Level Register	Set RTC interrupt level
0x300226		ITC_FSIO1_LV	FSIO Ch.1 Interrupt Level Register	Set FSIO Ch.1 interrupt level
0x300227		ITC_USIL_LV	USIL Interrupt Level Register	Set USIL interrupt level
0x300228		ITC_REMC_LV	REMC Interrupt Level Register	Set REMC interrupt level
0x300229		ITC_I2S_LV	I ² S Interrupt Level Register	Set I ² S interrupt level
0x30022a		ITC_GECOM_LV	GE Complete Interrupt Level Register	Set GE complete interrupt level
0x30022b		ITC_GEERR_LV	GE Error Interrupt Level Register	Set GE error interrupt level
0x30022c		ITC_USB_LV	USB Interrupt Level Register	Set USB interrupt level
GPIO & port MUX (8-bit device)	0x300300	GPIO_P0_DAT	P0 Port Data Register	P0 port input/output data
	0x300301	GPIO_P0_IOC	P0 Port I/O Control Register	Control P0 port input/output direction
	0x300302	GPIO_P1_DAT	P1 Port Data Register	P1 port input/output data
	0x300303	GPIO_P1_IOC	P1 Port I/O Control Register	Control P1 port input/output direction
	0x300304	GPIO_P2_DAT	P2 Port Data Register	P2 port input/output data
	0x300305	GPIO_P2_IOC	P2 Port I/O Control Register	Control P2 port input/output direction
	0x300306	GPIO_P3_DAT	P3 Port Data Register	P3 port input/output data
	0x300307	GPIO_P3_IOC	P3 Port I/O Control Register	Control P3 port input/output direction
	0x300308	GPIO_P4_DAT	P4 Port Data Register	P4 port input/output data
	0x300309	GPIO_P4_IOC	P4 Port I/O Control Register	Control P4 port input/output direction
	0x30030a	GPIO_P5_DAT	P5 Port Data Register	P5 port input/output data
	0x30030b	GPIO_P5_IOC	P5 Port I/O Control Register	Control P5 port input/output direction
	0x30030c	GPIO_P6_DAT	P6 Port Data Register	P6 port input/output data
	0x30030d	GPIO_P6_IOC	P6 Port I/O Control Register	Control P6 port input/output direction
	0x30030e	GPIO_P7_DAT	P7 Port Data Register	P7 port input data
	0x300310	GPIO_P8_DAT	P8 Port Data Register	P8 port input/output data
	0x300311	GPIO_P8_IOC	P8 Port I/O Control Register	Control P8 port input/output direction

APPENDIX A LIST OF I/O REGISTERS

Peripheral	Address	Register name	Function	
GPIO & port MUX (8-bit device)	0x300312	GPIO_P9_DAT	P9 Port Data Register	P9 port input/output data
	0x300313	GPIO_P9_IOC	P9 Port I/O Control Register	Control P9 port input/output direction
	0x300314	GPIO_PA_DAT	PA Port Data Register	PA port input/output data
	0x300315	GPIO_PA_IOC	PA Port I/O Control Register	Control PA port input/output direction
	0x300316	GPIO_PB_DAT	PB Port Data Register	PB port input/output data
	0x300317	GPIO_PB_IOC	PB Port I/O Control Register	Control PB port input/output direction
	0x300318	GPIO_PC_DAT	PC Port Data Register	PC port input/output data
	0x300319	GPIO_PC_IOC	PC Port I/O Control Register	Control PC port input/output direction
	0x300320	GPIO_BUS_DRV	Bus Drive Control Register	Set external data and address bus signals to low
	0x300321	GPIO_P0_PUP	P0 Port Pull-up Control Register	Enable/disable P0 port pull-up resistors
	0x300322	GPIO_P1_PUP	P1 Port Pull-up Control Register	Enable/disable P1 port pull-up resistors
	0x300323	GPIO_P2_PUP	P2 Port Pull-up Control Register	Enable/disable P2 port pull-up resistors
	0x300324	GPIO_P3_PUP	P3 Port Pull-up Control Register	Enable/disable P3 port pull-up resistors
	0x300325	GPIO_P4_PUP	P4 Port Pull-up Control Register	Enable/disable P4 port pull-up resistors
	0x300326	GPIO_P5_PUP	P5 Port Pull-up Control Register	Enable/disable P5 port pull-up resistors
	0x300327	GPIO_P6_PUP	P6 Port Pull-up Control Register	Enable/disable P6 port pull-up resistors
	0x300328	GPIO_P7_PUP	P7 Port Pull-up Control Register	Enable/disable P7 port pull-up resistors
	0x300329	GPIO_P8_PUP	P8 Port Pull-up Control Register	Enable/disable P8 port pull-up resistors
	0x30032a	GPIO_P9_PUP	P9 Port Pull-up Control Register	Enable/disable P9 port pull-up resistors
	0x30032b	GPIO_PA_PUP	PA Port Pull-up Control Register	Enable/disable PA port pull-up resistors
	0x30032c	GPIO_PB_PUP	PB Port Pull-up Control Register	Enable/disable PB port pull-up resistors
	0x300330	GPIO_FPT03_SEL	FPT0–3 Interrupt Port Select Register	Select ports used for FPT0–3 interrupts
	0x300331	GPIO_FPT47_SEL	FPT4–7 Interrupt Port Select Register	Select ports used for FPT4–7 interrupts
	0x300332	GPIO_FPT8B_SEL	FPT8–B Interrupt Port Select Register	Select ports used for FPT8–B interrupts
	0x300333	GPIO_FPTCF_SEL	FPTC–F Interrupt Port Select Register	Select ports used for FPTC–F interrupts
	0x300334	GPIO_FPT03_POL	FPT0–3 Interrupt Polarity Select Register	Select input signal polarity for FPT0–3 interrupts
	0x300335	GPIO_FPT47_POL	FPT4–7 Interrupt Polarity Select Register	Select input signal polarity for FPT4–7 interrupts
	0x300336	GPIO_FPT8B_POL	FPT8–B Interrupt Polarity Select Register	Select input signal polarity for FPT8–B interrupts
	0x300337	GPIO_FPTCF_POL	FPTC–F Interrupt Polarity Select Register	Select input signal polarity for FPTC–F interrupts
	0x300338	GPIO_FPT03_MOD	FPT0–3 Interrupt Mode Select Register	Select edge/level mode for FPT0–3 interrupts
	0x300339	GPIO_FPT47_MOD	FPT4–7 Interrupt Mode Select Register	Select edge/level mode for FPT4–7 interrupts
	0x30033a	GPIO_FPT8B_MOD	FPT8–B Interrupt Mode Select Register	Select edge/level mode for FPT8–B interrupts
	0x30033b	GPIO_FPTCF_MOD	FPTC–F Interrupt Mode Select Register	Select edge/level mode for FPTC–F interrupts
	0x30033c	GPIO_FPT03_MSK	FPT0–3 Interrupt Mask Register	Enable/disable FPT0–3 interrupts
	0x30033d	GPIO_FPT47_MSK	FPT4–7 Interrupt Mask Register	Enable/disable FPT4–7 interrupts
	0x30033e	GPIO_FPT8B_MSK	FPT8–B Interrupt Mask Register	Enable/disable FPT8–B interrupts
	0x30033f	GPIO_FPTCF_MSK	FPTC–F Interrupt Mask Register	Enable/disable FPTC–F interrupts
	0x300340	GPIO_FPT03_FLG	FPT0–3 Interrupt Flag Register	Indicate FPT0–3 interrupt cause status
	0x300341	GPIO_FPT47_FLG	FPT4–7 Interrupt Flag Register	Indicate FPT4–7 interrupt cause status
	0x300342	GPIO_FPT8B_FLG	FPT8–B Interrupt Flag Register	Indicate FPT8–B interrupt cause status
	0x300343	GPIO_FPTCF_FLG	FPTC–F Interrupt Flag Register	Indicate FPTC–F interrupt cause status
	0x300344	GPIO_FPT01_CHAT	FPT0–1 Interrupt Chattering Filter Control Register	Control FPT0–1 chattering filter
	0x300345	GPIO_FPT23_CHAT	FPT2–3 Interrupt Chattering Filter Control Register	Control FPT2–3 chattering filter
	0x300346	GPIO_FPT45_CHAT	FPT4–5 Interrupt Chattering Filter Control Register	Control FPT4–5 chattering filter
	0x300347	GPIO_FPT67_CHAT	FPT6–7 Interrupt Chattering Filter Control Register	Control FPT6–7 chattering filter
	0x300348	GPIO_FPT89_CHAT	FPT8–9 Interrupt Chattering Filter Control Register	Control FPT8–9 chattering filter
	0x300349	GPIO_FPTAB_CHAT	FPTA–B Interrupt Chattering Filter Control Register	Control FPTA–B chattering filter
	0x30034a	GPIO_FPTCD_CHAT	FPTC–D Interrupt Chattering Filter Control Register	Control FPTC–D chattering filter
	0x30034b	GPIO_FPTEF_CHAT	FPT E–F Interrupt Chattering Filter Control Register	Control FPTE–F chattering filter
	0x30034c	GPIO_DMA	Port DMA Trigger Source Select Register	Select port DMA trigger source
	0x300800	PMUX_P0_03	P0[3:0] Port Function Select Register	Select P0[3:0] port functions
	0x300801	PMUX_P0_47	P0[7:4] Port Function Select Register	Select P0[7:4] port functions
0x300802	PMUX_P1_03	P1[3:0] Port Function Select Register	Select P1[3:0] port functions	
0x300803	PMUX_P1_47	P1[7:4] Port Function Select Register	Select P1[7:4] port functions	
0x300804	PMUX_P2_01	P2[1:0] Port Function Select Register	Select P2[1:0] port functions	
0x300806	PMUX_P3_03	P3[3:0] Port Function Select Register	Select P3[3:0] port functions	
0x300807	PMUX_P3_46	P3[6:4] Port Function Select Register	Select P3[6:4] port functions	
0x300808	PMUX_P4_02	P4[2:0] Port Function Select Register	Select P4[2:0] port functions	
0x30080a	PMUX_P5_03	P5[3:0] Port Function Select Register	Select P5[3:0] port functions	
0x30080b	PMUX_P5_46	P5[6:4] Port Function Select Register	Select P5[6:4] port functions	
0x30080c	PMUX_P6_0	P6 Port Function Select Register	Select P6 port functions	

APPENDIX A LIST OF I/O REGISTERS

Peripheral	Address	Register name	Function	
GPIO & port MUX (8-bit device)	0x30080e	PMUX_P7_03	P7[3:0] Port Function Select Register	Select P7[3:0] port functions
	0x30080f	PMUX_P7_45	P7[5:4] Port Function Select Register	Select P7[5:4] port functions
	0x300810	PMUX_P8_03	P8[3:0] Port Function Select Register	Select P8[3:0] port functions
	0x300812	PMUX_P9_03	P9[3:0] Port Function Select Register	Select P9[3:0] port functions
	0x300813	PMUX_P9_47	P9[7:4] Port Function Select Register	Select P9[7:4] port functions
	0x300814	PMUX_PA_03	PA[3:0] Port Function Select Register	Select PA[3:0] port functions
	0x300815	PMUX_PA_46	PA[6:4] Port Function Select Register	Select PA[6:4] port functions
	0x300816	PMUX_PB_03	PB[3:0] Port Function Select Register	Select PB[3:0] port functions
	0x300817	PMUX_PB_47	PB[7:4] Port Function Select Register	Select PB[7:4] port functions
	0x300818	PMUX_PC_03	PC[3:0] Port Function Select Register	Select PC[3:0] port functions
	0x300819	PMUX_PC_47	PC[7:4] Port Function Select Register	Select PC[7:4] port functions
0x30083e	GPIO_FILTER	Port Noise Filter Control Register	Enable/disable port input noise filter	
0x30083f	GPIO_PROTECT	GPIO/PMUX Write Protect Register	Enable/disable write protection for PMUX, GPIO_FILTER, GPIO_BUS_DRV, and GPIO_Px_PUP registers	
USI (8-bit device)	0x300400	USI_GCFG	USI Global Configuration Register	Set interface and MSB/LSB modes
	0x300401	USI_TD	USI Transmit Data Buffer Register	Transmit data buffer
	0x300402	USI_RD	USI Receive Data Buffer Register	Receive data buffer
	0x300440	USI_UCFG	USI UART Mode Configuration Register	Set UART transfer conditions
	0x300441	USI_UIE	USI UART Mode Interrupt Enable Register	Enable/disable UART interrupts
	0x300442	USI_UIF	USI UART Mode Interrupt Flag Register	Indicate UART interrupt cause status
	0x300450	USI_SCFG	USI SPI Master/Slave Mode Configuration Register	Set SPI transfer conditions
	0x300451	USI_SIE	USI SPI Master/Slave Mode Interrupt Enable Register	Enable/disable SPI interrupts
	0x300452	USI_SIF	USI SPI Master/Slave Mode Interrupt Flag Register	Indicate SPI interrupt cause status
	0x300460	USI_IMTG	USI I ² C Master Mode Trigger Register	Start I ² C master operations
	0x300461	USI_IMIE	USI I ² C Master Mode Interrupt Enable Register	Enable/disable I ² C master interrupts
	0x300462	USI_IMIF	USI I ² C Master Mode Interrupt Flag Register	Indicate I ² C master interrupt cause status
	0x300470	USI_ISTG	USI I ² C Slave Mode Trigger Register	Start I ² C slave operations
	0x300471	USI_ISIE	USI I ² C Slave Mode Interrupt Enable Register	Enable/disable I ² C slave interrupts
0x300472	USI_ISIF	USI I ² C Slave Mode Interrupt Flag Register	Indicate I ² C slave interrupt cause status	
USIL (8-bit device)	0x300600	USIL_GCFG	USIL Global Configuration Register	Set interface and MSB/LSB modes
	0x300601	USIL_TD	USIL Transmit Data Buffer Register	Transmit data buffer
	0x300602	USIL_RD	USIL Receive Data Buffer Register	Receive data buffer
	0x300640	USIL_UCFG	USIL UART Mode Configuration Register	Set UART transfer conditions
	0x300641	USIL_UIE	USIL UART Mode Interrupt Enable Register	Enable/disable UART interrupts
	0x300642	USIL_UIF	USIL UART Mode Interrupt Flag Register	Indicate UART interrupt cause status
	0x300650	USIL_SCFG	USIL SPI Master/Slave Mode Configuration Register	Set SPI transfer conditions
	0x300651	USIL_SIE	USIL SPI Master/Slave Mode Interrupt Enable Register	Enable/disable SPI interrupts
	0x300652	USIL_SIF	USIL SPI Master/Slave Mode Interrupt Flag Register	Indicate SPI interrupt cause status
	0x300660	USIL_IMTG	USIL I ² C Master Mode Trigger Register	Start I ² C master operations
	0x300661	USIL_IMIE	USIL I ² C Master Mode Interrupt Enable Register	Enable/disable I ² C master interrupts
	0x300662	USIL_IMIF	USIL I ² C Master Mode Interrupt Flag Register	Indicate I ² C master interrupt cause status
	0x300670	USIL_ISTG	USIL I ² C Slave Mode Trigger Register	Start I ² C slave operations
	0x300671	USIL_ISIE	USIL I ² C Slave Mode Interrupt Enable Register	Enable/disable I ² C slave interrupts
	0x300672	USIL_ISIF	USIL I ² C Slave Mode Interrupt Flag Register	Indicate I ² C slave interrupt cause status
	0x300680	USIL_LSCFG	USIL LCD SPI Mode Configuration Register	Set LCD SPI transfer conditions
	0x300681	USIL_LSIE	USIL LCD SPI Mode Interrupt Enable Register	Enable/disable LCD SPI interrupts
	0x300682	USIL_LSIF	USIL LCD SPI Mode Interrupt Flag Register	Indicate LCD SPI interrupt cause status
	0x30068f	USIL_LSDCFG	USIL LCD SPI Mode Data Configuration Register	Select display data format
	0x300690	USIL_LPCFG	USIL LCD Parallel I/F Mode Configuration Register	Set LCD parallel interface conditions
	0x300691	USIL_LPIE	USIL LCD Parallel I/F Mode Interrupt Enable Register	Enable/disable LCD parallel interface interrupts
0x300692	USIL_LPIF	USIL LCD Parallel I/F Mode Interrupt Flag Register	Indicate LCD parallel interface interrupt cause status	
0x30069f	USIL_LPAC	USIL LCD Parallel I/F Mode Access Timing Register	Set LCD parallel interface access timing parameters	
FSIO Ch.0 (8-bit device)	0x300700	FSIO_TXD0	FSIO Ch.0 Transmit Data Register	Transmit data
	0x300701	FSIO_RXD0	FSIO Ch.0 Receive Data Register	Receive data
	0x300702	FSIO_STATUS0	FSIO Ch.0 Status Register	Indicate transfer/error statuses
	0x300703	FSIO_CTL0	FSIO Ch.0 Control Register	Set transfer mode and control data transfer
	0x300704	FSIO_IRDA0	FSIO Ch.0 IrDA Register	Set IrDA conditions
	0x300705	FSIO_BRTRUN0	FSIO Ch.0 Baud-rate Timer Control Register	Control baud-rate timer

APPENDIX A LIST OF I/O REGISTERS

Peripheral	Address		Register name	Function
FSIO Ch.0 (8-bit device)	0x300706	FSIO_BRTRDL0	FSIO Ch.0 Baud-rate Timer Reload Data L Register	Baud-rate timer initial count data
	0x300707	FSIO_BRTRDH0	FSIO Ch.0 Baud-rate Timer Reload Data H Register	
	0x300708	FSIO_BRTCDL0	FSIO Ch.0 Baud-rate Timer Count Data L Register	Baud-rate timer count data
	0x300709	FSIO_BRTCDH0	FSIO Ch.0 Baud-rate Timer Count Data H Register	
	0x30070a	FSIO_INTF0	FSIO Ch.0 Interrupt Flag Register	Indicate FSIO interrupt cause status
	0x30070b	FSIO_INTE0	FSIO Ch.0 Interrupt Enable Register	Enable/disable FSIO interrupts
0x30070f	FSIO_ADV0	FSIO Ch.0 STD/ADV Mode Select Register	Select standard/advanced mode	
FSIO Ch.1 (8-bit device)	0x300710	FSIO_TXD1	FSIO Ch.1 Transmit Data Register	Transmit data
	0x300711	FSIO_RXD1	FSIO Ch.1 Receive Data Register	Receive data
	0x300712	FSIO_STATUS1	FSIO Ch.1 Status Register	Indicate transfer/error statuses
	0x300713	FSIO_CTL1	FSIO Ch.1 Control Register	Set transfer mode and control data transfer
	0x300714	FSIO_IRDA1	FSIO Ch.1 IrDA Register	Set IrDA conditions
	0x300715	FSIO_BRTRUN1	FSIO Ch.1 Baud-rate Timer Control Register	Control baud-rate timer
	0x300716	FSIO_BRTRDL1	FSIO Ch.1 Baud-rate Timer Reload Data L Register	Baud-rate timer initial count data
	0x300717	FSIO_BRTRDH1	FSIO Ch.1 Baud-rate Timer Reload Data H Register	
	0x300718	FSIO_BRTCDL1	FSIO Ch.1 Baud-rate Timer Count Data L Register	Baud-rate timer count data
	0x300719	FSIO_BRTCDH1	FSIO Ch.1 Baud-rate Timer Count Data H Register	
	0x30071a	FSIO_INTF1	FSIO Ch.1 Interrupt Flag Register	Indicate FSIO interrupt cause status
	0x30071b	FSIO_INTE1	FSIO Ch.1 Interrupt Enable Register	Enable/disable FSIO interrupts
	0x30071f	FSIO_ADV1	FSIO Ch.1 STD/ADV Mode Select Register	Select standard/advanced mode
	Real-Time Clock (RTC) (8-bit device)	0x300a00	RTC_INTSTAT	RTC Interrupt Status Register
0x300a01		RTC_INTMODE	RTC Interrupt Mode Register	Set RTC interrupt modes
0x300a02		RTC_CNTL0	RTC Control 0 Register	Control RTC
0x300a03		RTC_CNTL1	RTC Control 1 Register	
0x300a04		RTC_SEC	RTC Second Register	Second counter data
0x300a05		RTC_MIN	RTC Minute Register	Minute counter data
0x300a06		RTC_HOUR	RTC Hour Register	Hour counter data
0x300a07		RTC_DAY	RTC Day Register	Day counter data
0x300a08		RTC_MONTH	RTC Month Register	Month counter data
0x300a09		RTC_YEAR	RTC Year Register	Year counter data
0x300a0a		RTC_WEEK	RTC Days of Week Register	Days of week counter data
0x300a0f	RTC_WAKEUP	RTC Wakeup Configuration Register	Set RTC wakeup conditions	
BBRAM (8-bit device)	0x300b00	BBRAM_0	BBRAM byte 0	BBRAM
	0x300b0f	BBRAM_15	BBRAM byte 15	
USB function controller (USB) (8-bit device)	0x300c00	MainIntStat	Main Interrupt Status Register	Indicate main interrupt status
	0x300c01	SIE_IntStat	SIE Interrupt Status Register	Indicate SIE interrupt status
	0x300c02	EPrIntStat	EPr Interrupt Status Register	Indicate EPr interrupt status
	0x300c03	DMA_IntStat	DMA Interrupt Status Register	Indicate DMA interrupt status
	0x300c04	FIFO_IntStat	FIFO Interrupt Status Register	Indicate FIFO interrupt status
	0x300c07	EP0IntStat	EP0 Interrupt Status Register	Indicate EP0 interrupt status
	0x300c08	EPaIntStat	EPa Interrupt Status Register	Indicate EPa interrupt status
	0x300c09	EPbIntStat	EPb Interrupt Status Register	Indicate EPb interrupt status
	0x300c0a	EPcIntStat	EPc Interrupt Status Register	Indicate EPc interrupt status
	0x300c0b	EPdIntStat	EPd Interrupt Status Register	Indicate EPd interrupt status
	0x300c10	MainIntEnb	Main Interrupt Enable Register	Enable main interrupts
	0x300c11	SIE_IntEnb	SIE Interrupt Enable Register	Enable SIE interrupts
	0x300c12	EPrIntEnb	EPr Interrupt Enable Register	Enable EPr interrupts
	0x300c13	DMA_IntEnb	DMA Interrupt Enable Register	Enable DMA interrupts
	0x300c14	FIFO_IntEnb	FIFO Interrupt Enable Register	Enable FIFO interrupts
	0x300c17	EP0IntEnb	EP0 Interrupt Enable Register	Enable EP0 interrupts
	0x300c18	EPaIntEnb	EPa Interrupt Enable Register	Enable EPa interrupts
	0x300c19	EPbIntEnb	EPb Interrupt Enable Register	Enable EPb interrupts
	0x300c1a	EPcIntEnb	EPc Interrupt Enable Register	Enable EPc interrupts
	0x300c1b	EPdIntEnb	EPd Interrupt Enable Register	Enable EPd interrupts
	0x300c20	RevisionNum	Revision Number Register	Indicate revision number of USB controller
	0x300c21	USB_Control	USB Control Register	Control USB conditions
	0x300c22	USB_Status	USB Status Register	Indicate USB status
	0x300c23	XcvrControl	Xcvr Control Register	Control transceiver macro
0x300c24	USB_Test	USB Test Register	Set USB test mode	

Peripheral	Address	Register name		Function
USB function controller (USB) (8-bit device)	0x300c25	EPnControl	Endpoint Control Register	Clear all FIFOs and set NAK/STALL bits
	0x300c26	EPnFIFO_Clr	EPn FIFO Clear Register	Clear each FIFO
	0x300c2e	FrameNumber_H	Frame Number High Register	Frame number
	0x300c2f	FrameNumber_L	Frame Number Low Register	
	0x300c30	EP0Setup_0	EP0 Setup 0 Register	EP0 setup data (BmRequestType)
	0x300c31	EP0Setup_1	EP0 Setup 1 Register	EP0 setup data (BRequest)
	0x300c32	EP0Setup_2	EP0 Setup 2 Register	EP0 setup data (low-order Wvalue bits)
	0x300c33	EP0Setup_3	EP0 Setup 3 Register	EP0 setup data (high-order Wvalue bits)
	0x300c34	EP0Setup_4	EP0 Setup 4 Register	EP0 setup data (low-order WIndex bits)
	0x300c35	EP0Setup_5	EP0 Setup 5 Register	EP0 setup data (high-order WIndex bits)
	0x300c36	EP0Setup_6	EP0 Setup 6 Register	EP0 setup data (low-order WLength bits)
	0x300c37	EP0Setup_7	EP0 Setup 7 Register	EP0 setup data (high-order WLength bits)
	0x300c38	USB_Address	USB Address Register	Set USB address
	0x300c39	EP0Control	EP0 Control Register	Set up EP0
	0x300c3a	EP0ControlIN	EP0 Control In Register	Set EP0 IN transaction conditions
	0x300c3b	EP0ControlOUT	EP0 Control Out Register	Set EP0 OUT transaction conditions
	0x300c3f	EP0MaxSize	EP0 Max Packet Size Register	Set the EP0 max packet size
	0x300c40	EPaControl	EPa Control Register	Set up EPa
	0x300c41	EPbControl	EPb Control Register	Set up EPb
	0x300c42	EPcControl	EPc Control Register	Set up EPc
	0x300c43	EPdControl	EPd Control Register	Set up EPd
	0x300c50	EPaMaxSize_H	EPa Max Packet Size High Register	Set EPa max packet size
	0x300c51	EPaMaxSize_L	EPa Max Packet Size Low Register	
	0x300c52	EPaConfig_0	EPa Configuration 0 Register	Configure EPa
	0x300c53	EPaConfig_1	EPa Configuration 1 Register	
	0x300c54	EPbMaxSize_H	EPb Max Packet Size High Register	Set EPb max packet size
	0x300c55	EPbMaxSize_L	EPb Max Packet Size Low Register	
	0x300c56	EPbConfig_0	EPb Configuration 0 Register	Configure EPb
	0x300c57	EPbConfig_1	EPb Configuration 1 Register	
	0x300c58	EPcMaxSize_H	EPc Max Packet Size High Register	Set EPc max packet size
	0x300c59	EPcMaxSize_L	EPc Max Packet Size Low Register	
	0x300c5a	EPcConfig_0	EPc Configuration 0 Register	Configure EPc
	0x300c5b	EPcConfig_1	EPc Configuration 1 Register	
	0x300c5c	EPdMaxSize_H	EPd Max Packet Size High Register	Set EPd max packet size
	0x300c5d	EPdMaxSize_L	EPd Max Packet Size Low Register	
	0x300c5e	EPdConfig_0	EPd Configuration 0 Register	Configure EPd
	0x300c5f	EPdConfig_1	EPd Configuration 1 Register	
	0x300c70	EPaStartAdrs_H	EPa FIFO Start Address High Register	Set FIFO start address for EPa
	0x300c71	EPaStartAdrs_L	EPa FIFO Start Address Low Register	
	0x300c72	EPbStartAdrs_H	EPb FIFO Start Address High Register	Set FIFO start address for EPb
	0x300c73	EPbStartAdrs_L	EPb FIFO Start Address Low Register	
	0x300c74	EPcStartAdrs_H	EPc FIFO Start Address High Register	Set FIFO start address for EPc
	0x300c75	EPcStartAdrs_L	EPc FIFO Start Address Low Register	
	0x300c76	EPdStartAdrs_H	EPd FIFO Start Address High Register	Set FIFO start address for EPd
	0x300c77	EPdStartAdrs_L	EPd FIFO Start Address Low Register	
	0x300c80	CPU_JoinRd	CPU Join FIFO Read Register	Set up FIFO data read conditions
	0x300c81	CPU_JoinWr	CPU Join FIFO Write Register	Set up FIFO data write conditions
0x300c82	EnEPnFIFO_Access	EPn FIFO Access Enable Register	Enable CPU_JoinRd and CPU_JoinWr registers	
0x300c83	EPnFIFOforCPU	EPn FIFO for CPU Register	EPn FIFO for accessing by CPU	
0x300c84	EPnRdRemain_H	EPn FIFO Read Remain High Register	Indicate remained data quantity in FIFO	
0x300c85	EPnRdRemain_L	EPn FIFO Read Remain Low Register		
0x300c86	EPnWrRemain_H	EPn FIFO Write High Register	Indicate free space capacity in FIFO	
0x300c87	EPnWrRemain_L	EPn FIFO Write Low Register		
0x300c88	DescAdrs_H	Descriptor Address High Register	Specify FIFO start address for the descriptor reply function	
0x300c89	DescAdrs_L	Descriptor Address Low Register		
0x300c8a	DescSize_H	Descriptor Size High Register	Specify number of data for the descriptor reply function	
0x300c8b	DescSize_L	Descriptor Size Low Register		
0x300c8f	DescDoor	Descriptor Door Register	Read/write descriptors	
0x300c90	DMA_FIFO_Control	DMA FIFO Control Register	Control DMA FIFO	
0x300c91	DMA_Join	DMA Join FIFO	Enable endpoint to perform DMA transfer	
0x300c92	DMA_Control	DMA Control Register	Control DMA transfer and indicate DMA status	
0x300c94	DMA_Config_0	DMA Configuration 0 Register	Configure DMA interface signals	
0x300c95	DMA_Config_1	DMA Configuration 1 Register	Set DMA interface operating modes	
0x300c97	DMA_Latency	DMA Latency Register	Set data transfer latency	
0x300c98	DMA_Remain_H	DMA FIFO Remain High Register	Indicate remained data quantity in FIFO or free space capacity in FIFO	
0x300c99	DMA_Remain_L	DMA FIFO Remain Low Register		

APPENDIX A LIST OF I/O REGISTERS

Peripheral	Address		Register name	Function
USB function controller (USB) (8-bit device)	0x300c9c	DMA_Count_HH	DMA Transfer Byte Counter High/High Register	Specify/indicate data length in DMA transfer
	0x300c9d	DMA_Count_HL	DMA Transfer Byte Counter High/Low Register	
	0x300c9e	DMA_Count_LH	DMA Transfer Byte Counter Low/High Register	
	0x300c9f	DMA_Count_LL	DMA Transfer Byte Counter Low/Low Register	
Prescaler (PSC) (8-bit device)	0x300e00	PSC_CTL	PSC Control Register	Control prescaler
Watchdog timer (WDT) (16-bit device)	0x301000	WD_PROTECT	WDT Write Protect Register	Enable WDT register write protection
	0x301002	WD_EN	WDT Enable and Setup Register	Configure and start watchdog timer
	0x301004	WD_CMP_L	WDT Comparison Data L Register	Comparison data
	0x301006	WD_CMP_H	WDT Comparison Data H Register	
	0x301008	WD_CNT_L	WDT Count Data L Register	Watchdog timer counter data
	0x30100a	WD_CNT_H	WDT Count Data H Register	
8-bit timer (T8) Ch.0 (with fine mode) (16-bit device)	0x301100	T8_CLK0	T8 Ch.0 Input Clock Select Register	Select prescaler output clock
	0x301102	T8_TR0	T8 Ch.0 Reload Data Register	Set reload data
	0x301104	T8_TC0	T8 Ch.0 Counter Data Register	Counter data
	0x301106	T8_CTL0	T8 Ch.0 Control Register	Set timer mode and start/stop timer
	0x301108	T8_INT0	T8 Ch.0 Interrupt Control Register	Control interrupt
8-bit timer (T8) Ch.1 (with fine mode) (16-bit device)	0x301110	T8_CLK1	T8 Ch.1 Input Clock Select Register	Select prescaler output clock
	0x301112	T8_TR1	T8 Ch.1 Reload Data Register	Set reload data
	0x301114	T8_TC1	T8 Ch.1 Counter Data Register	Counter data
	0x301116	T8_CTL1	T8 Ch.1 Control Register	Set timer mode and start/stop timer
	0x301118	T8_INT1	T8 Ch.1 Interrupt Control Register	Control interrupt
8-bit timer (T8) Ch.2 (with fine mode) (16-bit device)	0x301120	T8_CLK2	T8 Ch.2 Input Clock Select Register	Select prescaler output clock
	0x301122	T8_TR2	T8 Ch.2 Reload Data Register	Set reload data
	0x301124	T8_TC2	T8 Ch.2 Counter Data Register	Counter data
	0x301126	T8_CTL2	T8 Ch.2 Control Register	Set timer mode and start/stop timer
	0x301128	T8_INT2	T8 Ch.2 Interrupt Control Register	Control interrupt
8-bit timer (T8) Ch.3 (with fine mode) (16-bit device)	0x301130	T8_CLK3	T8 Ch.3 Input Clock Select Register	Select prescaler output clock
	0x301132	T8_TR3	T8 Ch.3 Reload Data Register	Set reload data
	0x301134	T8_TC3	T8 Ch.3 Counter Data Register	Counter data
	0x301136	T8_CTL3	T8 Ch.3 Control Register	Set timer mode and start/stop timer
	0x301138	T8_INT3	T8 Ch.3 Interrupt Control Register	Control interrupt
8-bit timer (T8) Ch.4 (16-bit device)	0x301140	T8_CLK4	T8 Ch.4 Input Clock Select Register	Select prescaler output clock
	0x301142	T8_TR4	T8 Ch.4 Reload Data Register	Set reload data
	0x301144	T8_TC4	T8 Ch.4 Counter Data Register	Counter data
	0x301146	T8_CTL4	T8 Ch.4 Control Register	Set timer mode and start/stop timer
	0x301148	T8_INT4	T8 Ch.4 Interrupt Control Register	Control interrupt
8-bit timer (T8) Ch.5 (16-bit device)	0x301150	T8_CLK5	T8 Ch.5 Input Clock Select Register	Select prescaler output clock
	0x301152	T8_TR5	T8 Ch.5 Reload Data Register	Set reload data
	0x301154	T8_TC5	T8 Ch.5 Counter Data Register	Counter data
	0x301156	T8_CTL5	T8 Ch.5 Control Register	Set timer mode and start/stop timer
	0x301158	T8_INT5	T8 Ch.5 Interrupt Control Register	Control interrupt
8-bit timer (T8) Ch.6 (16-bit device)	0x301160	T8_CLK6	T8 Ch.6 Input Clock Select Register	Select prescaler output clock
	0x301162	T8_TR6	T8 Ch.6 Reload Data Register	Set reload data
	0x301164	T8_TC6	T8 Ch.6 Counter Data Register	Counter data
	0x301166	T8_CTL6	T8 Ch.6 Control Register	Set timer mode and start/stop timer
	0x301168	T8_INT6	T8 Ch.6 Interrupt Control Register	Control interrupt
8-bit timer (T8) Ch.7 (16-bit device)	0x301170	T8_CLK7	T8 Ch.7 Input Clock Select Register	Select prescaler output clock
	0x301172	T8_TR7	T8 Ch.7 Reload Data Register	Set reload data
	0x301174	T8_TC7	T8 Ch.7 Counter Data Register	Counter data
	0x301176	T8_CTL7	T8 Ch.7 Control Register	Set timer mode and start/stop timer
	0x301178	T8_INT7	T8 Ch.7 Interrupt Control Register	Control interrupt
16-bit PWM timer (T16A5) Ch.0 (16-bit device)	0x301180	T16A_CTL0	T16A5 Ch.0 Counter Control Register	Control counter
	0x301182	T16A_TC0	T16A5 Ch.0 Counter Data Register	Counter data
	0x301184	T16A_CCCTL0	T16A5 Ch.0 Comparator/Capture Control Register	Control comparator/capture block and TOUT
	0x301186	T16A_CCA0	T16A5 Ch.0 Comparator/Capture A Data Register	Compare A/capture A data
	0x301188	T16A_CCB0	T16A5 Ch.0 Comparator/Capture B Data Register	Compare B/capture B data
	0x30118a	T16A_IEN0	T16A5 Ch.0 Comparator/Capture Interrupt Enable Register	Enable/disable T16A5 interrupts
	0x30118c	T16A_IFLG0	T16A5 Ch.0 Comparator/Capture Interrupt Flag Register	Indicate T16A5 interrupt cause status
16-bit PWM timer (T16A5) Ch.1 (16-bit device)	0x301190	T16A_CTL1	T16A5 Ch.1 Counter Control Register	Control counter
	0x301192	T16A_TC1	T16A5 Ch.1 Counter Data Register	Counter data
	0x301194	T16A_CCCTL1	T16A5 Ch.1 Comparator/Capture Control Register	Control comparator/capture block and TOUT
	0x301196	T16A_CCA1	T16A5 Ch.1 Comparator/Capture A Data Register	Compare A/capture A data

APPENDIX A LIST OF I/O REGISTERS

Peripheral	Address	Register name		Function
16-bit PWM timer (T16A5) Ch.1 (16-bit device)	0x301198	T16A_CCB1	T16A5 Ch.1 Comparator/Capture B Data Register	Compare B/capture B data
	0x30119a	T16A_IEN1	T16A5 Ch.1 Comparator/Capture Interrupt Enable Register	Enable/disable T16A5 interrupts
	0x30119c	T16A_IFLG1	T16A5 Ch.1 Comparator/Capture Interrupt Flag Register	Indicate T16A5 interrupt cause status
16-bit audio PWM timer (T16P) (16-bit device)	0x301200	T16P_A	T16P Compare A Buffer Register	Compare A data
	0x301202	T16P_B	T16P Compare B Buffer Register	Compare B data
	0x301204	T16P_CNT_DATA	T16P Counter Data Register	Counter data
	0x301206	T16P_VOL_CTL	T16P Volume Control Register	Enable volume control and set volume level
	0x301208	T16P_CTL	T16P Control Register	Set timer operating conditions
	0x30120a	T16P_RUN	T16P Running Control Register	Start/stop timer
	0x30120c	T16P_CLK	T16P Internal Clock Control Register	Select internal count clock
A/D converter (ADC10) (16-bit device)	0x301300	ADC10_ADD	A/D Conversion Result Register	A/D converted data
	0x301302	ADC10_TRG	A/D Trigger/Channel Select Register	Set start/end channels and conversion mode
	0x301304	ADC10_CTL	A/D Control/Status Register	Control A/D converter and indicate conversion status
	0x301306	ADC10_CLK	A/D Clock Control Register	Control A/D converter clock
I ² S (16-bit device)	0x301400	I2S_CTL	I ² S Control Register	Set I ² S output conditions
	0x301404	I2S_DV_MCLK	I ² S Master Clock Division ratio Register	Configure master clock
	0x301406	I2S_DV_AUDIO_CLK	I ² S Audio Clock Division ratio Register	Configure audio clock
	0x301408	I2S_START	I ² S Start/Stop Register	Control/indicate I ² S start/stop status
	0x30140a	I2S_FIFO_STAT	I ² S FIFO Status Register	Indicate FIFO status
	0x30140c	I2S_INT	I ² S Interrupt Control Register	Control I ² S interrupts
	0x301410	I2S_FIFO	I ² S FIFO Register	L-channel output data
	0x301412			R-channel output data
Remote controller (REMC) (16-bit device)	0x301500	REMC_CFG	REMC Configuration Register	Control clock and data transfer
	0x301502	REMC_CAR	REMC Carrier Length Setup Register	Set carrier H/L section lengths
	0x301504	REMC_LCNT	REMC Length Counter Register	Set transmit/receive data length
	0x301506	REMC_INT	REMC Interrupt Control Register	Control REMC interrupts
LCD controller (LCDC) (32-bit device)	0x302000	LCDC_INT	LCDC Interrupt Enable Register	Enable/disable LCDC interrupts
	0x302004	LCDC_PSAVE	Status and Power Save Configuration Register	Indicate LCDC status and set power save mode
	0x302010	LCDC_HDISP	Horizontal Display Register	Set horizontal display period
	0x302014	LCDC_VDISP	Vertical Display Register	Set vertical display period
	0x302018	LCDC_MODR	MOD Rate Register	Set MOD rate
	0x302020	LCDC_HDPS	Horizontal Display Start Position Register	Set horizontal display start position for TFT
	0x302024	LCDC_VDPS	Vertical Display Start Position Register	Set vertical display start position for TFT
	0x302028	LCDC_FPLINE	FPLINE Pulse Setting Register	Configure FPLINE pulse for TFT
	0x30202c	LCDC_FPFRR	FPFRR Pulse Setting Register	Configure FPFRR pulse for TFT
	0x302030	LCDC_FPFROFS	FPFRR Pulse Offset Register	Adjust FPLINE pulse timings for TFT
	0x302040	LCDC_TFTSO	TFT Special Output Register	Set TFT control signals
	0x302044	LCDC_TFT_CTL1	TFT_CTL1 Pulse Register	Set TFT_CTL1 pulse timings
	0x302048	LCDC_TFT_CTL0	TFT_CTL0 Pulse Register	Set TFT_CTL0 pulse timings
	0x30204c	LCDC_TFT_CTL2	TFT_CTL2 Register	Set TFT_CTL2 signal timing
	0x302050	LCDC_RLDCTL	LCDC Reload Control Register	Control reloading
	0x302054	LCDC_RLDADR	LCDC Reload Table Base Address Register	Set reload table base address
	0x302060	LCDC_DISPMDR	LCDC Display Mode Register	Set display conditions
	0x302070	LCDC_MAINADR	Main Window Display Start Address Register	Set main window display start address
	0x302074	LCDC_MAINOFS	Main Screen Address Offset Register	Set virtual main screen width
	0x302080	LCDC_SUBADR	Sub-window Display Start Address Register	Set sub-window display start address
0x302084	LCDC_SUBOFS	Sub-screen Address Offset Register	Set virtual sub-screen width	
0x302088	LCDC_SUBSP	Sub-window Start Position Register	Set sub-window start position	
0x30208c	LCDC_SUBEP	Sub-window End Position Register	Set sub-window end position	
0x302090	LCDC_MLUT0	Monochrome Look-up Table Register 0	Monochrome look-up table data entries 0–7	
0x302094	LCDC_MLUT1	Monochrome Look-up Table Register 1	Monochrome look-up table data entries 8–15	
DMA controller (DMAC) (32-bit device)	0x302100	DMAC_CH_EN	DMAC Channel Enable Register	Enable DMAC channels
	0x302104	DMAC_TBL_BASE	DMAC Control Table Base Address Register	Set control table base address
	0x302108	DMAC_IE	DMAC Interrupt Enable Register	Enable/disable DMAC interrupts
	0x30210c	DMAC_TRG_SEL	DMAC Trigger Select Register	Select trigger sources
	0x302110	DMAC_TRG_FLG	DMAC Trigger Flag Register	Control software trigger and indicate trigger status
	0x302114	DMAC_END_TRANSFER	DMAC End-of-Transfer Flag Register	Indicate DMA completed channels
	0x302118	DMAC_RUN_STAT	DMAC Running Status Register	Indicates running channel
0x30211c	DMAC_PAUSE_STAT	DMAC Pause Status Register	Indicate DMA suspended channels	

APPENDIX A LIST OF I/O REGISTERS

Peripheral	Address	Register name		Function
SDRAM controller (SDRAMC) (32-bit device)	0x302200	SDRAMC_INIT	SDRAM Initialization Register	Enable SDRAMC and control SDRAM initialization
	0x302204	SDRAMC_CFG	SDRAM Configuration Register	Set SDRAM size and timing parameters
	0x302208	SDRAMC_REF	SDRAM Refresh Control Register	Control SDRAM refresh
	0x302210	SDRAMC_APP	SDRAM Application Configuration Register	Set CAS latency and double frequency mode
SRAM controller (SRAMC) (32-bit device)	0x302220	SRAMC_TMG47	#CE[7:4] Access Timing Configuration Register	Set #CE[7:4] access conditions
	0x302224	SRAMC_TMG810	#CE[10:8] Access Timing Configuration Register	Set #CE[10:8] access conditions
	0x302228	SRAMC_TYPE	#CE[10:4] Device Configuration Register	Set #CE[10:4] device types
Cache controller (CCU) (32-bit device)	0x302300	CCU_CFG	Cache Configuration Register	Enable instruction and data caches
	0x302304	CCU_AREA	Cacheable Area Select Register	Select cacheable areas
	0x302308	CCU_LK	Cache Lock Register	Configure cache lock function
	0x30230c	CCU_STAT	Cache Status Register	Indicate cache statuses
	0x302318	CCU_WB_STAT	Cache Write Buffer Status Register	Indicate write buffer status
	0x302360	CCU_CCLKDV	CCLK Division Ratio Select Register	Set CCLK clock frequency.
Graphics engine (GE) (32-bit device)	0x30240c	GE_CMD_ADDR	GE Command Address Register	Set command list start address
	0x302440	GE_CTL	GE Control Register	Control/indicate GE operation/status
	0x302444	GE_IE	GE Interrupt Enable Register	Enable/disable GE interrupts
	0x302448	GE_IF1	GE Interrupt Flag Register 1	Indicate calculation error status
	0x302449	GE_IF2	GE Interrupt Flag Register 2	Indicate drawing error status
	0x30244a	GE_IF3	GE Interrupt Flag Register 3	Indicate cause of termination
	0x30244c	GE_REAL_W	VRAM Work Area Width Register	Indicate work area width after VRAM rotation
	0x302450	GE_WK_ADDR	VRAM Work Area Start Address Register	Set work (drawing) area start address
	0x302454	GE_WK_SIZE	VRAM Work Area Size Register	Set work area size
	0x302458	GE_DISP_CFG	Display Configuration Register	Set color depth and LCD sync type
	0x30245c	GE_ROTATE	VRAM Rotation Control Register	Specify display rotation angle
	0x302460	GE_CLIP_ST	Clipping Area Start Position Register	Specify upper left corner of clipping area
	0x302464	GE_CLIP_END	Clipping Area End Position Register	Specify lower right corner of clipping area
	0x302468	GE_MESH	Mesh Configuration Register	Configure mesh size and color
	0x30246c	GE_MAGIC	Transparent Color Register	Specify transparent color
	0x302470	GE_UPDT_ST	Updated Area Start Position Register	Indicate upper left corner of updated area
	0x302474	GE_UPDT_END	Updated Area End Position Register	Indicate lower right corner of updated area
	0x302800 –0x3028ff	GE_PALETTE1	Palette 1	Palette 1
	0x302910 –0x30291f	GE_CCT1_4BIT	CCT1 4-bit Entries	CCT1 4 to 8 bpp conversion entries (user programmable)
	0x302920 –0x302923	GE_CCT1_2BIT	CCT1 2-bit Entries	CCT1 2 to 4/8 bpp conversion entries (user programmable)
0x302924 –0x302925	GE_CCT1_1BIT	CCT1 1-bit Entries	CCT1 1 to 2/4/8 bpp conversion entries (user programmable)	

Notes: • Do not access the unused peripheral circuit areas not listed in the table from the application program.

- When data is written to the registers, the “Reserved” bits must always be written as 0 and not 1, except RTCIMD/RTC_INTMODE register (D1/0x300a01) for RTC.
- When accessing the USB registers, the USBCLK clock must be supplied to the USB function controller in addition to USBREGCLK.

0x300010–0x300020

Misc Registers (MISC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC Wait Control Register (MISC_RTCWT)	0x300010 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.	
		D2–0	RTCWT[2:0]	RTC register access wait control	0 to 7 cycles	0x7	R/W	Write-protected	
USB Configuration Register (MISC_USB)	0x300012 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6	USBINTEN	USB interrupt enable	1 Enable 0 Disable	0	R/W	Write-protected	
		D5	USBSNZ	USB snooze control	1 Enable 0 Disable	0	R/W		
		D4–3	–	reserved	–	–	–	–	0 when being read.
		D2–0	USBWT[2:0]	USB register access wait control	0 to 7 cycles	0x7	R/W	Write-protected	
Internal RAM Wait Control Register (MISC_RAMWT)	0x300014 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.	
		D1	COREWT	IRAM (12KB) access wait control	1 1 cycle 0 0 cycles	1	R/W	Write-protected	
		D0	BUSWT	IVRAM (20KB) access wait control	1 1 cycle 0 0 cycles	1	R/W		
Boot Register (MISC_BOOT)	0x300016 (8 bits)	D7–4	BOOT[3:0]	Boot mode indicator	BOOT[3:0] Boot mode 0b1000 SPI/RS232C 0b0100 NOR/ROM Other reserved	*	R	* Depends on the BOOT pin status at initial reset	
		D3–2	–	reserved	–	–	–	0 when being read.	
		D1	BOOT_ENA	#CE10 area boot enable	1 Internal 0 External	1	R/W	Write-protected	
		D0	–	reserved	–	–	–	–	0 when being read.
RAM Location Select Register (MISC_RAM_LOC)	0x300018 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.	
		D4	DSTRAM_CFG	DSTRAM configuration	1 LUTRAM 0 DSTRAM	0	R/W	Write-protected	
		D3–1	–	reserved	–	–	–	–	0 when being read.
		D0	IVRAM_LOC	IVRAM location select	1 Area 3 0 Area 0	1	R/W	Write-protected	
Misc Protect Register (MISC_PROTECT)	0x300020 (8 bits)	D7–0	PROT[7:0]	Misc register write-protect flag	Writing 10010110 (0x96) removes the write protection of the Misc registers (0x300010–0x300018). Writing another value set the write protection.	0x0	R/W		

0x300100–0x300110

Clock Management Unit (CMU)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Clock Source Select Register (CMU_OSCSEL)	0x300100 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.		
		D1–0	CLKSEL [1:0]	System clock source select	CLKSEL[1:0] Clock source 0x3 Not allowed 0x2 PLL 0x1 OSC1 0x0 OSC3	0x0	R/W	Write-protected		
Oscillation Control Register (CMU_OSCCTL)	0x300101 (8 bits)	D7–4	OSC3WT[3:0]	OSC3 wait cycle select	OSC3WT[3:0] Wait cycle 0xf 128 cycles 0xe 256 cycles 0xd 512 cycles 0xc 1,024 cycles 0xb 2,048 cycles 0xa 4,096 cycles 0x9 8,192 cycles 0x8 16,384 cycles 0x7 32,768 cycles 0x6 65,536 cycles 0x5 131,072 cycles 0x4 262,144 cycles 0x3 524,288 cycles 0x2 1,048,576 cycles 0x1 2,097,152 cycles 0x0 4,194,304 cycles	0xf	R/W	Write-protected		
		D3–2	–	reserved	–	–	–	–	0 when being read.	
		D1	OSC1EN	OSC1 enable	1 Enable 0 Disable	1	R/W	Write-protected		
		D0	OSC3EN	OSC3 enable	1 Enable 0 Disable	1	R/W			

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
LCD Clock Division Ratio Select Register (CMU_LCLKDIV)	0x300103 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4-0	LCLKDIV[4:0]	LCDC clock division ratio select	LCLKDIV[4:0] Division ratio	0x7	R/W	Clock source = OSC3 Write-protected	
						0x1f	1/32		
						0x1e	1/31		
						0x1d	1/30		
						0x1c	1/29		
						0x1b	1/28		
						0x1a	1/27		
						0x19	1/26		
						0x18	1/25		
						0x17	1/24		
						0x16	1/23		
						0x15	1/22		
						0x14	1/21		
						0x13	1/20		
						0x12	1/19		
						0x11	1/18		
						0x10	1/17		
						0xf	1/16		
						0xe	1/15		
						0xd	1/14		
				0xc	1/13				
				0xb	1/12				
				0xa	1/11				
				0x9	1/10				
				0x8	1/9				
				0x7	1/8				
				0x6	1/7				
				0x5	1/6				
				0x4	1/5				
				0x3	1/4				
				0x2	1/3				
				0x1	1/2				
				0x0	1/1				
Clock Control Register (CMU_CLKCTL)	0x300104 (8 bits)	D7	USBREGCLK_EN	USB I/O register clock enable	1 Enable	0 Disable	0	R/W	Write-protected
		D6	LCLK_EN	LCDC clock enable	1 Enable	0 Disable	0	R/W	
		D5	USBCLK_EN	USB clock enable	1 Enable	0 Disable	0	R/W	
		D4	SDCLK_EN	SDCLK clock enable	1 Enable	0 Disable	1	R/W	
		D3	BCLK_EN	BCLK clock enable (in HALT)	1 Enable	0 Disable	1	R/W	
		D2	PCLK2_EN	PCLK2 clock enable	1 Enable	0 Disable	1	R/W	
		D0	PCLK1_EN	PCLK1 clock enable	1 Enable	0 Disable	1	R/W	
System Clock Division Ratio Select Register (CMU_SYSCLKDIV)	0x300105 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4	MCLKDIV	MCLK clock divider select	1 1/2	0 1/1	0	R/W	Write-protected
		D3	–	reserved	–	–	–	–	0 when being read.
		D2-0	SYSCLKDIV[2:0]	System clock division ratio select	SYSCLKDIV[2:0] Division ratio	0x0	R/W	Clock source = OSC (OSC3, PLL, or OSC1) Write-protected	
						0x7-0x6	1/1		
						0x5	1/32		
						0x4	1/16		
				0x3	1/8				
				0x2	1/4				
				0x1	1/2				
				0x0	1/1				
CMU_CLK Select Register (CMU_CMUCLK)	0x300106 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4-0	CMU_CLKSEL[4:0]	CMU_CLK select	CMU_CLKSEL[4:0] CMU_CLK	0x0	R/W	OSC: system clock (OSC3, PLL, OSC1) Write-protected	
						0xf-0xb	reserved		
						0xa	OSC•1/32		
						0x9	OSC•1/16		
						0x8	OSC•1/8		
						0x7	OSC•1/4		
						0x6	OSC•1/2		
						0x5	OSC•1/1		
						0x4	LCLK		
						0x3	BCLK		
						0x2	PLL		
						0x1	OSC1		
						0x0	OSC3		
PLL Input Clock Division Ratio Select Register (CMU_PLLINDIV)	0x300107 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.	
		D3-0	PLLINDIV[3:0]	PLL input clock division ratio select	PLLINDIV[3:0] Division ratio	0x7	R/W	Clock source = OSC3 Write-protected	
						0xf-0xa	1/8		
						0x9	1/10		
						0x8	1/9		
						0x7	1/8		
						0x6	1/7		
						0x5	1/6		
						0x4	1/5		
						0x3	1/4		
						0x2	1/3		
						0x1	1/2		
						0x0	1/1		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PLL Control Register 0 (CMU_PLLCTL0)	0x300108 (8 bits)	D7-4	PLLN[3:0]	PLL multiplication rate setup	PLLN[3:0]	Multiplication rate	0x0	R/W	Write-protected
		D3-2	PLLV[1:0]	PLL V-divider setup	PLLV[1:0]	W	0x1	R/W	
		D1	–	reserved			–	–	0 when being read.
		D0	PLLPOWR	PLL enable	1 Enable 0 Disable		0	R/W	Write-protected
PLL Control Register 1 (CMU_PLLCTL1)	0x300109 (8 bits)	D7-4	PLLVC[3:0]	PLL VCO Kv setup	PLLVC[3:0]	f _{vco} [MHz]	0x1	R/W	Write-protected
		D3-0	PLLRS[3:0]	PLL LPF resistance setup	PLLRS[3:0]	f _{REFCK} [MHz]	0x8	R/W	
PLL Control Register 2 (CMU_PLLCTL2)	0x30010a (8 bits)	D7-6	PLLCS[1:0]	PLL LPF capacitance		0x0	0x0	R	
		D5	PLLBYB	PLL bypass mode		0	0	R	
		D4-0	PLLCP[4:0]	PLL charge pump current		0x10	0x10	R	
SSCG Macro Control Register 0 (CMU_SSCG0)	0x30010c (8 bits)	D7-1	–	reserved		–	–	–	0 when being read.
		D0	SSMCON	SSCG enable	1 Enable 0 Disable		0	R/W	Write-protected
SSCG Macro Control Register 1 (CMU_SSCG1)	0x30010d (8 bits)	D7-4	SSMCITM[3:0]	SSCG interval timer (ITM) setting		0x0 to 0xf	X	R/W	Write-protected
		D3-0	SSMCIDT[3:0]	SSCG maximum frequency change width setting		0x0 to 0xf	X	R/W	
CMU Write Protect Register (CMU_PROTECT)	0x300110 (8 bits)	D7-0	CMUP[7:0]	CMU register write-protect flag		Writing 10010110 (0x96) removes the write protection of the CMU registers (0x300100–0x30010d). Writing another value set the write protection.	0x0	R/W	Write-protected

0x300210–0x30022c

Interrupt Controller (ITC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT0-3 Interrupt Level Register (ITC_FPT03_LV)	0x300210 (8 bits)	D7-3	–	reserved		–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	FPT0-3 interrupt level		1 to 7	0x0	R/W	
FPT4-7 Interrupt Level Register (ITC_FPT47_LV)	0x300211 (8 bits)	D7-3	–	reserved		–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	FPT4-7 interrupt level		1 to 7	0x0	R/W	
FPT8-B Interrupt Level Register (ITC_FPT8B_LV)	0x300212 (8 bits)	D7-3	–	reserved		–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	FPT8-B interrupt level		1 to 7	0x0	R/W	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPTC-F Interrupt Level Register (ITC_FPTCF_LV)	0x300213 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	FPTC-F interrupt level	1 to 7	0x0	R/W	
DMAC Ch.0 & 2 Interrupt Level Register (ITC_DMA02_LV)	0x300214 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	DMAC Ch.0 and 2 interrupt level	1 to 7	0x0	R/W	
DMAC Ch.1 & 3 Interrupt Level Register (ITC_DMA13_LV)	0x300215 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	DMAC Ch.1 and 3 interrupt level	1 to 7	0x0	R/W	
DMAC Ch.4 & 6 Interrupt Level Register (ITC_DMA46_LV)	0x300216 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	DMAC Ch.4 and 6 interrupt level	1 to 7	0x0	R/W	
DMAC Ch.5 & 7 Interrupt Level Register (ITC_DMA57_LV)	0x300217 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	DMAC Ch.5 and 7 interrupt level	1 to 7	0x0	R/W	
T16P Interrupt Level Register (ITC_T16P_LV)	0x300218 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	T16P interrupt level	1 to 7	0x0	R/W	
T16A5 Ch.0 Interrupt Level Register (ITC_T16A0_LV)	0x300219 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	T16A5 Ch.0 interrupt level	1 to 7	0x0	R/W	
T16A5 Ch.1 Interrupt Level Register (ITC_T16A1_LV)	0x30021a (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	T16A5 Ch.1 interrupt level	1 to 7	0x0	R/W	
LCDC Interrupt Level Register (ITC_LCDC_LV)	0x30021b (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	LCDC interrupt level	1 to 7	0x0	R/W	
T8 Ch.0 & 4 Interrupt Level Register (ITC_T804_LV)	0x30021d (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	T8 Ch.0 and 4 interrupt level	1 to 7	0x0	R/W	
T8 Ch.1 & 5 Interrupt Level Register (ITC_T815_LV)	0x30021e (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	T8 Ch.1 and 5 interrupt level	1 to 7	0x0	R/W	
T8 Ch.2 & 6 Interrupt Level Register (ITC_T826_LV)	0x30021f (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	T8 Ch.2 and 6 interrupt level	1 to 7	0x0	R/W	
T8 Ch.3 & 7 Interrupt Level Register (ITC_T837_LV)	0x300220 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	T8 Ch.3 and 7 interrupt level	1 to 7	0x0	R/W	
USI Interrupt Level Register (ITC_USI_LV)	0x300221 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	USI interrupt level	1 to 7	0x0	R/W	
FSIO Ch.0 Interrupt Level Register (ITC_FSIO0_LV)	0x300222 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	FSIO Ch.0 interrupt level	1 to 7	0x0	R/W	
ADC10 Interrupt Level Register (ITC_ADC10_LV)	0x300223 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	ADC10 interrupt level	1 to 7	0x0	R/W	
RTC Interrupt Level Register (ITC_RTC_LV)	0x300224 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	RTC interrupt level	1 to 7	0x0	R/W	
FSIO Ch.1 Interrupt Level Register (ITC_FSIO1_LV)	0x300226 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	FSIO Ch.1 interrupt level	1 to 7	0x0	R/W	
USIL Interrupt Level Register (ITC_USIL_LV)	0x300227 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	USIL interrupt level	1 to 7	0x0	R/W	
REMC Interrupt Level Register (ITC_REMC_LV)	0x300228 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	REMC interrupt level	1 to 7	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S Interrupt Level Register (ITC_I2S_LV)	0x300229 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	I ² S interrupt level	1 to 7	0x0	R/W	
GE Complete Interrupt Level Register (ITC_GECOM_LV)	0x30022a (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	GE complete interrupt level	1 to 7	0x0	R/W	
GE Error Interrupt Level Register (ITC_GEERR_LV)	0x30022b (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	GE error interrupt level	1 to 7	0x0	R/W	
USB Interrupt Level Register (ITC_USB_LV)	0x30022c (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	INT_LV[2:0]	USB interrupt level	1 to 7	0x0	R/W	

0x300300–0x30083f

GPIO & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P0 Port Data Register (GPIO_P0_DAT)	0x300300 (8 bits)	D7-0	P0[7:0]D	P0[7:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
P0 Port I/O Control Register (GPIO_P0_IOC)	0x300301 (8 bits)	D7-0	IOC0[7:0]	P0[7:0] I/O control	1 Output 0 Input	0x0	R/W	
P1 Port Data Register (GPIO_P1_DAT)	0x300302 (8 bits)	D7-0	P1[7:0]D	P1[7:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
P1 Port I/O Control Register (GPIO_P1_IOC)	0x300303 (8 bits)	D7-0	IOC1[7:0]	P1[7:0] I/O control	1 Output 0 Input	0x0	R/W	
P2 Port Data Register (GPIO_P2_DAT)	0x300304 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1-0	P2[1:0]D	P2[1:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
P2 Port I/O Control Register (GPIO_P2_IOC)	0x300305 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1-0	IOC2[1:0]	P2[1:0] I/O control	1 Output 0 Input	0x0	R/W	
P3 Port Data Register (GPIO_P3_DAT)	0x300306 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	P3[6:0]D	P3[6:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
P3 Port I/O Control Register (GPIO_P3_IOC)	0x300307 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	IOC3[6:0]	P3[6:0] I/O control	1 Output 0 Input	0x0	R/W	
P4 Port Data Register (GPIO_P4_DAT)	0x300308 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	P4[2:0]D	P4[2:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
P4 Port I/O Control Register (GPIO_P4_IOC)	0x300309 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	IOC4[2:0]	P4[2:0] I/O control	1 Output 0 Input	0x0	R/W	
P5 Port Data Register (GPIO_P5_DAT)	0x30030a (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	P5[6:0]D	P5[6:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
P5 Port I/O Control Register (GPIO_P5_IOC)	0x30030b (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	IOC5[6:0]	P5[6:0] I/O control	1 Output 0 Input	0x0	R/W	
P6 Port Data Register (GPIO_P6_DAT)	0x30030c (8 bits)	D7-1	–	reserved	–	–	–	0 when being read.
		D0	P60D	P60 I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
P6 Port I/O Control Register (GPIO_P6_IOC)	0x30030d (8 bits)	D7-1	–	reserved	–	–	–	0 when being read.
		D0	IOC60	P60 I/O control	1 Output 0 Input	0	R/W	
P7 Port Data Register (GPIO_P7_DAT)	0x30030e (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.
		D5-0	P7[5:0]D	P7[5:0] input port data	1 1 (High) 0 0 (Low)	Ext.	R	Ext.: Depends on the external pin status.
P8 Port Data Register (GPIO_P8_DAT)	0x300310 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.
		D3-0	P8[3:0]D	P8[3:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P8 Port I/O Control Register (GPIO_P8_IOC)	0x300311 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.
		D3-0	IOC8[3:0]	P8[3:0] I/O control	1 Output 0 Input	0x0	R/W	
P9 Port Data Register (GPIO_P9_DAT)	0x300312 (8 bits)	D7-0	P9[7:0]D	P9[7:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
P9 Port I/O Control Register (GPIO_P9_IOC)	0x300313 (8 bits)	D7-0	IOC9[7:0]	P9[7:0] I/O control	1 Output 0 Input	0x0	R/W	
PA Port Data Register (GPIO_PA_DAT)	0x300314 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	PA[6:0]D	PA[6:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
PA Port I/O Control Register (GPIO_PA_IOC)	0x300315 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	IOCA[6:0]	PA[6:0] I/O control	1 Output 0 Input	0x0	R/W	
PB Port Data Register (GPIO_PB_DAT)	0x300316 (8 bits)	D7-0	PB[7:0]D	PB[7:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
PB Port I/O Control Register (GPIO_PB_IOC)	0x300317 (8 bits)	D7-0	IOCB[7:0]	PB[7:0] I/O control	1 Output 0 Input	0x0	R/W	
PC Port Data Register (GPIO_PC_DAT)	0x300318 (8 bits)	D7-0	PC[7:0]D	PC[7:0] I/O port data	1 1 (High) 0 0 (Low)	Ext.	R/W	Ext.: Depends on the external pin status.
PC Port I/O Control Register (GPIO_PC_IOC)	0x300319 (8 bits)	D7-0	IOCC[7:0]	PC[7:0] I/O control	1 Output 0 Input	0x0	R/W	
Bus Drive Control Register (GPIO_BUS_DRV)	0x300320 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	LDRVDB	D[15:0] low drive	1 Low drive 0 Normal output	0	R/W	Write-protected
		D0	LDRVAD	A[25:0] low drive		0	R/W	
P0 Port Pull-up Control Register (GPIO_P0_PUP)	0x300321 (8 bits)	D7-0	PUP0[7:0]	P0[7:0] port pull-up enable	1 Enable 0 Disable	0x0 *	R/W	* P0[7:0]: not pulled up Write-protected
P1 Port Pull-up Control Register (GPIO_P1_PUP)	0x300322 (8 bits)	D7-0	PUP1[7:0]	P1[7:0] port pull-up enable	1 Enable 0 Disable	0x0 *	R/W	* P1[7:0]: not pulled up Write-protected
P2 Port Pull-up Control Register (GPIO_P2_PUP)	0x300323 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1-0	PUP2[1:0]	P2[1:0] port pull-up enable	1 Enable 0 Disable	0x0 *	R/W	* P2[1:0]: not pulled up Write-protected
P3 Port Pull-up Control Register (GPIO_P3_PUP)	0x300324 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	PUP3[6:0]	P3[6:0] port pull-up enable	1 Enable 0 Disable	0x20 *	R/W	* P35: pulled up, others: not pulled up Write-protected
P4 Port Pull-up Control Register (GPIO_P4_PUP)	0x300325 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2-0	PUP4[2:0]	P4[2:0] port pull-up enable	1 Enable 0 Disable	0x0 *	R/W	* P4[2:0]: not pulled up Write-protected
P5 Port Pull-up Control Register (GPIO_P5_PUP)	0x300326 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	PUP5[6:0]	P5[6:0] port pull-up enable	1 Enable 0 Disable	0x8 *	R/W	* P53: pulled up, others: not pulled up Write-protected
P6 Port Pull-up Control Register (GPIO_P6_PUP)	0x300327 (8 bits)	D7-1	–	reserved	–	–	–	0 when being read.
		D0	PUP6	P60 port pull-up enable	1 Enable 0 Disable	1 *	R/W	* P60: pulled up Write-protected
P7 Port Pull-up Control Register (GPIO_P7_PUP)	0x300328 (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.
		D5-0	PUP7[5:0]	P7[5:0] port pull-up enable	1 Enable 0 Disable	0x20 *	R/W	* P75: pulled up, others: not pulled up Write-protected
P8 Port Pull-up Control Register (GPIO_P8_PUP)	0x300329 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.
		D3-0	PUP8[3:0]	P8[3:0] port pull-up enable	1 Enable 0 Disable	0x0 *	R/W	* P8[3:0]: not pulled up Write-protected
P9 Port Pull-up Control Register (GPIO_P9_PUP)	0x30032a (8 bits)	D7-0	PUP9[7:0]	P9[7:0] port pull-up enable	1 Enable 0 Disable	0x0 *	R/W	* P9[7:0]: not pulled up Write-protected
PA Port Pull-up Control Register (GPIO_PA_PUP)	0x30032b (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6-0	PUPA[6:0]	PA[6:0] port pull-up enable	1 Enable 0 Disable	0x2 *	R/W	* PA1: pulled up, others: not pulled up Write-protected

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks		
PB Port Pull-up Control Register (GPIO_PB_PUP)	0x30032c (8 bits)	D7-0	PUPB[7:0]	PB[7:0] port pull-up enable	1	Enable	0	Disable	0x0 *	R/W * PB[7:0]: not pulled up Write-protected	
FPT0-3 Interrupt Port Select Register (GPIO_FPT03_SEL)	0x300330 (8 bits)	D7-6	SPT3[1:0]	FPT3 interrupt input port select	SPT3[1:0]		Port		0x0	R/W	
					0x3	P33					
					0x2	P13					
					0x1	P53					
		D5-4	SPT2[1:0]	FPT2 interrupt input port select	SPT2[1:0]		Port		0x0	R/W	
					0x3	P32					
					0x2	P12					
					0x1	P52					
		D3-2	SPT1[1:0]	FPT1 interrupt input port select	SPT1[1:0]		Port		0x0	R/W	
					0x3	P31					
					0x2	P11					
					0x1	P51					
D1-0	SPT0[1:0]	FPT0 interrupt input port select	SPT0[1:0]		Port		0x0	R/W			
			0x3	P30							
			0x2	P10							
			0x1	P50							
FPT4-7 Interrupt Port Select Register (GPIO_FPT47_SEL)	0x300331 (8 bits)	D7-6	SPT7[1:0]	FPT7 interrupt input port select	SPT7[1:0]		Port		0x0	R/W	
					0x3	PA3					
					0x2	P17					
					0x1	PC7					
		D5-4	SPT6[1:0]	FPT6 interrupt input port select	SPT6[1:0]		Port		0x0	R/W	
					0x3	PA2					
					0x2	P16					
					0x1	PC6					
		D3-2	SPT5[1:0]	FPT5 interrupt input port select	SPT5[1:0]		Port		0x0	R/W	
					0x3	PA1					
					0x2	P15					
					0x1	PC5					
D1-0	SPT4[1:0]	FPT4 interrupt input port select	SPT4[1:0]		Port		0x0	R/W			
			0x3	PA0							
			0x2	P14							
			0x1	PC4							
FPT8-B Interrupt Port Select Register (GPIO_FPT8B_SEL)	0x300332 (8 bits)	D7-6	SPTB[1:0]	FPTB interrupt input port select	SPTB[1:0]		Port		0x0	R/W	
					0x3	P93					
					0x2	P83					
					0x1	PB3					
		D5-4	SPTA[1:0]	FPTA interrupt input port select	SPTA[1:0]		Port		0x0	R/W	
					0x3	P92					
					0x2	P82					
					0x1	PB2					
		D3-2	SPT9[1:0]	FPT9 interrupt input port select	SPT9[1:0]		Port		0x0	R/W	
					0x3	P91					
					0x2	P81					
					0x1	PB1					
D1-0	SPT8[1:0]	FPT8 interrupt input port select	SPT8[1:0]		Port		0x0	R/W			
			0x3	P90							
			0x2	P80							
			0x1	PB0							
					0x0	P40					

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks	
FPTC-F Interrupt Port Select Register (GPIO_FPTCF_ SEL)	0x300333 (8 bits)	D7-6	SPTF[1:0]	FPTF interrupt input port select	SPTF[1:0]	Port	0x0	R/W		
					0x3	P97				
					0x2	P60				
					0x1	PB7				
			0x0	P75						
		D5-4	SPTE[1:0]	FPTTE interrupt input port select	SPTE[1:0]	Port	0x0	R/W		
					0x3	P96				
					0x2	PA6				
					0x1	PB6				
					0x0	P74				
		D3-2	SPTD[1:0]	FPTD interrupt input port select	SPTD[1:0]	Port	0x0	R/W		
					0x3	P95				
					0x2	PA5				
					0x1	PB5				
					0x0	P73				
		D1-0	SPTC[1:0]	FPTC interrupt input port select	SPTC[1:0]	Port	0x0	R/W		
					0x3	P94				
					0x2	PA4				
					0x1	PB4				
					0x0	P72				
FPT0-3 Interrupt Polarity Select Register (GPIO_FPT03_ POL)	0x300334 (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SPPT3	FPT3 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D2	SPPT2	FPT2 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D1	SPPT1	FPT1 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D0	SPPT0	FPT0 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
FPT4-7 Interrupt Polarity Select Register (GPIO_FPT47_ POL)	0x300335 (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SPPT7	FPT7 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D2	SPPT6	FPT6 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D1	SPPT5	FPT5 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D0	SPPT4	FPT4 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
FPT8-B Interrupt Polarity Select Register (GPIO_FPT8B_ POL)	0x300336 (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SPPTB	FPTB input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D2	SPPTA	FPTA input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D1	SPPT9	FPT9 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D0	SPPT8	FPT8 input polarity select	1	High / ↑	0	Low / ↓	1	R/W
FPTC-F Interrupt Polarity Select Register (GPIO_FPTCF_ POL)	0x300337 (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SPPTF	FPTF input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D2	SPPTE	FPTTE input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D1	SPPTD	FPTD input polarity select	1	High / ↑	0	Low / ↓	1	R/W
		D0	SPPTC	FPTC input polarity select	1	High / ↑	0	Low / ↓	1	R/W
FPT0-3 Interrupt Mode Select Register (GPIO_FPT03_ MOD)	0x300338 (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SEPT3	FPT3 interrupt mode select	1	Edge	0	Level	1	R/W
		D2	SEPT2	FPT2 interrupt mode select	1	Edge	0	Level	1	R/W
		D1	SEPT1	FPT1 interrupt mode select	1	Edge	0	Level	1	R/W
		D0	SEPT0	FPT0 interrupt mode select	1	Edge	0	Level	1	R/W
FPT4-7 Interrupt Mode Select Register (GPIO_FPT47_ MOD)	0x300339 (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SEPT7	FPT7 interrupt mode select	1	Edge	0	Level	1	R/W
		D2	SEPT6	FPT6 interrupt mode select	1	Edge	0	Level	1	R/W
		D1	SEPT5	FPT5 interrupt mode select	1	Edge	0	Level	1	R/W
		D0	SEPT4	FPT4 interrupt mode select	1	Edge	0	Level	1	R/W
FPT8-B Interrupt Mode Select Register (GPIO_FPT8B_ MOD)	0x30033a (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SEPTB	FPTB interrupt mode select	1	Edge	0	Level	1	R/W
		D2	SEPTA	FPTA interrupt mode select	1	Edge	0	Level	1	R/W
		D1	SEPT9	FPT9 interrupt mode select	1	Edge	0	Level	1	R/W
		D0	SEPT8	FPT8 interrupt mode select	1	Edge	0	Level	1	R/W
FPTC-F Interrupt Mode Select Register (GPIO_FPTCF_ MOD)	0x30033b (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SEPTF	FPTF interrupt mode select	1	Edge	0	Level	1	R/W
		D2	SEPTE	FPTTE interrupt mode select	1	Edge	0	Level	1	R/W
		D1	SEPTD	FPTD interrupt mode select	1	Edge	0	Level	1	R/W
		D0	SEPTC	FPTC interrupt mode select	1	Edge	0	Level	1	R/W
FPT0-3 Interrupt Mask Register (GPIO_FPT03_ MSK)	0x30033c (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SIET3	FPT3 interrupt enable	1	Enable	0	Disable	0	R/W
		D2	SIET2	FPT2 interrupt enable	1	Enable	0	Disable	0	R/W
		D1	SIET1	FPT1 interrupt enable	1	Enable	0	Disable	0	R/W
		D0	SIET0	FPT0 interrupt enable	1	Enable	0	Disable	0	R/W
FPT4-7 Interrupt Mask Register (GPIO_FPT47_ MSK)	0x30033d (8 bits)	D7-4	-	reserved	-		-	-	0 when being read.	
		D3	SIET7	FPT7 interrupt enable	1	Enable	0	Disable	0	R/W
		D2	SIET6	FPT6 interrupt enable	1	Enable	0	Disable	0	R/W
		D1	SIET5	FPT5 interrupt enable	1	Enable	0	Disable	0	R/W
		D0	SIET4	FPT4 interrupt enable	1	Enable	0	Disable	0	R/W

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT8-B Interrupt Mask Register (GPIO_FPT8B_ MSK)	0x30033e (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	SIETB	FPTB interrupt enable	1 Enable 0 Disable	0	R/W		
		D2	SIETA	FPTA interrupt enable	1 Enable 0 Disable	0	R/W		
		D1	SIET9	FPT9 interrupt enable	1 Enable 0 Disable	0	R/W		
		D0	SIET8	FPT8 interrupt enable	1 Enable 0 Disable	0	R/W		
FPTC-F Interrupt Mask Register (GPIO_FPTCF_ MSK)	0x30033f (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	SIETF	FPTF interrupt enable	1 Enable 0 Disable	0	R/W		
		D2	SIETE	FPT E interrupt enable	1 Enable 0 Disable	0	R/W		
		D1	SIETD	FPTD interrupt enable	1 Enable 0 Disable	0	R/W		
		D0	SIETC	FPTC interrupt enable	1 Enable 0 Disable	0	R/W		
FPT0-3 Interrupt Flag Register (GPIO_FPT03_ FLG)	0x300340 (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	SFGP3	FPT3 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	X	R/W	Reset by writing 1.	
		D2	SFGP2	FPT2 interrupt flag		X	R/W		
		D1	SFGP1	FPT1 interrupt flag		X	R/W		
		D0	SFGP0	FPT0 interrupt flag		X	R/W		
FPT4-7 Interrupt Flag Register (GPIO_FPT47_ FLG)	0x300341 (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	SFGP7	FPT7 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	X	R/W	Reset by writing 1.	
		D2	SFGP6	FPT6 interrupt flag		X	R/W		
		D1	SFGP5	FPT5 interrupt flag		X	R/W		
		D0	SFGP4	FPT4 interrupt flag		X	R/W		
FPT8-B Interrupt Flag Register (GPIO_FPT8B_ FLG)	0x300342 (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	SFGPB	FPTB interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	X	R/W	Reset by writing 1.	
		D2	SFGPA	FPTA interrupt flag		X	R/W		
		D1	SFGP9	FPT9 interrupt flag		X	R/W		
		D0	SFGP8	FPT8 interrupt flag		X	R/W		
FPTC-F Interrupt Flag Register (GPIO_FPTCF_ FLG)	0x300343 (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	SFGPF	FPTF interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	X	R/W	Reset by writing 1.	
		D2	SFGPE	FPT E interrupt flag		X	R/W		
		D1	SFGPD	FPTD interrupt flag		X	R/W		
		D0	SFGPC	FPTC interrupt flag		X	R/W		
FPT0-1 Interrupt Chattering Filter Control Register (GPIO_FPT01_ CHAT)	0x300344 (8 bits)	D7	-	reserved	-	-	-	0 when being read.	
		D6-4	SCTP1[2:0]	FPT1 chattering filter time select	SCTP1[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
		0x0	None						
		D3	-	reserved	-	-	-	-	0 when being read.
		D2-0	SCTP0[2:0]	FPT0 chattering filter time select	SCTP0[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
0x6	32/fPCLK2								
0x5	16/fPCLK2								
0x4	8/fPCLK2								
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								
FPT2-3 Interrupt Chattering Filter Control Register (GPIO_FPT23_ CHAT)	0x300345 (8 bits)	D7	-	reserved	-	-	-	0 when being read.	
		D6-4	SCTP3[2:0]	FPT3 chattering filter time select	SCTP3[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
		0x0	None						
		D3	-	reserved	-	-	-	-	0 when being read.
		D2-0	SCTP2[2:0]	FPT2 chattering filter time select	SCTP2[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
0x6	32/fPCLK2								
0x5	16/fPCLK2								
0x4	8/fPCLK2								
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPT4-5 Interrupt Chattering Filter Control Register (GPIO_FPT45_ CHAT)	0x300346 (8 bits)	D7	-	reserved	-	-	-	0 when being read.	
		D6-4	SCTP5[2:0]	FPT5 chattering filter time select	SCTP5[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
		0x0	None						
D3	-	reserved	-	-	-	0 when being read.			
D2-0	SCTP4[2:0]	FPT4 chattering filter time select	SCTP4[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
			0x3	4/fPCLK2					
			0x2	2/fPCLK2					
			0x1	1/fPCLK2					
			0x0	None					
FPT6-7 Interrupt Chattering Filter Control Register (GPIO_FPT67_ CHAT)	0x300347 (8 bits)	D7	-	reserved	-	-	-	0 when being read.	
		D6-4	SCTP7[2:0]	FPT7 chattering filter time select	SCTP7[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
		0x0	None						
D3	-	reserved	-	-	-	0 when being read.			
D2-0	SCTP6[2:0]	FPT6 chattering filter time select	SCTP6[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
			0x3	4/fPCLK2					
			0x2	2/fPCLK2					
			0x1	1/fPCLK2					
			0x0	None					
FPT8-9 Interrupt Chattering Filter Control Register (GPIO_FPT89_ CHAT)	0x300348 (8 bits)	D7	-	reserved	-	-	-	0 when being read.	
		D6-4	SCTP9[2:0]	FPT9 chattering filter time select	SCTP9[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
					0x3	4/fPCLK2			
					0x2	2/fPCLK2			
					0x1	1/fPCLK2			
		0x0	None						
D3	-	reserved	-	-	-	0 when being read.			
D2-0	SCTP8[2:0]	FPT8 chattering filter time select	SCTP8[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
			0x3	4/fPCLK2					
			0x2	2/fPCLK2					
			0x1	1/fPCLK2					
			0x0	None					

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FPTA-B Interrupt Chattering Filter Control Register (GPIO_FPTAB_ CHAT)	0x300349 (8 bits)	D7	-	reserved	-	-	-	0 when being read.	
		D6-4	SCTPB[2:0]	FPTB chattering filter time select	SCTPB[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								
D3	-	reserved	-	-	-	-	0 when being read.		
D2-0	SCTPA[2:0]	FPTA chattering filter time select	SCTPA[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								
FPTC-D Interrupt Chattering Filter Control Register (GPIO_FPTCD_ CHAT)	0x30034a (8 bits)	D7	-	reserved	-	-	-	0 when being read.	
		D6-4	SCTPD[2:0]	FPTD chattering filter time select	SCTPD[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								
D3	-	reserved	-	-	-	-	0 when being read.		
D2-0	SCTPC[2:0]	FPTC chattering filter time select	SCTPC[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								
FPTE-F Interrupt Chattering Filter Control Register (GPIO_FPTEF_ CHAT)	0x30034b (8 bits)	D7	-	reserved	-	-	-	0 when being read.	
		D6-4	SCTPF[2:0]	FPTE chattering filter time select	SCTPF[2:0]	Filter sampling time	0x0	R/W	
					0x7	64/fPCLK2			
					0x6	32/fPCLK2			
					0x5	16/fPCLK2			
					0x4	8/fPCLK2			
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								
D3	-	reserved	-	-	-	-	0 when being read.		
D2-0	SCTPE[2:0]	FPTE chattering filter time select	SCTPE[2:0]	Filter sampling time	0x0	R/W			
			0x7	64/fPCLK2					
			0x6	32/fPCLK2					
			0x5	16/fPCLK2					
			0x4	8/fPCLK2					
0x3	4/fPCLK2								
0x2	2/fPCLK2								
0x1	1/fPCLK2								
0x0	None								

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
Port DMA Trigger Source Select Register (GPIO_DMA)	0x30034c (8 bits)	D7-4	-	reserved	-	-	-	-	0 when being read.	
		D3-0	SPTRG[3:0]	Port DMA trigger source select	SPTRG[3:0]	Trigger source	0x0	R/W		
						0xf	FPTF			
						0xe	FPTF			
						0xd	FPTD			
						0xc	FPTC			
						0xb	FPTB			
						0xa	FPTA			
						0x9	FPT9			
						0x8	FPT8			
						0x7	FPT7			
						0x6	FPT6			
						0x5	FPT5			
						0x4	FPT4			
						0x3	FPT3			
				0x2	FPT2					
				0x1	FPT1					
				0x0	FPT0					
P0[3:0] Port Function Select Register (PMUX_P0_03)	0x300800 (8 bits)	D7-6	CFP03[1:0]	P03 port function select	CFP03[1:0]	Function	0x0	R/W	Write-protected	
					0x3	REMC_I				
					0x2	#SRDY1				
					0x1	USI_CK				
					0x0	P03				
		D5-4	CFP02[1:0]	P02 port function select	CFP02[1:0]	Function	0x0	R/W		
					0x3	REMC_O				
					0x2	SCLK1				
					0x1	USI_CS				
					0x0	P02				
		D3-2	CFP01[1:0]	P01 port function select	CFP01[1:0]	Function	0x0	R/W		
					0x3	#NAND_RD				
					0x2	SOUT1				
					0x1	USI_DO				
					0x0	P01				
		D1-0	CFP00[1:0]	P00 port function select	CFP00[1:0]	Function	0x0	R/W		
0x3	#NAND_WR									
0x2	SIN1									
0x1	USI_DI									
0x0	P00									
P0[7:4] Port Function Select Register (PMUX_P0_47)	0x300801 (8 bits)	D7-6	CFP07[1:0]	P07 port function select	CFP07[1:0]	Function	0x0	R/W	Write-protected	
					0x3	PWM_L				
					0x2	I2S_MCLK				
					0x1	#SRDY0				
					0x0	P07				
		D5-4	CFP06[1:0]	P06 port function select	CFP06[1:0]	Function	0x0	R/W		
					0x3	PWM_H				
					0x2	I2S_SCLK				
					0x1	SCLK0				
					0x0	P06				
		D3-2	CFP05[1:0]	P05 port function select	CFP05[1:0]	Function	0x0	R/W		
					0x3	T16A_ATMB_0				
					0x2	I2S_WS				
					0x1	SOUT0				
					0x0	P05				
		D1-0	CFP04[1:0]	P04 port function select	CFP04[1:0]	Function	0x0	R/W		
0x3	T16A_ATMA_0									
0x2	I2S_SDO									
0x1	SIN0									
0x0	P04									
P1[3:0] Port Function Select Register (PMUX_P1_03)	0x300802 (8 bits)	D7-6	CFP13[1:0]	P13 port function select	CFP13[1:0]	Function	0x0	R/W	Write-protected	
					0x3	T16A_ATMB_1				
					0x2	FPDAT11				
					0x1	USIL_CK				
					0x0	P13				
		D5-4	CFP12[1:0]	P12 port function select	CFP12[1:0]	Function	0x0	R/W		
					0x3	T16A_ATMA_1				
					0x2	FPDAT10				
					0x1	USIL_CS				
					0x0	P12				
		D3-2	CFP11[1:0]	P11 port function select	CFP11[1:0]	Function	0x0	R/W		
					0x3	reserved				
					0x2	FPDAT9				
					0x1	USIL_DO				
					0x0	P11				
		D1-0	CFP10[1:0]	P10 port function select	CFP10[1:0]	Function	0x0	R/W		
0x3	reserved									
0x2	FPDAT8									
0x1	USIL_DI									
0x0	P10									

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks					
P1[7:4] Port Function Select Register (PMUX_P1_47)	0x300803 (8 bits)	D7-6	CFP17[1:0]	P17 port function select	CFP17[1:0]	Function	0x0	R/W	Write-protected				
					0x3	reserved							
					0x2	FPDAT15							
					0x1	P17							
		D5-4	CFP16[1:0]	P16 port function select	CFP16[1:0]	Function	0x0	R/W					
					0x3	reserved							
					0x2	FPDAT14							
					0x1	P16							
		D3-2	CFP15[1:0]	P15 port function select	CFP15[1:0]	Function	0x0	R/W					
					0x3	reserved							
					0x2	FPDAT13							
					0x1	P15							
D1-0	CFP14[1:0]	P14 port function select	CFP14[1:0]	Function	0x0	R/W							
			0x3	CMU_CLK									
			0x2	FPDAT12									
			0x1	FPDAT19									
P2[1:0] Port Function Select Register (PMUX_P2_01)	0x300804 (8 bits)	D7-4	-	reserved	-	-	-	-	0 when being read.				
					D3-2	CFP21[1:0]				P21 port function select	CFP21[1:0]	Function	0x0
		0x3	reserved										
		0x2	reserved										
		0x1	P21										
		D1-0	CFP20[1:0]	P20 port function select	CFP20[1:0]	Function	0x0	R/W					
					0x3	reserved							
					0x2	reserved							
					0x1	P20							
		P3[3:0] Port Function Select Register (PMUX_P3_03)	0x300806 (8 bits)	D7-6	CFP33[1:0]	P33 port function select	CFP33[1:0]	Function		0x0	R/W	Write-protected	
							0x3	REMC_I					
							0x2	TFT_CTL3					
0x1	reserved												
D5-4	CFP32[1:0]			P32 port function select	CFP32[1:0]	Function	0x0	R/W					
					0x3	REMC_O							
					0x2	TFT_CTL2							
					0x1	reserved							
D3-2	CFP31[1:0]			P31 port function select	CFP31[1:0]	Function	0x0	R/W					
					0x3	T16A_ATMB_0							
					0x2	TFT_CTL1							
					0x1	reserved							
D1-0	CFP30[1:0]	P30 port function select	CFP30[1:0]	Function	0x0	R/W							
			0x3	T16A_ATMA_0									
			0x2	TFT_CTL0									
			0x1	reserved									
P3[6:4] Port Function Select Register (PMUX_P3_46)	0x300807 (8 bits)	D7-6	-	reserved	-	-	-	-	0 when being read.				
					D5-4	CFP36[1:0]				P36 port function select	CFP36[1:0]	Function	0x0
		0x3	reserved										
		0x2	reserved										
		0x1	P36										
		D3-2	CFP35[1:0]	P35 port function select	CFP35[1:0]	Function	0x0	R/W					
					0x3	reserved							
					0x2	reserved							
					0x1	P35							
		D1-0	CFP34[1:0]	P34 port function select	CFP34[1:0]	Function	0x0	R/W					
					0x3	reserved							
					0x2	reserved							
0x1	P34												
					0x0								

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
P4[2:0] Port Function Select Register (PMUX_P4_02)	0x300808 (8 bits)	D7-6	-	reserved	-	-	-	0 when being read.			
		D5-4	CFP42[1:0]	P42 port function select	CFP42[1:0]	Function	0x0	R/W	Write-protected		
					0x3 reserved	FPDAT16					
				0x2 P42							
				0x1 A23							
				0x0							
D3-2	CFP41[1:0]	P41 port function select	CFP41[1:0]	Function	0x0	R/W					
			0x3 #NAND_WR	FPDAT17							
			0x2 P41								
			0x1 A22								
			0x0								
D1-0	CFP40[1:0]	P40 port function select	CFP40[1:0]	Function	0x0	R/W					
			0x3 #NAND_RD	FPDAT18							
			0x2 P40								
			0x1 A21								
			0x0								
P5[3:0] Port Function Select Register (PMUX_P5_03)	0x30080a (8 bits)	D7-6	CFP53[1:0]	P53 port function select	CFP53[1:0]	Function	0x0		R/W	Write-protected	
					0x3 reserved	reserved					
				0x2 reserved							
				0x1 P53							
				0x0 #CE10							
		D5-4	CFP52[1:0]	P52 port function select	CFP52[1:0]	Function	0x0	R/W			
					0x3 reserved	#CE5					
					0x2 #CE5						
			0x1 P52								
			0x0 #CE9								
D3-2	CFP51[1:0]	P51 port function select	CFP51[1:0]	Function	0x0	R/W					
			0x3 reserved	#CE4							
			0x2 #CE4	P51							
			0x1 P51								
			0x0 #CE8								
D1-0	CFP50[1:0]	P50 port function select	CFP50[1:0]	Function	0x0	R/W					
			0x3 reserved	#SDCS							
			0x2 #SDCS	P50							
			0x1 P50								
			0x0 #CE7								
P5[6:4] Port Function Select Register (PMUX_P5_46)	0x30080b (8 bits)	D7-6	-	reserved	-	-	-		0 when being read.		
		D5-4	CFP56[1:0]	P56 port function select	CFP56[1:0]	Function	0x0		R/W	Write-protected	
					0x3 reserved	reserved					
				0x2 reserved							
				0x1 P56							
				0x0 #WRH/#BSH							
D3-2	CFP55[1:0]	P55 port function select	CFP55[1:0]	Function	0x0	R/W					
			0x3 reserved	reserved							
			0x2 reserved								
			0x1 P55								
			0x0 #WRL								
D1-0	CFP54[1:0]	P54 port function select	CFP54[1:0]	Function	0x0	R/W					
			0x3 reserved	reserved							
			0x2 reserved								
			0x1 P54								
			0x0 #RD								
P60 Port Function Select Register (PMUX_P6_0)	0x30080c (8 bits)	D7-2	-	reserved	-	-	-		0 when being read.		
		D1-0	CFP60[1:0]	P60 port function select	CFP60[1:0]	Function	0x0	R/W	Write-protected		
			0x3 #WDT_NMI								
			0x2 WDT_CLK								
			0x1 #WAIT								
			0x0 P60								
P7[3:0] Port Function Select Register (PMUX_P7_03)	0x30080e (8 bits)	D7-6	CFP73[1:0]	P73 port function select	CFP73[1:0]	Function	0x0	R/W	Write-protected		
					0x3 reserved	reserved					
				0x2 reserved							
				0x1 AIN3							
				0x0 P73							
		D5-4	CFP72[1:0]	P72 port function select	CFP72[1:0]	Function	0x0	R/W			
					0x3 reserved	PWM_EXCL					
					0x2 PWM_EXCL						
			0x1 AIN2								
			0x0 P72								
D3-2	CFP71[1:0]	P71 port function select	CFP71[1:0]	Function	0x0	R/W					
			0x3 reserved	T16A_EXCL_1							
			0x2 T16A_EXCL_1								
			0x1 AIN1								
			0x0 P71								
D1-0	CFP70[1:0]	P70 port function select	CFP70[1:0]	Function	0x0	R/W					
			0x3 reserved	T16A_EXCL_0							
			0x2 T16A_EXCL_0								
			0x1 AIN0								
			0x0 P70								

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
P7[5:4] Port Function Select Register (PMUX_P7_45)	0x30080f (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3-2	CFP75[1:0]	P75 port function select	CFP75[1:0]	Function	0x0	R/W	Write-protected
					0x3 #ADTRIG				
		0x2 #WAIT							
0x1 AIN5									
0x0 P75									
D1-0	CFP74[1:0]	P74 port function select	CFP74[1:0]	Function	0x0	R/W			
			0x3 reserved						
			0x2 reserved						
			0x1 AIN4						
0x0 P74									
P8[3:0] Port Function Select Register (PMUX_P8_03)	0x300810 (8 bits)	D7-6	CFP83[1:0]	P83 port function select	CFP83[1:0]	Function	0x0	R/W	Write-protected
					0x3 reserved				
					0x2 USIL_DO				
					0x1 FPDRDY				
		0x0 P83							
		D5-4	CFP82[1:0]	P82 port function select	CFP82[1:0]	Function	0x0	R/W	
					0x3 reserved				
					0x2 USIL_DI				
					0x1 FPSHIFT				
		0x0 P82							
		D3-2	CFP81[1:0]	P81 port function select	CFP81[1:0]	Function	0x0	R/W	
					0x3 reserved				
0x2 USIL_CK									
0x1 FPLINE									
0x0 P81									
D1-0	CFP80[1:0]	P80 port function select	CFP80[1:0]	Function	0x0	R/W			
			0x3 reserved						
			0x2 USIL_CS						
			0x1 FPFRAME						
0x0 P80									
P9[3:0] Port Function Select Register (PMUX_P9_03)	0x300812 (8 bits)	D7-6	CFP93[1:0]	P93 port function select	CFP93[1:0]	Function	0x0	R/W	Write-protected
					0x3 #SRDY0				
					0x2 LCD_D3				
					0x1 FPDAT3				
		0x0 P93							
		D5-4	CFP92[1:0]	P92 port function select	CFP92[1:0]	Function	0x0	R/W	
					0x3 SCLK0				
					0x2 LCD_D2				
					0x1 FPDAT2				
		0x0 P92							
		D3-2	CFP91[1:0]	P91 port function select	CFP91[1:0]	Function	0x0	R/W	
					0x3 SOUT0				
0x2 LCD_D1									
0x1 FPDAT1									
0x0 P91									
D1-0	CFP90[1:0]	P90 port function select	CFP90[1:0]	Function	0x0	R/W			
			0x3 SIN0						
			0x2 LCD_D0						
			0x1 FPDAT0						
0x0 P90									
P9[7:4] Port Function Select Register (PMUX_P9_47)	0x300813 (8 bits)	D7-6	CFP97[1:0]	P97 port function select	CFP97[1:0]	Function	0x0	R/W	Write-protected
					0x3 reserved				
					0x2 LCD_D7				
					0x1 FPDAT7				
		0x0 P97							
		D5-4	CFP96[1:0]	P96 port function select	CFP96[1:0]	Function	0x0	R/W	
					0x3 reserved				
					0x2 LCD_D6				
					0x1 FPDAT6				
		0x0 P96							
		D3-2	CFP95[1:0]	P95 port function select	CFP95[1:0]	Function	0x0	R/W	
					0x3 reserved				
0x2 LCD_D5									
0x1 FPDAT5									
0x0 P95									
D1-0	CFP94[1:0]	P94 port function select	CFP94[1:0]	Function	0x0	R/W			
			0x3 reserved						
			0x2 LCD_D4						
			0x1 FPDAT4						
0x0 P94									

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
PA[3:0] Port Function Select Register (PMUX_PA_03)	0x300814 (8 bits)	D7-6	CFPA3[1:0]	PA3 port function select	CFPA3[1:0]	Function	0x0	R/W	Write-protected		
					0x3	FPDAT23					
					0x2	FPDAT19					
					0x1	#SRDY1					
		0x0	PA3								
		D5-4	CFPA2[1:0]	PA2 port function select	CFPA2[1:0]	Function	0x0	R/W			
					0x3	FPDAT22					
					0x2	FPDAT18					
					0x1	SCLK1					
		0x0	PA2								
		D3-2	CFPA1[1:0]	PA1 port function select	CFPA1[1:0]	Function	0x0	R/W			
					0x3	FPDAT21					
0x2	FPDAT17										
0x1	SOUT1										
0x0	PA1										
D1-0	CFPA0[1:0]	PA0 port function select	CFPA0[1:0]	Function	0x0	R/W					
			0x3	FPDAT20							
			0x2	FPDAT16							
			0x1	SIN1							
0x0	PA0										
PA[6:4] Port Function Select Register (PMUX_PA_46)	0x300815 (8 bits)	D7-6	-	reserved	-	-	0x0	R/W	0 when being read.		
					-	-					
		D5-4	CFPA6[1:0]	PA6 port function select	CFPA6[1:0]	Function	0x0	R/W			
					0x3	reserved					
					0x2	reserved					
					0x1	#ADTRIG					
		0x0	PA6								
		D3-2	CFPA5[1:0]	PA5 port function select	CFPA5[1:0]	Function	0x0	R/W			
					0x3	REMC_I					
					0x2	T16A_ATMB_1					
					0x1	PA5					
		0x0	A25								
D1-0	CFPA4[1:0]	PA4 port function select	CFPA4[1:0]	Function	0x0	R/W					
			0x3	REMC_O							
			0x2	T16A_ATMA_1							
			0x1	PA4							
0x0	A24										
PB[3:0] Port Function Select Register (PMUX_PB_03)	0x300816 (8 bits)	D7-6	CFPB3[1:0]	PB3 port function select	CFPB3[1:0]	Function	0x0	R/W	Write-protected		
					0x3	PWM_L					
					0x2	I2S_MCLK					
					0x1	FPDAT11					
		0x0	PB3								
		D5-4	CFPB2[1:0]	PB2 port function select	CFPB2[1:0]	Function	0x0	R/W			
					0x3	PWM_H					
					0x2	I2S_SCLK					
					0x1	FPDAT10					
		0x0	PB2								
		D3-2	CFPB1[1:0]	PB1 port function select	CFPB1[1:0]	Function	0x0	R/W			
					0x3	reserved					
0x2	I2S_WS										
0x1	FPDAT9										
0x0	PB1										
D1-0	CFPB0[1:0]	PB0 port function select	CFPB0[1:0]	Function	0x0	R/W					
			0x3	reserved							
			0x2	I2S_SDO							
			0x1	FPDAT8							
0x0	PB0										
PB[7:4] Port Function Select Register (PMUX_PB_47)	0x300817 (8 bits)	D7-6	CFPB7[1:0]	PB7 port function select	CFPB7[1:0]	Function	0x0	R/W	Write-protected		
					0x3	FPDAT23					
					0x2	reserved					
					0x1	FPDAT15					
		0x0	PB7								
		D5-4	CFPB6[1:0]	PB6 port function select	CFPB6[1:0]	Function	0x0	R/W			
					0x3	FPDAT22					
					0x2	reserved					
					0x1	FPDAT14					
		0x0	PB6								
		D3-2	CFPB5[1:0]	PB5 port function select	CFPB5[1:0]	Function	0x0	R/W			
					0x3	FPDAT21					
0x2	reserved										
0x1	FPDAT13										
0x0	PB5										
D1-0	CFPB4[1:0]	PB4 port function select	CFPB4[1:0]	Function	0x0	R/W					
			0x3	FPDAT20							
			0x2	reserved							
			0x1	FPDAT12							
0x0	PB4										

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PC[3:0] Port Function Select Register (PMUX_PC_03)	0x300818 (8 bits)	D7-6	CFPC3[1:0]	PC3 port function select	CFPC3[1:0]	Function	0x0	R/W	Write-protected
					0x3 reserved	0x2 reserved			
		D5-4	CFPC2[1:0]	PC2 port function select	CFPC2[1:0]	Function	0x0	R/W	
					0x3 reserved	0x2 reserved			
	D3-2	CFPC1[1:0]	PC1 port function select	CFPC1[1:0]	Function	0x0	R/W		
				0x3 reserved	0x2 reserved			0x1 PC1	
	D1-0	CFPC0[1:0]	PC0 port function select	CFPC0[1:0]	Function	0x0	R/W		
				0x3 reserved	0x2 reserved			0x1 PC0	
PC[7:4] Port Function Select Register (PMUX_PC_47)	0x300819 (8 bits)	D7-6	CFPC7[1:0]	PC7 port function select	CFPC7[1:0]	Function	0x0	R/W	Write-protected
					0x3 reserved	0x2 reserved			
		D5-4	CFPC6[1:0]	PC6 port function select	CFPC6[1:0]	Function	0x0	R/W	
					0x3 reserved	0x2 reserved			
	D3-2	CFPC5[1:0]	PC5 port function select	CFPC5[1:0]	Function	0x0	R/W		
				0x3 reserved	0x2 reserved			0x1 PC5	
	D1-0	CFPC4[1:0]	PC4 port function select	CFPC4[1:0]	Function	0x0	R/W		
				0x3 reserved	0x2 reserved			0x1 PC4	
Port Noise Filter Control Register (GPIO_FILTER)	0x30083e (8 bits)	D7-1	-	reserved	-	-	-	0 when being read.	
		D0	ANFEN	Input port noise filter enable	1 Enable 0 Disable	0	R/W	Write-protected	
GPIO/PMUX Write Protect Register (GPIO_PROTECT)	0x30083f (8 bits)	D7-0	PPROT[7:0]	GPIO/PMUX register protect flag	Writing 10010110 (0x96) removes the write protection of the GPIO registers (0x300320-0x30032c and 0x30083e) and PMUX registers (0x300800-0x300819). Writing another value set the write protection.	0x0	R/W		

0x300400-0x300472

USI

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Global Configuration Register (USI_GCFG)	0x300400 (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.
		D3	LSBFST	MSB/LSB first mode select	1 MSB first 0 LSB first	0	R/W	
		D2-0	USIMOD [2:0]	Interface mode configuration	USIMOD[2:0]	I/F mode	0x0	R/W
					0x7-0x6 reserved	0x5 I ² C slave		
USI Transmit Data Buffer Register (USI_TD)	0x300401 (8 bits)	D7-0	TD[7:0]	USI transmit data buffer TD7 = MSB TD0 = LSB	0x0 to 0xff	0x0	R/W	
USI Receive Data Buffer Register (USI_RD)	0x300402 (8 bits)	D7-0	RD[7:0]	USI receive data buffer RD7 = MSB RD0 = LSB	0x0 to 0xff	0x0	R	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
USI UART Mode Configuration Register (USI_UCFG)	0x300440 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.		
		D3	UCHLN	Character length select	1 8 bits	0 7 bits	0	R/W		
		D2	USTPB	Stop bit select	1 2 bits	0 1 bit	0	R/W		
		D1	UPMD	Parity mode select	1 Even	0 Odd	0	R/W		
		D0	UPREN	Parity enable	1 With parity	0 No parity	0	R/W		
USI UART Mode Interrupt Enable Register (USI_UIE)	0x300441 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.		
		D2	UEIE	Receive error interrupt enable	1 Enable	0 Disable	0	R/W		
		D1	URDIE	Receive buffer full interrupt enable	1 Enable	0 Disable	0	R/W		
		D0	UTDIE	Transmit buffer empty int. enable	1 Enable	0 Disable	0	R/W		
USI UART Mode Interrupt Flag Register (USI_UIF)	0x300442 (8 bits)	D7	–	reserved	–	–	–	0 when being read.		
		D6	URBSY	Receive busy flag	1 Busy	0 Idle	0	R		
		D5	UTBSY	Transmit busy flag	1 Busy	0 Idle	0	R		
		D4	UPEIF	Parity error flag	1 Error	0 Normal	0	R/W	Reset by writing 1.	
		D3	USEIF	Framing error flag	1 Error	0 Normal	0	R/W		
		D2	UOEIF	Overrun error flag	1 Error	0 Normal	0	R/W		
		D1	URDIF	Receive buffer full flag	1 Full	0 Not full	0	R/W		
		D0	UTDIF	Transmit buffer empty flag	1 Empty	0 Not empty	0	R/W		
USI SPI Master/Slave Mode Configuration Register (USI_SCFG)	0x300450 (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.		
		D5	SCMD	Command bit (for 9-bit data)	1 High	0 Low	0	R/W		
		D4	SCHLN	Character length select	1 9 bits	0 8 bits	0	R/W		
		D3	SCPFA	Clock phase select	1 Phase 1	0 Phase 0	0	R/W		
		D2	SCPOL	Clock polarity select	1 Active L	0 Active H	0	R/W		
		D1	–	reserved	–	–	–	–	Do not set to 1.	
		D0	SFSTMOD	Fast mode select	1 Fast	0 Normal	0	R/W		
USI SPI Master/Slave Mode Interrupt Enable Register (USI_SIE)	0x300451 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.		
		D2	SEIE	Receive error interrupt enable	1 Enable	0 Disable	0	R/W		
		D1	SRDIE	Receive buffer full interrupt enable	1 Enable	0 Disable	0	R/W		
		D0	STDIE	Transmit buffer empty int. enable	1 Enable	0 Disable	0	R/W		
USI SPI Master/Slave Mode Interrupt Flag Register (USI_SIF)	0x300452 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.		
		D3	SSIF	Transfer busy flag (master) ss signal low flag (slave)	1 Busy	0 Idle	0	R	Reset by writing 1.	
					1 ss = H	0 ss = L				
		D2	SEIF	Overrun error flag	1 Error	0 Normal	0	R/W		
		D1	SRDIF	Receive buffer full flag	1 Full	0 Not full	0	R/W		
		D0	STDIF	Transmit buffer empty flag	1 Empty	0 Not empty	0	R/W		
USI I ² C Master Mode Trigger Register (USI_IMTG)	0x300460 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.		
		D4	IMTG	I ² C master operation trigger	1 Trigger	0 Ignored	0	W	0 when being read.	
					1 Waiting	0 Finished				
		D3	–	reserved	–	–	–	–		
		D2-0	IMTGMOD [2:0]	I ² C master trigger mode select	IMTGMOD[2:0]	Trigger mode	0x0	R/W		
0x7	reserved									
0x6	Receive ACK/NAK									
0x5	Transmit NAK									
0x4	Transmit ACK									
0x3	Receive data									
0x2	Transmit data									
0x1	Stop condition									
0x0	Start condition									
USI I ² C Master Mode Interrupt Enable Register (USI_IMIE)	0x300461 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.		
		D1	IMEIE	Receive error interrupt enable	1 Enable	0 Disable	0	R/W		
		D0	IMIE	Operation completion int. enable	1 Enable	0 Disable	0	R/W		
USI I ² C Master Mode Interrupt Flag Register (USI_IMIF)	0x300462 (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.		
		D5	IMBSY	I ² C master busy flag	1 Busy	0 Standby	0	R		
		D4-2	IMSTA[2:0]	I ² C master status	IMSTA[2:0]	Status	0x0	R		
					0x7	reserved				
					0x6	NAK received				
					0x5	ACK received				
					0x4	ACK/NAK sent				
					0x3	End of Rx data				
					0x2	End of Tx data				
					0x1	Stop generated				
0x0	Start generated									
D1	IMEIF	Overrun error flag	1 Error	0 Normal	0	R/W	Reset by writing 1.			
D0	IMIF	Operation completion flag	1 Completed	0 Not completed	0	R/W				

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USI I ² C Slave Mode Trigger Register (USI_ISTG)	0x300470 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4	ISTG	I ² C slave operation trigger	1 Trigger 0 Ignored	0	W		
						1 Waiting 0 Finished		R	
		D3	–	reserved	–	–	–	–	0 when being read.
		D2-0	ISTGMOD [2:0]	I ² C slave trigger mode select	ISTGMOD[2:0] Trigger mode	0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data/ Detect stop 0x2 Transmit data 0x1 reserved 0x0 Wait for start	0x0	R/W	
USI I ² C Slave Mode Interrupt Enable Register (USI_ISIE)	0x300471 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.	
		D1	ISEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W		
		D0	ISIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W		
USI I ² C Slave Mode Interrupt Flag Register (USI_ISIF)	0x300472 (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.	
		D5	ISBSY	I ² C slave busy flag	1 Busy 0 Standby	0	R		
		D4-2	ISSTA[2:0]	I ² C slave status	ISSTA[2:0] Status	0x0 reserved 0x6 NAK received 0x5 ACK received 0x4 ACK/NAK sent 0x3 End of Rx data 0x2 End of Tx data 0x1 Stop detected 0x0 Start detected	0x0	R	
		D1	ISEIF	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.	
		D0	ISIF	Operation completion flag	1 Completed 0 Not completed	0	R/W		

0x300600–0x30069f

USIL

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL Global Configuration Register (USIL_GCFG)	0x300600 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	LSBFST	MSB/LSB first mode select	1 MSB first 0 LSB first	0	R/W	
		D2-0	USILMOD [2:0]	Interface mode configuration	USILMOD[2:0] I/F mode	0x0 LCD Parallel 0x6 LCD SPI 0x5 I ² C slave 0x4 I ² C master 0x3 SPI slave 0x2 SPI master 0x1 UART 0x0 Software reset	0x0	R/W
USIL Transmit Data Buffer Register (USIL_TD)	0x300601 (8 bits)	D7-0	TD[7:0]	USIL transmit data buffer TD7 = MSB TD0 = LSB	0x0 to 0xff	0x0	R/W	
USIL Receive Data Buffer Register (USIL_RD)	0x300602 (8 bits)	D7-0	RD[7:0]	USIL receive data buffer RD7 = MSB RD0 = LSB	0x0 to 0xff	0x0	R	
USIL UART Mode Configuration Register (USIL_UCFG)	0x300640 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	UCHLN	Character length select	1 8 bits 0 7 bits	0	R/W	
		D2	USTPB	Stop bit select	1 2 bits 0 1 bit	0	R/W	
		D1	UPMD	Parity mode select	1 Even 0 Odd	0	R/W	
		D0	UPREN	Parity enable	1 With parity 0 No parity	0	R/W	
USIL UART Mode Interrupt Enable Register (USIL_UIE)	0x300641 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	UEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	URDIE	Receive buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	UTDIE	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
USIL UART Mode Interrupt Flag Register (USIL_UIF)	0x300642 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	URBSY	Receive busy flag	1 Busy 0 Idle	0	R	
		D5	UTBSY	Transmit busy flag	1 Busy 0 Idle	0	R	
		D4	UPEIF	Parity error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D3	USEIF	Framing error flag	1 Error 0 Normal	0	R/W	
		D2	UOEIF	Overrun error flag	1 Error 0 Normal	0	R/W	
		D1	URDIF	Receive buffer full flag	1 Full 0 Not full	0	R/W	
		D0	UTDIF	Transmit buffer empty flag	1 Empty 0 Not empty	0	R/W	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USIL SPI Master/Slave Mode Configuration Register (USIL_SCFG)	0x300650 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	SCPHA	Clock phase select	1 Phase 1 0 Phase 0	0	R/W	
		D2	SCPOL	Clock polarity select	1 Active L 0 Active H	0	R/W	
		D1	–	reserved	–	–	–	Do not set to 1.
	D0	SFSTMOD	Fast mode select	1 Fast 0 Normal	0	R/W		
USIL SPI Master/Slave Mode Interrupt Enable Register (USIL_SIE)	0x300651 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.
		D2	SEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SRDIE	Receive buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	STDIE	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
USIL SPI Master/Slave Mode Interrupt Flag Register (USIL_SIF)	0x300652 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.
		D3	SSIF	Transfer busy flag (master) ss signal low flag (slave)	1 Busy 0 Idle 1 ss = H 0 ss = L	0	R	
		D2	SEIF	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D1	SRDIF	Receive buffer full flag	1 Full 0 Not full	0	R/W	
		D0	STDIF	Transmit buffer empty flag	1 Empty 0 Not empty	0	R/W	
USIL I ² C Master Mode Trigger Register (USIL_IMTG)	0x300660 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	IMTG	I ² C master operation trigger	1 Trigger 0 Ignored 1 Waiting 0 Finished	0	R	
		D3	–	reserved	–	–	–	0 when being read.
		D2-0	IMTGMOD[2:0]	I ² C master trigger mode select	IMTGMOD[2:0] Trigger mode	0x0	R/W	
				0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 Stop condition 0x0 Start condition				
USIL I ² C Master Mode Interrupt Enable Register (USIL_IMIE)	0x300661 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	IMEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	IMIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W	
USIL I ² C Master Mode Interrupt Flag Register (USIL_IMIF)	0x300662 (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.
		D5	IMBSY	I ² C master busy flag	1 Busy 0 Standby	0	R	
		D4-2	IMSTA[2:0]	I ² C master status	IMSTA[2:0] Status	0x0	R	
					0x7 reserved 0x6 NAK received 0x5 ACK received 0x4 ACK/NAK sent 0x3 End of Rx data 0x2 End of Tx data 0x1 Stop generated 0x0 Start generated			
		D1	IMEIF	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D0	IMIF	Operation completion flag	1 Completed 0 Not completed	0	R/W	
USIL I ² C Slave Mode Trigger Register (USIL_ISTG)	0x300670 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.
		D4	ISTG	I ² C slave operation trigger	1 Trigger 0 Ignored 1 Waiting 0 Finished	0	W	
		D3	–	reserved	–	–	–	0 when being read.
		D2-0	ISTGMOD[2:0]	I ² C slave trigger mode select	ISTGMOD[2:0] Trigger mode	0x0	R/W	
			0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data/ Detect stop 0x2 Transmit data 0x1 reserved 0x0 Wait for start					
USIL I ² C Slave Mode Interrupt Enable Register (USIL_ISIE)	0x300671 (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.
		D1	ISEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	ISIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
USIL I ² C Slave Mode Interrupt Flag Register (USIL_ISIF)	0x300672 (8 bits)	D7-6	-	reserved	-	-	-	-	0 when being read.
		D5	ISBSY	I ² C slave busy flag	1 Busy	0 Standby	0	R	
		D4-2	ISSTA[2:0]	I ² C slave status	ISSTA[2:0] Status		0x0	R	
					0x7		reserved		
					0x6		NAK received		
0x5					ACK received				
0x4		ACK/NAK sent							
0x3		End of Rx data							
0x2		End of Tx data							
0x1		Stop detected							
0x0		Start detected							
D1	ISEIF	Overrun error flag	1 Error	0 Normal	0	R/W	Reset by writing 1.		
D0	ISIF	Operation completion flag	1 Completed	0 Not completed	0	R/W			
USIL LCD SPI Mode Configuration Register (USIL_LSCFG)	0x300680 (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3	LSCPHA	Clock phase select	1 Phase 1	0 Phase 0	0	R/W	
		D2	LSCPOL	Clock polarity select	1 Active L	0 Active H	0	R/W	
		D1	LSCMD	Command bit	1 High	0 Low	0	R/W	
D0	LSCMDEN	Command bit enable	1 Enable	0 Disable	0	R/W			
USIL LCD SPI Mode Interrupt Enable Register (USIL_LSIE)	0x300681 (8 bits)	D7-1	-	reserved	-	-	-	0 when being read.	
		D0	LSTDIE	Transmit buffer empty int. enable	1 Enable	0 Disable	0	R/W	
USIL LCD SPI Mode Interrupt Flag Register (USIL_LSIF)	0x300682 (8 bits)	D7-2	-	reserved	-	-	-	X when being read.	
		D1	LSBSY	Transfer busy flag	1 Busy	0 Idle	0	R	
		D0	LSTDIF	Transmit buffer empty flag	1 Empty	0 Not empty	0	R/W	Reset by writing 1.
USIL LCD SPI Mode Data Configuration Register (USIL_LSDFG)	0x30068f (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.	
		D3-2	LS18DFM[1:0]	LCD SPI 18-bit data format select	LS18DFM[1:0] Data format		0x0	R/W	
					0x3		Format 3		
					0x2		Format 2		
					0x1		Format 1		
0x0		Format 0							
D1-0	LSDMOD[1:0]	LCD SPI data mode select	LSDMOD[1:0] Data mode		0x0	R/W			
			0x3		24-bit mode				
			0x2		18-bit mode				
			0x1		16-bit mode				
0x0		8-bit mode							
USIL LCD Parallel I/F Mode Configuration Register (USIL_LPCFG)	0x300690 (8 bits)	D7-3	-	reserved	-	-	-	0 when being read.	
		D2	LPSRDEN	Successive read enable	1 Enable	0 Disable	0	R/W	
		D1	LPCMD	Command bit	1 High	0 Low	0	R/W	
		D0	LPRD	Read trigger	1 Trigger	0 Ignored	0	W	
1 Read cycle	0 Read finished					R			
USIL LCD Parallel I/F Mode Interrupt Enable Register (USIL_LPIE)	0x300691 (8 bits)	D7-2	-	reserved	-	-	-	0 when being read.	
		D1	LPRDIE	Read buffer full interrupt enable	1 Enable	0 Disable	0	R/W	
		D0	LPWRIE	Write buffer empty interrupt enable	1 Enable	0 Disable	0	R/W	
USIL LCD Parallel I/F Mode Interrupt Flag Register (USIL_LPIF)	0x300692 (8 bits)	D7-3	-	reserved	-	-	-	X when being read.	
		D2	LPBSY	Transfer busy flag	1 Busy	0 Idle	0	R	
		D1	LPRDIF	Read buffer full flag	1 Full	0 Not full	0	R/W	
		D0	LPWRIF	Write buffer empty flag	1 Empty	0 Not empty	0	R/W	Reset by writing 1.
USIL LCD Parallel I/F Mode Access Timing Register (USIL_LPAC)	0x30069f (8 bits)	D7-6	LPHD[1:0]	Hold cycle	LPHD[1:0] Hold cycle		0x0	R/W	
					0x3		4 cycles		
					0x2		3 cycles		
					0x1		2 cycles		
		0x0		1 cycle					
		D5-4	LPST[1:0]	Setup cycle	LPST[1:0] Setup cycle		0x0	R/W	
					0x3		4 cycles		
0x2					3 cycles				
0x1		2 cycles							
0x0		1 cycle							
D3-0	LPWT[3:0]	Wait cycle	LPWT[3:0] Wait cycle		0x0	R/W			
			0xf		15 cycles				
			0xe		14 cycles				
			:		:				
0x1		1 cycle							
0x0		0 cycles							

0x300700–0x30070f

FSIO Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
FSIO Ch.0 Transmit Data Register (FSIO_TXD0)	0x300700 (8 bits)	D7–0	TXD[7:0]	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	X	R/W	7-bit asynchronous mode does not use TXD7.			
FSIO Ch.0 Receive Data Register (FSIO_RXD0)	0x300701 (8 bits)	D7–0	RXD[7:0]	Receive data RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	X	R	7-bit asynchronous mode does not use RXD7 (fixed at 0).			
FSIO Ch.0 Status Register (FSIO_STATUS0)	0x300702 (8 bits)	D7–6	RXDNUM [1:0]	Receive FIFO data count	RXDNUM[1:0]	Number of data	0x0	R			
					0x3	4					
					0x2	3					
					0x1	2					
					0x0	1 or 0					
		D5	TEND	Transmit status flag	1	Busy	0	End/Idle	0	R	
		D4	FER	Framing error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
		D3	PER	Parity error flag	1	Error	0	Normal	0	R/W	
D2	OER	Overrun error flag	1	Error	0	Normal	0	R/W			
D1	TDBE	Transmit data buffer empty flag	1	Empty	0	Full	1	R			
D0	RDBF	Receive data buffer status flag	1	Contained	0	Not contained	0	R			
FSIO Ch.0 Control Register (FSIO_CTL0)	0x300703 (8 bits)	D7	TXEN	Transmit enable	1	Enable	0	Disable	0	R/W	
		D6	RXEN	Receive enable	1	Enable	0	Disable	0	R/W	
		D5	EPR	Parity enable	1	With parity	0	No parity	0	R/W	Valid only in asynchronous mode.
		D4	PMD	Parity mode select	1	Odd	0	Even	0	R/W	
		D3	STPB	Stop bit select	1	2 bits	0	1 bit	0	R/W	
		D2	SSCK	Input clock select	1	SCLK	0	Internal	0	R/W	
		D1–0	SMD[1:0]	Transfer mode select		SMD[1:0]	Transfer mode	0x0	R/W		
						0x3	8-bit async				
				0x2	7-bit async						
				0x1	Clk sync slave						
				0x0	Clk sync master						
FSIO Ch.0 IrDA Register (FSIO_IRDA0)	0x300704 (8 bits)	D7	SRDYCTL	#SRDY control	1	High mask	0	Normal	0	R/W	Writing is disabled when SIOADV = 0.
		D6–5	FIFOINT [1:0]	Receive buffer full interrupt timing		FIFOINT[1:0]	Receive level	0x0	R/W		
						0x3	4				
						0x2	3				
						0x1	2				
						0x0	1				
		D4	DIVMD	Async clock division ratio	1	1/8	0	1/16	0	R/W	
		D3	IRTL	IrDA I/F output logic inversion	1	Inverted	0	Direct	0	R/W	Valid only in async mode.
D2	IRRL	IrDA I/F input logic inversion	1	Inverted	0	Direct	0	R/W			
D1–0	IRMD[1:0]	Interface mode select		IRMD[1:0]	I/F mode	0x0	R/W				
				0x3	reserved						
				0x2	IrDA 1.0						
				0x1	reserved						
				0x0	General I/F						
FSIO Ch.0 Baud-rate Timer Control Register (FSIO_BRTRUN0)	0x300705 (8 bits)	D7–1	–	reserved		–	–	–	0 when being read.		
		D0	BRTRUN	Baud-rate timer run/stop control	1	Run	0	Stop	0	R/W	
FSIO Ch.0 Baud-rate Timer Reload Data L Register (FSIO_BRTRDL0)	0x300706 (8 bits)	D7–0	BRTRD[7:0]	Baud-rate timer reload data [7:0]	0x0 to 0xff (BRTRD[11:0] = 0x0 to 0xffff)	0x0	R/W				
FSIO Ch.0 Baud-rate Timer Reload Data H Register (FSIO_BRTRDH0)	0x300707 (8 bits)	D7–4	–	reserved		–	–	–	0 when being read.		
		D3–0	BRTRD [11:8]	Baud-rate timer reload data [11:8]		0x0 to 0xf (BRTRD[11:0] = 0x0 to 0xffff)	0x0	R/W			
FSIO Ch.0 Baud-rate Timer Count Data L Register (FSIO_BRTCDL0)	0x300708 (8 bits)	D7–0	BRTCD[7:0]	Baud-rate timer count data [7:0]	0x0 to 0xff (BRTCD[11:0] = 0x0 to 0xffff)	0x0	R				
FSIO Ch.0 Baud-rate Timer Count Data H Register (FSIO_BRTCDH0)	0x300709 (8 bits)	D7–4	–	reserved		–	–	–	0 when being read.		
		D3–0	BRTCD [11:8]	Baud-rate timer count data [11:8]		0x0 to 0xf (BRTCD[11:0] = 0x0 to 0xffff)	0x0	R			

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.0 Interrupt Flag Register (FSIO_INTF0)	0x30070a (8 bits)	D7-2	-	reserved	-	-	-	0 when being read.
		D1	TDBE_IF	Transmit data buffer empty int. flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0 R/W	Reset by writing 0.
		D0	RDBF_IF	Receive data buffer full int. flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0 R/W	
FSIO Ch.0 Interrupt Enable Register (FSIO_INTE0)	0x30070b (8 bits)	D7-3	-	reserved	-	-	-	0 when being read.
		D2	RERR_IE	Receive error interrupt enable	1 Enable	0 Disable	0 R/W	
		D1	TDBE_IE	Transmit data buffer empty interrupt enable	1 Enable	0 Disable	0 R/W	
FSIO Ch.0 STD/ ADV Mode Select Register (FSIO_ADV0)	0x30070f (8 bits)	D7-1	-	reserved	-	-	-	Writing 1 not allowed.
		D0	SIOADV	Standard/advanced mode select	1 Advanced mode	0 Standard mode	0 R/W	

0x300710-0x30071f

FSIO Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FSIO Ch.1 Transmit Data Register (FSIO_TXD1)	0x300710 (8 bits)	D7-0	TXD[7:0]	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	X	R/W	7-bit asynchronous mode does not use TXD7.
FSIO Ch.1 Receive Data Register (FSIO_RXD1)	0x300711 (8 bits)	D7-0	RXD[7:0]	Receive data RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	X	R	7-bit asynchronous mode does not use RXD7 (fixed at 0).
FSIO Ch.1 Status Register (FSIO_STATUS1)	0x300712 (8 bits)	D7-6	RXDNUM[1:0]	Receive FIFO data count	RXDNUM[1:0] Number of data	0x0	R	
					0x3 4			
					0x2 3			
					0x1 2			
					0x0 1 or 0			
		D5	TEND	Transmit status flag	1 Busy	0 End/Idle	0 R	
		D4	FER	Framing error flag	1 Error	0 Normal	0 R/W	Reset by writing 0.
		D3	PER	Parity error flag	1 Error	0 Normal	0 R/W	
D2	OER	Overrun error flag	1 Error	0 Normal	0 R/W			
D1	TDBE	Transmit data buffer empty flag	1 Empty	0 Full	1 R			
D0	RDBF	Receive data buffer status flag	1 Contained	0 Not contained	0 R			
FSIO Ch.1 Control Register (FSIO_CTL1)	0x300713 (8 bits)	D6	TXEN	Transmit enable	1 Enable	0 Disable	0 R/W	
		D7	RXEN	Receive enable	1 Enable	0 Disable	0 R/W	
		D5	EPR	Parity enable	1 With parity	0 No parity	0 R/W	Valid only in asynchronous mode.
		D4	PMD	Parity mode select	1 Odd	0 Even	0 R/W	
		D3	STPB	Stop bit select	1 2 bits	0 1 bit	0 R/W	
		D2	SSCK	Input clock select	1 SCLK	0 Internal	0 R/W	
		D1-0	SMD[1:0]	Transfer mode select	SMD[1:0] Transfer mode	0x0	R/W	
					0x3 8-bit async			
			0x2 7-bit async					
			0x1 Clk sync slave					
			0x0 Clk sync master					
FSIO Ch.1 IrDA Register (FSIO_IRDA1)	0x300714 (8 bits)	D7	SRDYCTL	#SRDY control	1 High mask	0 Normal	0 R/W	Writing is disabled when SIOADV = 0.
		D6-5	FIFOINT[1:0]	Receive buffer full interrupt timing	FIFOINT[1:0] Receive level	0x0	R/W	
					0x3 4			
					0x2 3			
					0x1 2			
					0x0 1			
		D4	DIVMD	Async clock division ratio	1 1/8	0 1/16	0 R/W	
		D3	IRTL	IrDA I/F output logic inversion	1 Inverted	0 Direct	0 R/W	Valid only in async mode.
D2	IRRL	IrDA I/F input logic inversion	1 Inverted	0 Direct	0 R/W			
D1-0	IRMD[1:0]	Interface mode select	IRMD[1:0] I/F mode	0x0	R/W			
			0x3 reserved					
			0x2 IrDA 1.0					
			0x1 reserved					
			0x0 General I/F					
FSIO Ch.1 Baud-rate Timer Control Register (FSIO_BRTRUN1)	0x300715 (8 bits)	D7-1	-	reserved	-	-	-	0 when being read.
		D0	BRTRUN	Baud-rate timer run/stop control	1 Run	0 Stop	0 R/W	
FSIO Ch.1 Baud-rate Timer Reload Data L Register (FSIO_BRTRDL1)	0x300716 (8 bits)	D7-0	BRTRD[7:0]	Baud-rate timer reload data [7:0]	0x0 to 0xff (BRTRD[11:0] = 0x0 to 0xffff)	0x0	R/W	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
FSIO Ch.1 Baud-rate Timer Reload Data H Register (FSIO_BRTRDH1)	0x300717 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.	
		D3-0	BRTRD [11:8]	Baud-rate timer reload data [11:8]	0x0 to 0xf (BRTRD[11:0] = 0x0 to 0xff)	0x0	R/W		
FSIO Ch.1 Baud-rate Timer Count Data L Register (FSIO_BRTCDL1)	0x300718 (8 bits)	D7-0	BRTCD [7:0]	Baud-rate timer count data [7:0]	0x0 to 0xff (BRTCD[11:0] = 0x0 to 0xff)	0x0	R		
FSIO Ch.1 Baud-rate Timer Count Data H Register (FSIO_BRTCDH1)	0x300719 (8 bits)	D7-4	–	reserved	–	–	–	0 when being read.	
		D3-0	BRTCD [11:8]	Baud-rate timer count data [11:8]	0x0 to 0xf (BRTCD[11:0] = 0x0 to 0xff)	0x0	R		
FSIO Ch.1 Interrupt Flag Register (FSIO_INTF1)	0x30071a (8 bits)	D7-2	–	reserved	–	–	–	0 when being read.	
		D1	TDBE_IF	Transmit data buffer empty int. flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 0.
		D0	RDBF_IF	Receive data buffer full int. flag			0	R/W	
FSIO Ch.1 Interrupt Enable Register (FSIO_INTE1)	0x30071b (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	RERR_IE	Receive error interrupt enable	1 Enable	0 Disable	0	R/W	
		D1	TDBE_IE	Transmit data buffer empty interrupt enable	1 Enable	0 Disable	0	R/W	
		D0	RDBF_IE	Receive data buffer full int. enable	1 Enable	0 Disable	0	R/W	
FSIO Ch.1 STD/ADV Mode Select Register (FSIO_ADV1)	0x30071f (8 bits)	D7-1	–	reserved	–	–	–	Writing 1 not allowed.	
		D0	SIOADV	Standard/advanced mode select	1 Advanced mode	0 Standard mode	0	R/W	

0x300a00–0x300a0f

Real-time Clock (RTC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
RTC Interrupt Status Register (RTC_INTSTAT)	0x300a00 (8 bits)	D7-1	–	reserved	–	–	–	0 when being read.	
		D0	RTCIRQ	Interrupt status	1 Occurred	0 Not occurred	X (0)	R/W	Reset by writing 1.
RTC Interrupt Mode Register (RTC_INTMODE)	0x300a01 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4-2	RTCT [2:0]	RTC interrupt cycle setup	RTCT[2:0] Cycle		X (0x1)	R/W	
					0x7 reserved				
					0x6 1/128 second				
					0x5 1/256 second				
			0x4 1/512 second						
			0x3 1 hour						
			0x2 1 minute						
			0x1 1 second						
			0x0 1/64 second						
		D1	RTCIMD	reserved	1	X (1)	R/W	Always set to 1.	
		D0	RTCEN	RTC interrupt enable	1 Enable	0 Disable	X (0)	R/W	
RTC Control 0 Register (RTC_CNTL0)	0x300a02 (8 bits)	D7-5	–	reserved	–	–	–	0 when being read.	
		D4	RTC24H	24H/12H mode select	1 24H	0 12H	X (0)	R/W	
		D3	–	reserved	–	–	–	–	0 when being read.
		D2	RTCADJ	30-second adjustment	1 Adjust	0 –	X (0)	R/W	
		D1	RTCSTP	Divider run/stop control	1 Stop	0 Run	X (0)	R/W	
		D0	RTCSTP	Software reset	1 Reset	0 –	X (0)	R/W	
RTC Control 1 Register (RTC_CNTL1)	0x300a03 (8 bits)	D7-3	–	reserved	–	–	–	0 when being read.	
		D2	RTC RDHLD	Read buffer enable	1 Enable	0 Disable	X (0)	R/W	
		D1	RTCBSY	Counter busy flag	1 Busy	0 R/W possible	X (0)	R	
		D0	RTC HLD	Counter hold control	1 Hold	0 Running	X (0)	R/W	
RTC Second Register (RTC_SEC)	0x300a04 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6-4	RTCSH [2:0]	RTC 10-second counter	0 to 5	X (*)	R/W		
		D3-0	RTCSL [3:0]	RTC 1-second counter	0 to 9	X (*)	R/W		
RTC Minute Register (RTC_MIN)	0x300a05 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6-4	RTCMIH [2:0]	RTC 10-minute counter	0 to 5	X (*)	R/W		
		D3-0	RTCMIL [3:0]	RTC 1-minute counter	0 to 9	X (*)	R/W		
RTC Hour Register (RTC_HOUR)	0x300a06 (8 bits)	D7	–	reserved	–	–	–	0 when being read.	
		D6	RTCAP	AM/PM indicator	1 PM	0 AM	X (*)	R/W	
		D5-4	RTC HH [1:0]	RTC 10-hour counter	0 to 2 or 0 to 1	X (*)	R/W		
		D3-0	RTC HL [3:0]	RTC 1-hour counter	0 to 9	X (*)	R/W		
RTC Day Register (RTC_DAY)	0x300a07 (8 bits)	D7-6	–	reserved	–	–	–	0 when being read.	
		D5-4	RTCDH [1:0]	RTC 10-day counter	0 to 3	X (*)	R/W		
		D3-0	RTCDL [3:0]	RTC 1-day counter	0 to 9	X (*)	R/W		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Month Register (RTC_MONTH)	0x300a08 (8 bits)	D7-5	-	reserved	-	-	-	0 when being read.
		D4	RTCMOH	RTC 10-month counter	0 to 1	X (*)	R/W	
		D3-0	RTCMOL[3:0]	RTC 1-month counter	0 to 9	X (*)	R/W	
RTC Year Register (RTC_YEAR)	0x300a09 (8 bits)	D7-4	RTCYH[3:0]	RTC 10-year counter	0 to 9	X (*)	R/W	
		D3-0	RTCYL[3:0]	RTC 1-year counter	0 to 9	X (*)	R/W	
RTC Days of Week Register (RTC_WEEK)	0x300a0a (8 bits)	D7-3	-	reserved	-	-	-	0 when being read.
		D2-0	RTCWK[2:0]	RTC days of week counter	RTCWK[2:0] Days of week	X (*)	R/W	
					0x7 - 0x6 Saturday 0x5 Friday 0x4 Thursday 0x3 Wednesday 0x2 Tuesday 0x1 Monday 0x0 Sunday			
RTC Wakeup Configuration Register (RTC_WAKEUP)	0x300a0f (8 bits)	D7-2	-	reserved	-	-	-	0 when being read.
		D1	WUP_CTL	WAKEUP control	1 Active 0 Inactive	X (0)	R/W	
		D0	WUP_POL	WAKEUP polarity select	1 Active low 0 Active high	X (0)	R/W	

Init.: () indicates the value set after a software reset (RTCST → 1 → 0) is performed.

* Software reset (RTCST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

0x300b00-0x300b0f

BBRAM

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
BBRAM byte 0 -byte 15 (BBRAM_0 -BBRAM_15)	0x300b00 0x300b0f (8 bits)	D7-0	-	BBRAM byte data	0x0 to 0xff	X	R/W	

0x300c00-0x300c9f

USB Function Controller (USB)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
MainIntStat (Main interrupt status)	0x300c00 (8 bits)	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)		
SIE_IntStat (SIE interrupt status)	0x300c01 (8 bits)	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)	0x300c02 (8 bits)	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)	0x300c03 (8 bits)	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)	0x300c04 (8 bits)	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)	0x300c07 (8 bits)	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)		

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks
EPaIntStat (EPa interrupt status)	0x300c08 (8 bits)	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)
EPbIntStat (EPb interrupt status)	0x300c09 (8 bits)	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)
EPcIntStat (EPc interrupt status)	0x300c0a (8 bits)	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)
EPdIntStat (EPd interrupt status)	0x300c0b (8 bits)	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)
MainIntEnb (Main interrupt enable)	0x300c10 (8 bits)	D7	EnSIE_IntStat	1	Enable	0	Disable	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	Enable	0	Disable	0 R/W
		D0	EnRcvEPOSETUP					0 R/W
SIE_IntEnb (SIE interrupt enable)	0x300c11 (8 bits)	D7	EnVBUS_Changed	1	Enable	0	Disable	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	Enable	0	Disable	0 R/W
EPrintEnb (EPr interrupt enable)	0x300c12 (8 bits)	D7–4	–		–	–	–	0 when being read.
		D3	EnEPdIntStat	1	Enable	0	Disable	0 R/W
		D2	EnEPcIntStat					0 R/W
		D1	EnEPbIntStat					0 R/W
		D0	EnEPaIntStat					0 R/W
DMA_IntEnb (DMA interrupt enable)	0x300c13 (8 bits)	D7–2	–		–	–	–	0 when being read.
		D1	EnDMA_CountUp	1	Enable	0	Disable	0 R/W
		D0	EnDMA_Cmp					0 R/W
FIFO_IntEnb (FIFO interrupt enable)	0x300c14 (8 bits)	D7	EnDescriptorCmp	1	Enable	0	Disable	0 R/W
		D6–2	–		–	–	0 when being read.	
		D1	EnFIFO_IN_Cmp	1	Enable	0	Disable	0 R/W
		D0	EnFIFO_OUT_Cmp					0 R/W
EP0IntEnb (EP0 interrupt enable)	0x300c17 (8 bits)	D7–6	–		–	–	–	0 when being read.
		D5	EnIN_TranACK	1	Enable	0	Disable	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		

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks	
EPaIntEnb (EPa interrupt enable)	0x300c18 (8 bits)	D7	-	-		-	-	0 when being read.	
		D6	EnOUT_ShortACK	1	Enable	0	Disable	0 R/W	
		D5	EnIN_TranACK					0 R/W	
		D4	EnOUT_TranACK					0 R/W	
		D3	EnIN_TranNAK					0 R/W	
		D2	EnOUT_TranNAK					0 R/W	
		D1	EnIN_TranErr					0 R/W	
		D0	EnOUT_TranErr					0 R/W	
EPbIntEnb (EPb interrupt enable)	0x300c19 (8 bits)	D7	-	-		-	-	0 when being read.	
		D6	EnOUT_ShortACK	1	Enable	0	Disable	0 R/W	
		D5	EnIN_TranACK					0 R/W	
		D4	EnOUT_TranACK					0 R/W	
		D3	EnIN_TranNAK					0 R/W	
		D2	EnOUT_TranNAK					0 R/W	
		D1	EnIN_TranErr					0 R/W	
		D0	EnOUT_TranErr					0 R/W	
EPcIntEnb (EPc interrupt enable)	0x300c1a (8 bits)	D7	-	-		-	-	0 when being read.	
		D6	EnOUT_ShortACK	1	Enable	0	Disable	0 R/W	
		D5	EnIN_TranACK					0 R/W	
		D4	EnOUT_TranACK					0 R/W	
		D3	EnIN_TranNAK					0 R/W	
		D2	EnOUT_TranNAK					0 R/W	
		D1	EnIN_TranErr					0 R/W	
		D0	EnOUT_TranErr					0 R/W	
EPdIntEnb (EPd interrupt enable)	0x300c1b (8 bits)	D7	-	-		-	-	0 when being read.	
		D6	EnOUT_ShortACK	1	Enable	0	Disable	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	
RevisionNum (Revision number)	0x300c20 (8 bits)	D7-0	RevisionNum[7:0]	Revision number (0x12)		0x12	R		
USB_Control (USB control)	0x300c21 (8 bits)	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	Deactivate USB	0 R/W	
USB_Status (USB status)	0x300c22 (8 bits)	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.
		D1-0	LineState[1:0]		LineState[1:0]		DP/DM	X R	
			0x3	SE1					
			0x2	K					
			0x1	J					
			0x0	SE0					
XcvrControl (Xcvr control)	0x300c23 (8 bits)	D7	RpuEnb	1	Enable pull-up	0	Disable pull-up	0 R/W	
		D6-2	-					-	0 when being read.
		D1-0	OpMode[1:0]		OpMode[1:0]		Operation mode	0x1 R/W	
			0x3	reserved					
			0x2	Disable bitstuffing and NRZI encoding					
			0x1	Non-driving					
			0x0	Normal operation					
USB_Test (USB test)	0x300c24 (8 bits)	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)	0x300c25 (8 bits)	D7	AllForceNAK	1	Set all ForceNAK	0	Do nothing	0 W	
		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)	0x300c26 (8 bits)	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	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Setting		Init.	R/W	Remarks			
FrameNumber_H (Frame number high)	0x300c2e (8 bits)	D7	FnlInvalid	1	Invalid frame number	0	Valid frame number	1	R		
		D6-3	-	-	-	-	-	-	-	0 when being read.	
		D2-0	FrameNumber[10:8]					Frame number high	0x0	R	
FrameNumber_L (Frame number low)	0x300c2f (8 bits)	D7-0	FrameNumber[7:0]				Frame number low	0x0	R		
EP0Setup_0 (EP0 setup 0)	0x300c30 (8 bits)	D7-0	EP0Setup_0[7:0]				Endpoint 0 setup data 0	0x0	R		
EP0Setup_1 (EP0 setup 1)	0x300c31 (8 bits)	D7-0	EP0Setup_1[7:0]				Endpoint 0 setup data 1	0x0	R		
EP0Setup_2 (EP0 setup 2)	0x300c32 (8 bits)	D7-0	EP0Setup_2[7:0]				Endpoint 0 setup data 2	0x0	R		
EP0Setup_3 (EP0 setup 3)	0x300c33 (8 bits)	D7-0	EP0Setup_3[7:0]				Endpoint 0 setup data 3	0x0	R		
EP0Setup_4 (EP0 setup 4)	0x300c34 (8 bits)	D7-0	EP0Setup_4[7:0]				Endpoint 0 setup data 4	0x0	R		
EP0Setup_5 (EP0 setup 5)	0x300c35 (8 bits)	D7-0	EP0Setup_5[7:0]				Endpoint 0 setup data 5	0x0	R		
EP0Setup_6 (EP0 setup 6)	0x300c36 (8 bits)	D7-0	EP0Setup_6[7:0]				Endpoint 0 setup data 6	0x0	R		
EP0Setup_7 (EP0 setup 7)	0x300c37 (8 bits)	D7-0	EP0Setup_7[7:0]				Endpoint 0 setup data 7	0x0	R		
USB_Address (USB address)	0x300c38 (8 bits)	D7	AutoSetAddress	1	Auto set address	0	Do nothing	0	R/W		
		D6-0	USB_Address[6:0]				USB address	0x0	R/W		
EP0Control (EP0 control)	0x300c39 (8 bits)	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)	0x300c3a (8 bits)	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)	0x300c3b (8 bits)	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)	0x300c3f (8 bits)	D7	-	-	-	-	-	-	-	0 when being read.	
		D6-3	EP0MaxSize[6:3]		Endpoint EP0 max packet size	0x1	R/W				
		D2-0	-	-	-	-	-	-	-	0 when being read.	
EPaControl (EPa control)	0x300c40 (8 bits)	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)	0x300c41 (8 bits)	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)	0x300c42 (8 bits)	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		

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks				
EPdControl (EPd control)	0x300c43 (8 bits)	D7	AutoForceNAK	1	Auto force NAK	0	Do nothing	0	R/W	0 when being read.	
		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		
		D2	ToggleClr	1	Clear toggle sequence bit	0	Do nothing	0	W		
		D1	ForceNAK	1	Force NAK	0	Do nothing	0	R/W		
		D0	ForceSTALL	1	Force STALL	0	Do nothing	0	R/W		
EPaMaxSize_H (EPa max packet size high)	0x300c50 (8 bits)	D7-2	-			-	-	-	-	0 when being read.	
		D1-0	EPaMaxSize[9:8]				Endpoint EPa max packet size	0x0	R/W		
EPaMaxSize_L (EPa max packet size low)	0x300c51 (8 bits)	D7-0	EPaMaxSize[7:0]				Endpoint EPa max packet size	0x0	R/W		
EPaConfig_0 (EPa configuration 0)	0x300c52 (8 bits)	D7	INxOUT	1	In	0	Out	0	R/W	0 when being read.	
		D6	ToggleMode	1	Always toggle	0	Normal toggle	0	R/W		
		D5	EnEndPoint	1	Enable endpoint	0	Disable endpoint	0	R/W		
		D4	-				-	-	-		-
		D3-0	EndPointNumber [3:0]				Endpoint number (0x1 to 0xf)	0x0	R/W		
EPaConfig_1 (EPa configuration 1)	0x300c53 (8 bits)	D7	ISO	1	ISO	0	Non-ISO	0	R/W	0 when being read.	
		D6	ISO_CRCmode	1	CRC mode	0	Normal ISO	0	R/W		
		D5-0	-				-	-	-		-
EPbMaxSize_H (EPb max packet size high)	0x300c54 (8 bits)	D7-2	-			-	-	-	-	0 when being read.	
		D1-0	EPbMaxSize[9:8]				Endpoint EPb max packet size	0x0	R/W		
EPbMaxSize_L (EPb max packet size low)	0x300c55 (8 bits)	D7-0	EPbMaxSize[7:0]				Endpoint EPb max packet size	0x0	R/W		
EPbConfig_0 (EPb configuration 0)	0x300c56 (8 bits)	D7	INxOUT	1	In	0	Out	0	R/W	0 when being read.	
		D6	ToggleMode	1	Always toggle	0	Normal toggle	0	R/W		
		D5	EnEndPoint	1	Enable endpoint	0	Disable endpoint	0	R/W		
		D4	-				-	-	-		-
		D3-0	EndPointNumber [3:0]				Endpoint number (0x1 to 0xf)	0x0	R/W		
EPbConfig_1 (EPb configuration 1)	0x300c57 (8 bits)	D7	ISO	1	ISO	0	Non-ISO	0	R/W	0 when being read.	
		D6	ISO_CRCmode	1	CRC mode	0	Normal ISO	0	R/W		
		D5-0	-				-	-	-		-
EPcMaxSize_H (EPc max packet size high)	0x300c58 (8 bits)	D7-2	-			-	-	-	-	0 when being read.	
		D1-0	EPcMaxSize[9:8]				Endpoint EPc max packet size	0x0	R/W		
EPcMaxSize_L (EPc max packet size low)	0x300c59 (8 bits)	D7-0	EPcMaxSize[7:0]				Endpoint EPc max packet size	0x0	R/W		
EPcConfig_0 (EPc configuration 0)	0x300c5a (8 bits)	D7	INxOUT	1	In	0	Out	0	R/W	0 when being read.	
		D6	ToggleMode	1	Always toggle	0	Normal toggle	0	R/W		
		D5	EnEndPoint	1	Enable endpoint	0	Disable endpoint	0	R/W		
		D4	-				-	-	-		-
		D3-0	EndPointNumber [3:0]				Endpoint number (0x1 to 0xf)	0x0	R/W		
EPcConfig_1 (EPc configuration 1)	0x300c5b (8 bits)	D7	ISO	1	ISO	0	Non-ISO	0	R/W	0 when being read.	
		D6	ISO_CRCmode	1	CRC mode	0	Normal ISO	0	R/W		
		D5-0	-				-	-	-		-
EPdMaxSize_H (EPd max packet size high)	0x300c5c (8 bits)	D7-2	-			-	-	-	-	0 when being read.	
		D1-0	EPdMaxSize[9:8]				Endpoint EPd max packet size	0x0	R/W		
EPdMaxSize_L (EPd max packet size low)	0x300c5d (8 bits)	D7-0	EPdMaxSize[7:0]				Endpoint EPd max packet size	0x0	R/W		
EPdConfig_0 (EPd configuration 0)	0x300c5e (8 bits)	D7	INxOUT	1	In	0	Out	0	R/W	0 when being read.	
		D6	ToggleMode	1	Always toggle	0	Normal toggle	0	R/W		
		D5	EnEndPoint	1	Enable endpoint	0	Disable endpoint	0	R/W		
		D4	-				-	-	-		-
		D3-0	EndPointNumber [3:0]				Endpoint number (0x1 to 0xf)	0x0	R/W		

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
EPdConfig_1 (EPd configuration 1)	0x300c5f (8 bits)	D7	ISO	1 ISO	0 Non-ISO	0	R/W	
		D6	ISO_CRCmode	1 CRC mode	0 Normal ISO	0	R/W	
		D5-0	—	—	—	—	—	—
EPaStartAdrs_H (EPa FIFO start address high)	0x300c70 (8 bits)	D7-4	—	—	—	—	—	0 when being read.
		D3-0	EPaStartAdrs[11:8]	—	Endpoint EPa start address	0x0	R/W	
EPaStartAdrs_L (EPa FIFO start address low)	0x300c71 (8 bits)	D7-2	EPaStartAdrs[7:2]	—	Endpoint EPa start address	0x0	R/W	
		D1-0	—	—	—	—	—	0 when being read.
EPbStartAdrs_H (EPb FIFO start address high)	0x300c72 (8 bits)	D7-4	—	—	—	—	—	0 when being read.
		D3-0	EPbStartAdrs[11:8]	—	Endpoint EPb start address	0x0	R/W	
EPbStartAdrs_L (EPb FIFO start address low)	0x300c73 (8 bits)	D7-2	EPbStartAdrs[7:2]	—	Endpoint EPb start address	0x0	R/W	
		D1-0	—	—	—	—	—	0 when being read.
EPcStartAdrs_H (EPc FIFO start address high)	0x300c74 (8 bits)	D7-4	—	—	—	—	—	0 when being read.
		D3-0	EPcStartAdrs[11:8]	—	Endpoint EPc start address	0x0	R/W	
EPcStartAdrs_L (EPc FIFO start address low)	0x300c75 (8 bits)	D7-2	EPcStartAdrs[7:2]	—	Endpoint EPc start address	0x0	R/W	
		D1-0	—	—	—	—	—	0 when being read.
EPdStartAdrs_H (EPd FIFO start address high)	0x300c76 (8 bits)	D7-4	—	—	—	—	—	0 when being read.
		D3-0	EPdStartAdrs[11:8]	—	Endpoint EPd start address	0x0	R/W	
EPdStartAdrs_L (EPd FIFO start address low)	0x300c77 (8 bits)	D7-2	EPdStartAdrs[7:2]	—	Endpoint EPd start address	0x0	R/W	
		D1-0	—	—	—	—	—	0 when being read.
CPU_JoinRd (CPU join FIFO read)	0x300c80 (8 bits)	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)	0x300c81 (8 bits)	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 (EPn FIFO access enable)	0x300c82 (8 bits)	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)	0x300c83 (8 bits)	D7-0	EPnFIFOData[7:0]	—	Endpoint n FIFO access from CPU	X	R/W	
EPnRdRemain_H (EPn FIFO read remain high)	0x300c84 (8 bits)	D7-4	—	—	—	—	—	0 when being read.
		D3-0	EPnRdRemain[11:8]	—	Endpoint n FIFO read remain	0x0	R	
EPnRdRemain_L (EPn FIFO read remain low)	0x300c85 (8 bits)	D7-0	EPnRdRemain[7:0]	—	Endpoint n FIFO read remain	0x0	R	
EPnWrRemain_H (EPn FIFO write remain high)	0x300c86 (8 bits)	D7-4	—	—	—	—	—	0 when being read.
		D3-0	EPnWrRemain[11:8]	—	Endpoint n FIFO write remain	0x0	R	
EPnWrRemain_L (EPn FIFO write remain low)	0x300c87 (8 bits)	D7-0	EPnWrRemain[7:0]	—	Endpoint n FIFO write remain	0x0	R	
DescAdrs_H (Descriptor address high)	0x300c88 (8 bits)	D7-4	—	—	—	—	—	0 when being read.
		D3-0	DescAdrs[11:8]	—	Descriptor address	0x0	R/W	
DescAdrs_L (Descriptor address low)	0x300c89 (8 bits)	D7-0	DescAdrs[7:0]	—	Descriptor address	0x0	R/W	
DescSize_H (Descriptor size high)	0x300c8a (8 bits)	D7-2	—	—	—	—	—	0 when being read.
		D1-0	DescSize[9:8]	—	Descriptor size	0x0	R/W	
DescSize_L (Descriptor size low)	0x300c8b (8 bits)	D7-0	DescSize[7:0]	—	Descriptor size	0x0	R/W	

Register name	Address	Bit	Name	Setting	Init.	R/W	Remarks	
DescDoor (Descriptor door)	0x300c8f (8 bits)	D7-0	DescMode[7:0]	Descriptor door		0x0	R/W	
DMA_FIFO_Control (DMA FIFO control)	0x300c90 (8 bits)	D7	FIFO_Running	1 FIFO is running	0 FIFO is not running	0	R	0 when being read.
		D6	AutoEnShort	1 Auto enable short packet	0 Do nothing	0	R/W	
		D5-0	-	-		-	-	
DMA_Join (DMA join FIFO)	0x300c91 (8 bits)	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)	0x300c92 (8 bits)	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	PDAACK	PDAACK 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	
DMA_Config_0 (DMA configuration 0)	0x300c94 (8 bits)	D7	ActivePort	1 Activate DMA port	0 Deactivate DMA port	0	R/W	0 when being read.
		D6-4	-	-		-	-	
		D3	PDREQ_Level	1 Active-low	0 Active-high	0	R/W	
		D2	PDAACK_Level	1 Active-low	0 Active-high	0	R/W	
		D1	PDRDWR_Level	1 Active-low	0 Active-high	0	R/W	
		D0	-	-		-	-	
DMA_Config_1 (DMA configuration 1)	0x300c95 (8 bits)	D7	RcvLimitMode	1 Receive limit mode	0 Normal	0	R/W	0 when being read.
		D6-4	-	-		-	-	
		D3	SingleWord	1 Single word	0 Multi word	0	R/W	
		D2-1	-	-		-	-	
		D0	CountMode	1 Count-down mode	0 Free-run mode	0	R/W	
DMA_Latency (DMA latency)	0x300c97 (8 bits)	D7-4	-	-		-	-	0 when being read.
		D3-0	DMA_Latency[3:0]	Latency		0x0	R/W	
DMA_Remain_H (DMA FIFO remain high)	0x300c98 (8 bits)	D7-4	-	-		-	-	0 when being read.
		D3-0	DMA_Remain[11:8]	DMA FIFO remain		0x0	R	
DMA_Remain_L (DMA FIFO remain low)	0x300c99 (8 bits)	D7-0	DMA_Remain[7:0]	DMA FIFO remain		0x0	R	
DMA_Count_HH (DMA transfer byte counter high/high)	0x300c9c (8 bits)	D7-0	DMA_Count[31:24]	DMA transfer byte counter		0x0	R/W	
DMA_Count_HL (DMA transfer byte counter high/low)	0x300c9d (8 bits)	D7-0	DMA_Count[23:16]	DMA transfer byte counter		0x0	R/W	
DMA_Count_LH (DMA transfer byte counter low/high)	0x300c9e (8 bits)	D7-0	DMA_Count[15:8]	DMA transfer byte counter		0x0	R/W	
DMA_Count_LL (DMA transfer byte counter low/low)	0x300c9f (8 bits)	D7-0	DMA_Count[7:0]	DMA transfer byte counter		0x0	R/W	

0x300e00

Prescaler (PSC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
PSC Control Register (PSC_CTL)	0x300e00 (8 bits)	D7-2	-	reserved		-	-	-	0 when being read.
		D1	PRUND	Prescaler run/stop in debug mode	1 Run	0 Stop	0	R/W	
		D0	PRUN	Prescaler run/stop control	1 Run	0 Stop	0	R/W	

0x301000-0x30100c

Watchdog Timer (WDT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Write Protect Register (WD_PROTECT)	0x301000 (16 bits)	D15-0	WDPTC [15:0]	WDT register write protect flag	Writing 0x96 removes the write protection of the WD_EN, WD_CMP_L, and WD_CMP_H registers (0x301002-0x301006). Writing another value set the write protection.	X	W	0 when being read.

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Enable and Setup Register (WD_EN)	0x301002 (16 bits)	D15-7	–	reserved	–	–	–	0 when being read.
		D6	CLKSEL	WDT input clock select	1 External clk 0 Internal clk	0	R/W	Write-protected
		D5	CLKEN	WDT clock output control	1 On 0 Off	0	R/W	
		D4	RUNSTP	WDT Run/Stop control	1 Run 0 Stop	0	R/W	
		D3-2	–	reserved	–	–	–	0 when being read.
		D1	NMIEN	WDT NMI enable	1 Enable 0 Disable	0	R/W	Write-protected
		D0	RESEN	WDT RESET enable	1 Enable 0 Disable	0	R/W	
WDT Comparison Data L Register (WD_CMP_L)	0x301004 (16 bits)	D15-0	CMPDT [15:0]	WDT comparison data CMPDT0 = LSB	0x0 to 0x3ffffff (low-order 16 bits)	0x0	R/W	Write-protected
WDT Comparison Data H Register (WD_CMP_H)	0x301006 (16 bits)	D15-14 D13-0	– CMPDT [29:16]	reserved WDT comparison data CMPDT29 = MSB	– 0x0 to 0x3ffffff (high-order 14 bits)	– 0x0	– R/W	0 when being read. Write-protected
WDT Count Data L Register (WD_CNT_L)	0x301008 (16 bits)	D15-0	CTRDT [15:0]	WDT counter data CTRDT0 = LSB	0x0 to 0x3ffffff (low-order 16 bits)	X	R	
WDT Count Data H Register (WD_CNT_H)	0x30100a (16 bits)	D15-14	–	reserved	–	–	–	0 when being read.
		D13-0	CTRDT [29:16]	WDT counter data CTRDT29 = MSB	0x0 to 0x3ffffff (high-order 14 bits)	X	R	
WDT Control Register (WD_CTL)	0x30100c (16 bits)	D15-1	–	reserved	–	–	–	0 when being read.
		D0	WDRESEN	WDT reset	1 Reset 0 ignored	0	W	

0x301100–0x301108

8-bit Timer (T8) Ch.0 (with Fine mode)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T8 Ch.0 Input Clock Select Register (T8_CLK0)	0x301100 (16 bits)	D15-4	–	reserved	–	–	–	0 when being read.	
		D3-0	DF[3:0]	T8 clock division ratio select (Prescaler output clock)	DF[3:0] Division ratio	0x0	R/W	Source clock = PCLK1	
						0xf reserved			
						0xe 1/16384			
						0xd 1/8192			
						0xc 1/4096			
						0xb 1/2048			
						0xa 1/1024			
						0x9 1/512			
						0x8 1/256			
						0x7 1/128			
						0x6 1/64			
						0x5 1/32			
						0x4 1/16			
						0x3 1/8			
						0x2 1/4			
				0x1 1/2					
				0x0 1/1					
T8 Ch.0 Reload Data Register (T8_TR0)	0x301102 (16 bits)	D15-8 D7-0	– TR[7:0]	reserved T8 reload data TR7 = MSB TR0 = LSB	– 0x0 to 0xff	– 0x0	– R/W	0 when being read.	
T8 Ch.0 Counter Data Register (T8_TC0)	0x301104 (16 bits)	D15-8 D7-0	– TC[7:0]	reserved T8 counter data TC7 = MSB TC0 = LSB	– 0x0 to 0xff	– 0xff	– R	0 when being read.	
T8 Ch.0 Control Register (T8_CTL0)	0x301106 (16 bits)	D15-12 D11-8	– TFMD[3:0]	reserved Fine mode setup	– 0x0 to 0xf	– 0x0	– R/W	0 when being read. Set a number of times to insert delay into a 16-underflow period.	
		D7-5	–	reserved	–	–	–	0 when being read.	
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W		
		D3-2	–	reserved	–	–	–	0 when being read.	
		D1	PRESER	Timer reset	1 Reset 0 ignored	0	W		
		D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W		
T8 Ch.0 Interrupt Control Register (T8_INT0)	0x301108 (16 bits)	D15-9	–	reserved	–	–	–	0 when being read.	
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W		
		D7-1	–	reserved	–	–	–	0 when being read.	
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.	

0x301110–0x301118

8-bit Timer (T8) Ch.1 (with Fine mode)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.1 Input Clock Select Register (T8_CLK1)	0x301110 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	DF[3:0]	T8 clock division ratio select (Prescaler output clock)	DF[3:0] Division ratio	0x0	R/W	Source clock = PCLK2
					0xf reserved			
					0xe 1/16384			
					0xd 1/8192			
					0xc 1/4096			
					0xb 1/2048			
					0xa 1/1024			
					0x9 1/512			
					0x8 1/256			
					0x7 1/128			
					0x6 1/64			
					0x5 1/32			
					0x4 1/16			
			0x3 1/8					
			0x2 1/4					
			0x1 1/2					
			0x0 1/1					
T8 Ch.1 Reload Data Register (T8_TR1)	0x301112 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	T8 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.1 Counter Data Register (T8_TC1)	0x301114 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	T8 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8 Ch.1 Control Register (T8_CTL1)	0x301116 (16 bits)	D15–12	–	reserved	–	–	–	0 when being read.
		D11–8	TFMD[3:0]	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	–	reserved	–	–	–	0 when being read.
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset 0 Ignored	0	W	
T8 Ch.1 Interrupt Control Register (T8_INT1)	0x301118 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x301120–0x301128

8-bit Timer (T8) Ch.2 (with Fine mode)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.2 Input Clock Select Register (T8_CLK2)	0x301120 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	DF[3:0]	T8 clock division ratio select (Prescaler output clock)	DF[3:0] Division ratio	0x0	R/W	Source clock = PCLK1
					0xf reserved			
					0xe 1/16384			
					0xd 1/8192			
					0xc 1/4096			
					0xb 1/2048			
					0xa 1/1024			
					0x9 1/512			
					0x8 1/256			
					0x7 1/128			
					0x6 1/64			
					0x5 1/32			
					0x4 1/16			
			0x3 1/8					
			0x2 1/4					
			0x1 1/2					
			0x0 1/1					
T8 Ch.2 Reload Data Register (T8_TR2)	0x301122 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	T8 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.2 Counter Data Register (T8_TC2)	0x301124 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	T8 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.2 Control Register (T8_CTL2)	0x301126 (16 bits)	D15–12	–	reserved	–	–	–	0 when being read.
		D11–8	TFMD[3:0]	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	–	reserved	–	–	–	0 when being read.
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset 0 Ignored	0	W	
	D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W		
T8 Ch.2 Interrupt Control Register (T8_INT2)	0x301128 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x301130–0x301138

8-bit Timer (T8) Ch.3 (with Fine mode)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T8 Ch.3 Input Clock Select Register (T8_CLK3)	0x301130 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.	
		D3–0	DF[3:0]	T8 clock division ratio select (Prescaler output clock)	DF[3:0] Division ratio	0x0	R/W	Source clock = PCLK2	
					0xf	reserved			
					0xe	1/16384			
					0xd	1/8192			
					0xc	1/4096			
					0xb	1/2048			
					0xa	1/1024			
					0x9	1/512			
					0x8	1/256			
					0x7	1/128			
					0x6	1/64			
					0x5	1/32			
					0x4	1/16			
			0x3	1/8					
			0x2	1/4					
			0x1	1/2					
			0x0	1/1					
T8 Ch.3 Reload Data Register (T8_TR3)	0x301132 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.	
		D7–0	TR[7:0]	T8 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W		
T8 Ch.3 Counter Data Register (T8_TC3)	0x301134 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.	
		D7–0	TC[7:0]	T8 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R		
T8 Ch.3 Control Register (T8_CTL3)	0x301136 (16 bits)	D15–12	–	reserved	–	–	–	0 when being read.	
		D11–8	TFMD[3:0]	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.	
		D7–5	–	reserved	–	–	–	0 when being read.	
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W		
		D3–2	–	reserved	–	–	–	0 when being read.	
		D1	PRESER	Timer reset	1 Reset 0 Ignored	0	W		
	D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W			
T8 Ch.3 Interrupt Control Register (T8_INT3)	0x301138 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.	
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W		
		D7–1	–	reserved	–	–	–	0 when being read.	
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.	

0x301140–0x301148

8-bit Timer (T8) Ch.4

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.4 Input Clock Select Register (T8_CLK4)	0x301140 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	DF[3:0]	T8 clock division ratio select (Prescaler output clock)	DF[3:0] Division ratio	0x0	R/W	Source clock = PCLK1
					0xf reserved			
					0xe 1/16384			
					0xd 1/8192			
					0xc 1/4096			
					0xb 1/2048			
					0xa 1/1024			
					0x9 1/512			
					0x8 1/256			
					0x7 1/128			
					0x6 1/64			
					0x5 1/32			
			0x4 1/16					
			0x3 1/8					
			0x2 1/4					
			0x1 1/2					
			0x0 1/1					
T8 Ch.4 Reload Data Register (T8_TR4)	0x301142 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	T8 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.4 Counter Data Register (T8_TC4)	0x301144 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	T8 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8 Ch.4 Control Register (T8_CTL4)	0x301146 (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset 0 Ignored	0	W	
	D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W		
T8 Ch.4 Interrupt Control Register (T8_INT4)	0x301148 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
	D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.	

0x301150–0x301158

8-bit Timer (T8) Ch.5

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.5 Input Clock Select Register (T8_CLK5)	0x301150 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	DF[3:0]	T8 clock division ratio select (Prescaler output clock)	DF[3:0] Division ratio	0x0	R/W	Source clock = PCLK2
					0xf reserved			
					0xe 1/16384			
					0xd 1/8192			
					0xc 1/4096			
					0xb 1/2048			
					0xa 1/1024			
					0x9 1/512			
					0x8 1/256			
					0x7 1/128			
					0x6 1/64			
					0x5 1/32			
			0x4 1/16					
			0x3 1/8					
			0x2 1/4					
			0x1 1/2					
			0x0 1/1					
T8 Ch.5 Reload Data Register (T8_TR5)	0x301152 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	T8 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.5 Counter Data Register (T8_TC5)	0x301154 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	T8 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8 Ch.5 Control Register (T8_CTL5)	0x301156 (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset 0 Ignored	0	W	
	D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W		

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.5 Interrupt Control Register (T8_INT5)	0x301158 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x301160–0x301168

8-bit Timer (T8) Ch.6

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T8 Ch.6 Input Clock Select Register (T8_CLK6)	0x301160 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.	
		D3–0	DF[3:0]	T8 clock division ratio select (Prescaler output clock)	DF[3:0]	Division ratio	0x0	R/W	Source clock = PCLK1
					0xf	reserved			
					0xe	1/16384			
					0xd	1/8192			
					0xc	1/4096			
					0xb	1/2048			
					0xa	1/1024			
					0x9	1/512			
					0x8	1/256			
					0x7	1/128			
					0x6	1/64			
					0x5	1/32			
					0x4	1/16			
0x3	1/8								
0x2	1/4								
0x1	1/2								
0x0	1/1								

T8 Ch.6 Reload Data Register (T8_TR6)	0x301162 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	T8 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.6 Counter Data Register (T8_TC6)	0x301164 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	T8 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8 Ch.6 Control Register (T8_CTL6)	0x301166 (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset 0 Ignored	0	W	
		D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W	
T8 Ch.6 Interrupt Control Register (T8_INT6)	0x301168 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x301170–0x301178

8-bit Timer (T8) Ch.7

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T8 Ch.7 Input Clock Select Register (T8_CLK7)	0x301170 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.	
		D3–0	DF[3:0]	T8 clock division ratio select (Prescaler output clock)	DF[3:0]	Division ratio	0x0	R/W	Source clock = PCLK2
					0xf	reserved			
					0xe	1/16384			
					0xd	1/8192			
					0xc	1/4096			
					0xb	1/2048			
					0xa	1/1024			
					0x9	1/512			
					0x8	1/256			
					0x7	1/128			
					0x6	1/64			
					0x5	1/32			
					0x4	1/16			
0x3	1/8								
0x2	1/4								
0x1	1/2								
0x0	1/1								

T8 Ch.7 Reload Data Register (T8_TR7)	0x301172 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	T8 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.7 Counter Data Register (T8_TC7)	0x301174 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	T8 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.7 Control Register (T8_CTL7)	0x301176 (16 bits)	D15-5	-	reserved	-	-	-	0 when being read.
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3-2	-	reserved	-	-	-	0 when being read.
		D1	PRESE	Timer reset	1 Reset 0 Ignored	0	W	
		D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W	
T8 Ch.7 Interrupt Control Register (T8_INT7)	0x301178 (16 bits)	D15-9	-	reserved	-	-	-	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7-1	-	reserved	-	-	-	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x301180-0x30118c

16-bit PWM Timer (T16A5) Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks							
T16A5 Ch.0 Counter Control Register (T16A_CTL0)	0x301180 (16 bits)	D15-14	-	reserved	-	-	-	0 when being read.							
		D13-12	DMASEL [1:0]	DMAC channel select	DMASEL[1:0]	DMAC channel	0x0	R/W							
					0x3	Ch.4/5									
				0x2	Ch.2/3										
				0x1	Ch.4/5										
				0x0	Ch.2/3										
		D11-8	CLKS[3:0]	Counter clock (division ratio) select	CLKS[3:0]	Division ratio	0x0	R/W	Source clock = PCLK1						
					0xf	External clock									
					0xe	1/16384									
					0xd	1/8192									
					0xc	1/4096									
					0xb	1/2048									
					0xa	1/1024									
					0x9	1/512									
0x8	1/256														
0x7	1/128														
		0x6	1/64												
		0x5	1/32												
		0x4	1/16												
		0x3	1/8												
		0x2	1/4												
		0x1	1/2												
		0x0	1/1												
D7	BUSY	Register writing status	1 Busy 0 Idle	0	R										
D6	-	reserved	-	-	-	0 when being read.									
D5-4	T16SEL [1:0]	Counter select	T16SEL[1:0]	Counter channel	0x0	R/W									
			0x3	Ch.1											
			0x2	Ch.0											
			0x1	Ch.1											
		0x0	Ch.0												
D3	CBUFEN	Compare buffer enable	1 Enable 0 Disable	0	R/W										
D2	TMMD	Count mode select	1 One-shot 0 Repeat	0	R/W										
D1	PRESET	Counter reset	1 Reset 0 Ignored	0	W	0 when being read.									
D0	PRUN	Counter run/stop control	1 Run 0 Stop	0	R/W										
T16A5 Ch.0 Counter Data Register (T16A_TC0)	0x301182 (16 bits)	D15-0	T16ATC [15:0]	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R								
T16A5 Ch.0 Comparator/ Capture Control Register (T16A_CCCTL0)	0x301184 (16 bits)	D15-14	CAPBTRG [1:0]	Capture B trigger select	CAPBTRG[1:0]	Trigger edge	0x0	R/W							
					0x3	↑ and ↓									
					0x2	↓									
					0x1	↑									
				0x0	None										
		D13-12	TOUTBMD [1:0]	TOUT B mode select	TOUTBMD[1:0]	Mode	0x0	R/W							
					0x3	cmp B: ↑ or ↓ cmp A: ↑ or ↓									
					0x2	cmp A: ↑ or ↓									
					0x1	cmp A: ↑, B: ↓									
				0x0	Off										
		D11-10	-	reserved	-	-	-	-	-	0 when being read.					
D9	TOUTBINV										TOUT B invert	1 Invert 0 Normal	0	R/W	
D8	CCBMD										T16A_CCB register mode select	1 Capture 0 Comparator	0	R/W	
D7-6	CAPATRG [1:0]	Capture A trigger select	CAPATRG[1:0]	Trigger edge	0x0	R/W									
			0x3	↑ and ↓											
			0x2	↓											
			0x1	↑											
		0x0	None												
D5-4	TOUTAMD [1:0]	TOUT A mode select	TOUTAMD[1:0]	Mode	0x0	R/W									
			0x3	cmp B: ↑ or ↓ cmp A: ↑ or ↓											
			0x2	cmp A: ↑ or ↓											
			0x1	cmp A: ↑, B: ↓											
		0x0	Off												
D3-2	-	reserved	-	-	-	-	0 when being read.								
D1	TOUTAINV	TOUT A invert	1 Invert 0 Normal	0	R/W										
D0	CCAMD	T16A_CCA register mode select	1 Capture 0 Comparator	0	R/W										

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T16A5 Ch.0 Comparator/Capture A Data Register (T16A_CCA0)	0x301186 (16 bits)	D15-0	CCA[15:0]	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W		
T16A5 Ch.0 Comparator/Capture B Data Register (T16A_CCB0)	0x301188 (16 bits)	D15-0	CCB[15:0]	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W		
T16A5 Ch.0 Comparator/Capture Interrupt Enable Register (T16A_IEN0)	0x30118a (16 bits)	D15-6	–	reserved	–	–	–	0 when being read.	
		D5	CAPBOWIE	Capture B overwrite interrupt enable	1 Enable 0 Disable	0	R/W	write interrupt enable	
		D4	CAPAOWIE	Capture A overwrite interrupt enable	1 Enable 0 Disable	0	R/W		
		D3	CAPBIE	Capture B interrupt enable	1 Enable 0 Disable	0	R/W		
		D2	CAPAIE	Capture A interrupt enable	1 Enable 0 Disable	0	R/W		
		D1	CBIE	Compare B interrupt enable	1 Enable 0 Disable	0	R/W		
		D0	CAIE	Compare A interrupt enable	1 Enable 0 Disable	0	R/W		
T16A5 Ch.0 Comparator/Capture Interrupt Flag Register (T16A_IFLG0)	0x30118c (16 bits)	D15-6	–	reserved	–	–	–		0 when being read.
		D5	CAPBOWIF	Capture B overwrite interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	CAPAOWIF	Capture A overwrite interrupt flag			0	R/W	
		D3	CAPBIF	Capture B interrupt flag			0	R/W	
		D2	CAPAIF	Capture A interrupt flag			0	R/W	
		D1	CBIF	Compare B interrupt flag			0	R/W	
		D0	CAIF	Compare A interrupt flag			0	R/W	

0x301190–0x30119c

16-bit PWM Timer (T16A5) Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A5 Ch.1 Counter Control Register (T16A_CTL1)	0x301190 (16 bits)	D15-14	–	reserved	–	–	–	0 when being read.
		D13-12	DMASEL [1:0]	DMAC channel select	DMASEL[1:0] DMAC channel	0x1	R/W	
					0x3 Ch.4/5			
					0x2 Ch.2/3			
					0x1 Ch.4/5			
					0x0 Ch.2/3			
		D11-8	CLKS[3:0]	Counter clock (division ratio) select	CLKS[3:0] Division ratio	0x0	R/W	Source clock = PCLK1
					0xf External clock			
					0xe 1/16384			
					0xd 1/8192			
					0xc 1/4096			
					0xb 1/2048			
					0xa 1/1024			
					0x9 1/512			
					0x8 1/256			
					0x7 1/128			
					0x6 1/64			
					0x5 1/32			
					0x4 1/16			
					0x3 1/8			
					0x2 1/4			
					0x1 1/2			
					0x0 1/1			
		D7	BUSY	Register writing status	1 Busy 0 Idle	0	R	
		D6	–	reserved	–	–	–	0 when being read.
		D5-4	T16SEL [1:0]	Counter select	T16SEL[1:0] Counter channel	0x1	R/W	
					0x3 Ch.1			
					0x2 Ch.0			
					0x1 Ch.1			
					0x0 Ch.0			
		D3	CBUFEN	Compare buffer enable	1 Enable 0 Disable	0	R/W	
		D2	TMMD	Count mode select	1 One-shot 0 Repeat	0	R/W	
		D1	PRESET	Counter reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	PRUN	Counter run/stop control	1 Run 0 Stop	0	R/W	
T16A5 Ch.1 Counter Data Register (T16A_TC1)	0x301192 (16 bits)	D15-0	T16ATC [15:0]	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
T16A5 Ch.1 Comparator/Capture Control Register (T16A_CCCTL1)	0x301194 (16 bits)	D15-14	CAPBTRG [1:0]	Capture B trigger select	CAPBTRG[1:0] Trigger edge 0x3 ↑ and ↓ 0x2 ↓ 0x1 ↑ 0x0 None	0x0	R/W		
		D13-12	TOUTBMD [1:0]	TOUT B mode select	TOUTBMD[1:0] Mode 0x3 cmp B: ↑ or ↓ 0x2 cmp A: ↑ or ↓ 0x1 cmp A: ↑, B: ↓ 0x0 Off	0x0	R/W		
		D11-10	–	reserved	–	–	–	–	0 when being read.
		D9	TOUTBINV	TOUT B invert	1 Invert	0 Normal	0	R/W	
		D8	CCBMD	T16A_CCB register mode select	1 Capture	0 Comparator	0	R/W	
		D7-6	CAPATRG [1:0]	Capture A trigger select	CAPATRG[1:0] Trigger edge 0x3 ↑ and ↓ 0x2 ↓ 0x1 ↑ 0x0 None	0x0	R/W		
		D5-4	TOUTAMD [1:0]	TOUT A mode select	TOUTAMD[1:0] Mode 0x3 cmp B: ↑ or ↓ 0x2 cmp A: ↑ or ↓ 0x1 cmp A: ↑, B: ↓ 0x0 Off	0x0	R/W		
		D3-2	–	reserved	–	–	–	–	0 when being read.
		D1	TOUTAINV	TOUT A invert	1 Invert	0 Normal	0	R/W	
		D0	CCAMD	T16A_CCA register mode select	1 Capture	0 Comparator	0	R/W	
T16A5 Ch.1 Comparator/Capture A Data Register (T16A_CCA1)	0x301196 (16 bits)	D15-0	CCA[15:0]	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W		
T16A5 Ch.1 Comparator/Capture B Data Register (T16A_CCB1)	0x301198 (16 bits)	D15-0	CCB[15:0]	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W		
T16A5 Ch.1 Comparator/Capture Interrupt Enable Register (T16A_IEN1)	0x30119a (16 bits)	D15-6	–	reserved	–	–	–	0 when being read.	
		D5	CAPBOWIE	Capture B overwrite interrupt enable	1 Enable	0 Disable	0	R/W	
		D4	CAPAOWIE	Capture A overwrite interrupt enable	1 Enable	0 Disable	0	R/W	
		D3	CAPBIE	Capture B interrupt enable	1 Enable	0 Disable	0	R/W	
		D2	CAPAIE	Capture A interrupt enable	1 Enable	0 Disable	0	R/W	
		D1	CBIE	Compare B interrupt enable	1 Enable	0 Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1 Enable	0 Disable	0	R/W	
T16A5 Ch.1 Comparator/Capture Interrupt Flag Register (T16A_IFLG1)	0x30119c (16 bits)	D15-6	–	reserved	–	–	–	0 when being read.	
		D5	CAPBOWIF	Capture B overwrite interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	CAPAOWIF	Capture A overwrite interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	
		D3	CAPBIF	Capture B interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	
		D2	CAPAIF	Capture A interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	
		D1	CBIF	Compare B interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	
		D0	CAIF	Compare A interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	

0x301200–0x30120e

16-bit Audio PWM Timer (T16P)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16P Compare A Buffer Register (T16P_A)	0x301200 (16 bits)	D15-0	CMPA[15:0]	Compare A data CMPA15 = MSB CMPA0 = LSB	0x0 to 0xffff	X	R/W	
T16P Compare B Buffer Register (T16P_B)	0x301202 (16 bits)	D15-0	CMPB[15:0]	Compare B data CMPB15 = MSB CMPB0 = LSB	0x0 to 0xffff	X	R/W	
T16P Counter Data Register (T16P_CNT_DATA)	0x301204 (16 bits)	D15-0	CNT_DATA [15:0]	Counter data CNT_DATA15 = MSB CNT_DATA0 = LSB	0x0 to 0xffff	X	R/W	
T16P Volume Control Register (T16P_VOL_CTL)	0x301206 (16 bits)	D15-8	–	reserved	–	–	–	0 when being read.
		D7	VOLBPS	Volume control enable	1 Disable	0 Enable	1	R/W
		D6-0	VOLSEL [6:0]	Volume level select	VOLSEL[6:0] Volume level 0x7f × 127/64 0x7e × 126/64 : : 0x40 × 64/64 : : 0x2 × 2/64 0x1 × 1/64 0x0 × 0 (mute)	0x40	R/W	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
T16P Control Register (T16P_CTL)	0x301208 (16 bits)	D15–12	BCNT[3:0]	B match count	0x0 to 0xf		0x0	R/W	Effective only for 16-bit data	
		D11	RESSEL	PCM data resolution select	1 16 bits	0 8 bits	1	R/W		
		D10	SGNSEL	PCM data format select	1 Signed	0 Unsigned	1	R/W		
		D9–8	SPLTMD [1:0]	Split mode select	SPLTMD[1:0]	Split mode	0x0	R/W		
		D7	–	reserved	–	–	–	–		0 when being read.
		D6	SELFPM	Fine mode select	1 Fine mode	0 Normal	0	R/W		
		D5	–	reserved	–	–	–	–		0 when being read.
		D4	INITOL	Initial output level select	1 High	0 Low	0	R/W		
		D3	CLKSEL	Input clock select	1 External	0 Internal	0	R/W		
		D2	–	reserved	–	–	–	–		0 when being read.
		D1	PRESET	T16P reset	1 Reset	0 Ignored	0	W		
D0	–	reserved	–	–	–	–				
T16P Running Control Register (T16P_RUN)	0x30120a (16 bits)	D15–1	–	reserved	–		–	–	0 when being read.	
		D0	PRUN	T16P run/stop control	1 Run	0 Stop	0	R/W		
T16P Internal Clock Control Register (T16P_CLK)	0x30120c (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.	
		D3–0	CLKDIV [3:0]	Counter clock division ratio select (Prescaler output clock)	CLKDIV[3:0]	Division ratio	0x0	R/W	Source clock = PCLK1	
					0xf–0xd	reserved				
					0xc	1/4096				
					0xb	1/2048				
					0xa	1/1024				
					0x9	1/512				
					0x8	1/256				
					0x7	1/128				
					0x6	1/64				
					0x5	1/32				
					0x4	1/16				
					0x3	1/8				
					0x2	1/4				
					0x1	1/2				
					0x0	1/1				
		T16P Interrupt Control Register (T16P_INT)	0x30120e (16 bits)	D15–11	–	reserved	–			–
D10	BUFEF			Buffer empty interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	X	R/W		Reset by writing 1.
D9	INTBF			B match interrupt flag			0	R/W		
D8	INTAF			A match interrupt flag			0	R/W		
D7–3	–			reserved	–		–	–	0 when being read.	
D2	INTBEEN			Buffer empty interrupt enable	1 Enable	0 Disable	0	R/W		
D1	INTBEN			B match interrupt enable	1 Enable	0 Disable	0	R/W		
D0	INTAEN			A match interrupt enable	1 Enable	0 Disable	0	R/W		

0x301300–0x301306

A/D Converter (ADC10)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
A/D Conversion Result Register (ADC10_ADD)	0x301300 (16 bits)	D15–0	ADD[15:0]	A/D converted data ADD[9:0] are effective when STMD = 0 (ADD[15:10] = 0) ADD[15:6] are effective when STMD = 1 (ADD[5:0] = 0)	0x0 to 0x3ff		0x0	R	
A/D Trigger/Channel Select Register (ADC10_TRG)	0x301302 (16 bits)	D15–14	–	reserved	–		–	–	0 when being read.
		D13–11	ADCE[2:0]	End channel select	0x0 to 0x5		0x0	R/W	
		D10–8	ADCS[2:0]	Start channel select	0x0 to 0x5		0x0	R/W	
		D7	STMD	Conversion result storing mode	1 ADD[15:6]	0 ADD[9:0]	0	R/W	
		D6	ADMS	Conversion mode select	1 Continuous	0 Single	0	R/W	
		D5–4	ADTS[1:0]	Conversion trigger select	ADTS[1:0]	Trigger	0x0	R/W	
					0x3	#ADTRIG pin reserved			
					0x2	T8 Ch.2			
					0x1	Software			
					0x0	Software			
		D3	–	reserved	–		–	–	
		D2–0	ADST[2:0]	Sampling time setting	ADST[2:0]	Sampling time	0x7	R/W	Always set to 0x7.
					0x7	9•ADCCLK			
					0x6	8•ADCCLK			
					0x5	7•ADCCLK			
					0x4	6•ADCCLK			
					0x3	5•ADCCLK			
					0x2	4•ADCCLK			
					0x1	3•ADCCLK			
					0x0	2•ADCCLK			

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Control/ Status Register (ADC10_CTL)	0x301304 (16 bits)	D15	–	reserved	–	–	–	0 when being read.
		D14–12	ADICH[2:0]	Conversion channel indicator	0x0 to 0x5	0x0	R	
		D11	–	reserved	–	–	–	0 when being read.
		D10	ADIBS	ADC10 status	1 Busy 0 Idle	0	R	
		D9	ADOWE	Overwrite error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D8	ADCF	Conversion completion flag	1 Completed 0 Run/Stand-by	0	R	Reset when ADC10_ADD is read.
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	ADDOIE	Overwrite error interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	ADCIE	Conversion completion int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	ADCTL	A/D conversion control	1 Start 0 Stop	0	R/W	
D0	ADEN	ADC10 enable	1 Enable 0 Disable	0	R/W			
A/D Clock Control Register (ADC10_CLK)	0x301306 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	ADDF[3:0]	A/D converter clock division ratio select	ADDF[3:0] Division ratio	0x0	R/W	Source clock = PCLK1
					0xf reserved			
					0xe 1/32768			
					0xd 1/16384			
					0xc 1/8192			
					0xb 1/4096			
					0xa 1/2048			
					0x9 1/1024			
					0x8 1/512			
					0x7 1/256			
					0x6 1/128			
					0x5 1/64			
					0x4 1/32			
					0x3 1/16			
			0x2 1/8					
			0x1 1/4					
			0x0 1/2					

0x301400–0x301412

I²S

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I ² S Control Register (I2S_CTL)	0x301400 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	DTSIGN	I ² S signed/unsigned data format select	1 Signed 0 Unsigned	0	R/W	
		D7	WCLKMD	I ² S output word clock mode select	1 L: High R: Low 0 L: Low R: High	0	R/W	
		D6	BCLKPOL	I ² S output bit clock polarity select	1 Negative 0 Positive	0	R/W	
		D5	DTFORM	I ² S output data format select	1 LSB first 0 MSB first	0	R/W	
		D4	I2SOUTEN	I ² S output enable	1 Enable 0 Disable	0	R/W	
		D3–2	DTTMG[1:0]	I ² S output data timing select	DTTMG[1:0] Timing mode	0x0	R/W	
					0x3 reserved 0x2 Right justified 0x1 Left justified 0x0 I ² S			
I ² S Master Clock Division Ratio Register (I2S_DV_MCLK)	0x301404 (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.
		D5–0	MCLKDIV[5:0]	I2S_MCLK division ratio select	MCLKDIV[5:0] Division ratio	0x0	R/W	Source clock = PCLK1
					0x3f 1/64			
					0x3e 1/63			
					0x3d 1/62			
			:					
			0x2 1/3					
			0x1 1/2					
			0x0 1/1					

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
I ² S Audio Clock Division Ratio Register (I2S_DV_AUDIO_CLK)	0x301406 (16 bits)	D15–13	–	reserved	–	–	–	0 when being read.			
		D12–8	WSCLKCYC[4:0]	I ² S WS clock cycle setup	WSCLKCYC[4:0] Clock period Other 0x10 32 clocks 0xf 31 clocks 0xe 30 clocks 0xd 29 clocks 0xc 28 clocks 0xb 27 clocks 0xa 26 clocks 0x9 25 clocks 0x8 24 clocks 0x7 23 clocks 0x6 22 clocks 0x5 21 clocks 0x4 20 clocks 0x3 19 clocks 0x2 18 clocks 0x1 17 clocks 0x0 16 clocks	0x0	R/W				
		D7–0	BCLKDIV[7:0]	I ² S bit clock division ratio select	BCLKDIV[7:0] Division ratio 0xff 1/512 0xfe 1/510 0xfd 1/508 : 0x2 1/6 0x1 1/4 0x0 1/2	0x0	R/W	Source clock = PCLK1			
I ² S Start/Stop Register (I2S_START)	0x301408 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.			
		D7	I2SBUSY	I ² S busy flag	1 Busy 0 Idle	0	R				
		D6–1	–	reserved	–	–	–	0 when being read.			
		D0	I2SSTART	I ² S start/stop control	1 Start (run) 0 Stop	0	R/W				
I ² S FIFO Status Register (I2S_FIFO_STAT)	0x30140a (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.			
		D4–2	FIFOSTAT[2:0]	I ² S FIFO state machine	FIFOSTAT[2:0] State 0x7–0x6 reserved 0x5 FLUSH 0x4 EMPTY 0x3 LACK 0x2 FULL 0x1 INIT 0x0 STOP	0x0	R				
					D1	I2SFIFOFF	I ² S FIFO full flag	1 Full 0 Not full	0	R	
					D0	I2SFIFOE	I ² S FIFO empty flag	1 Empty 0 Not empty	1	R	
I ² S Interrupt Control Register (I2S_INT)	0x30140c (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.			
		D10	WEIF	I ² S FIFO whole empty int. flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.		
		D9	HEIF	I ² S FIFO half empty interrupt flag	1	0	R/W				
		D8	OEIF	I ² S FIFO one empty interrupt flag	1	0	R/W				
		D7–3	–	reserved	–	–	–	0 when being read.			
		D2	WEIE	I ² S FIFO whole empty int. enable	1 Enable 0 Disable	0	R/W				
		D1	HEIE	I ² S FIFO half empty int. enable	1 Enable 0 Disable	0	R/W				
D0	OEIE	I ² S FIFO one empty int. enable	1 Enable 0 Disable	0	R/W						
I ² S FIFO Register (I2S_FIFO)	0x301410 (16 bits)	D15–0	I2SFIFO[31:0]	I ² S FIFO (L-channel output data)	0 to 0xffffffff	0x0	W	0 when being read.			
	0x301412 (16 bits)	D15–0		I ² S FIFO (R-channel output data)							

0x301500–0x301506

Remote Controller (REMC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
REMC Configuration Register (REMC_CFG)	0x301500 (16 bits)	D15–12	CGCLK[3:0]	Carrier generator clock division ratio select (Prescaler output clock)	CGCLK[3:0] LCCLK[3:0] Division ratio 0xf reserved 0xe 1/16384 0xd 1/8192 0xc 1/4096 0xb 1/2048 0xa 1/1024 0x9 1/512 0x8 1/256 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	Source clock = PCLK2	
					D11–8	LCCLK[3:0]	Length counter clock division ratio select (Prescaler output clock)		0x0
		D7–2	–	reserved	–	–	–	–	0 when being read.
		D1	REMMD	REMC mode select	1 Receive 0 Transmit	0	R/W		
		D0	REMEM	REMC enable	1 Enable 0 Disable	0	R/W		

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
REMC Carrier Length Setup Register (REMC_CAR)	0x301502 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.	
		D13–8	REMCL[5:0]	Carrier L length setup	0x0 to 0x3f	0x0	R/W		
		D7–6	–	reserved	–	–	–	0 when being read.	
		D5–0	REMCH[5:0]	Carrier H length setup	0x0 to 0x3f	0x0	R/W		
REMC Length Counter Register (REMC_LCNT)	0x301504 (16 bits)	D15–8	REMLEN[7:0]	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W		
		D7–1	–	reserved	–	–	–	0 when being read.	
		D0	REMDT	Transmit/receive data	1 1 (H) 0 0 (L)	0	R/W		
REMC Interrupt Control Register (REMC_INT)	0x301506 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.	
		D10	REMFIF	Falling edge interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	REMRFIF	Rising edge interrupt flag	–	–	0	R/W	
		D8	REMUIF	Underflow interrupt flag	–	–	0	R/W	
		D7–3	–	reserved	–	–	–	–	0 when being read.
		D2	REMFIE	Falling edge interrupt enable	1 Enable	0 Disable	0	R/W	
		D1	REMRIF	Rising edge interrupt enable	1 Enable	0 Disable	0	R/W	
		D0	REMUIE	Underflow interrupt enable	1 Enable	0 Disable	0	R/W	

0x302000–0x302094

LCD Controller (LCDC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCDC Interrupt Enable Register (LCDC_INT)	0x302000 (32 bits)	D31–1	–	reserved	–	–	–	0 when being read.
		D0	FRINTEN	Frame interrupt enable	1 Enable 0 Disable	0	R/W	
Status and Power Save Configuration Register (LCDC_PSAVE)	0x302004 (32 bits)	D31	FRINTF	Frame interrupt flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.
		D30–8	–	reserved	–	–	–	0 when being read.
		D7	VNDPF	Vertical display status flag	1 VNDP 0 VDP	1	R	
		D6–2	–	reserved	–	–	–	0 when being read.
		D1–0	PSAVE[1:0]	Power save mode select	PSAVE[1:0] Mode	0x0	R/W	
Horizontal Display Register (LCDC_HDISP)	0x302010 (32 bits)	D31–23	–	reserved	–	–	–	0 when being read.
		D22–16	HTCNT[6:0]	Horizontal total period (HT) setup HT = HDP + HNDP HT > HDPS + HDP (for HR-TFT)	HT = (HTCNT + 1) × 8 [Ts] HNDP = (HTCNT - HDPCNT) × 8 [Ts]	0x0	R/W	
		D15–7	–	reserved	–	–	–	0 when being read.
		D6–0	HDPCNT [6:0]	Horizontal display period (HDP) setup	HDP = (HDPCNT + 1) × 8 [Ts]	0x0	R/W	
Vertical Display Register (LCDC_VDISP)	0x302014 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	VTCNT[9:0]	Vertical total period (VT) setup VT = VDP + VNDP VT > VDPS + VDP (for HR-TFT)	VT = VTCNT + 1 [lines] VNDP = VTCNT - VDPCNT [lines]	0x0	R/W	
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	VDPCNT [9:0]	Vertical display period (VDP) setup	VDP = VDPCNT + 1 [lines]	0x0	R/W	
MOD Rate Register (LCDC_MODR)	0x302018 (32 bits)	D31–6	–	reserved	–	–	–	0 when being read.
		D5–0	MOD[5:0]	LCD MOD rate setup	0x0 to 0x3f	0x0	R/W	
Horizontal Display Start Position Register (LCDC_HDPS)	0x302020 (32 bits)	D31–10	–	reserved	–	–	–	0 when being read.
		D9–0	HDPSCNT [9:0]	Horizontal display period start position for TFT HT > HDP + HDPS + 1 (HR-TFT) HT > HDP + HDPS (other TFT)	HDPS = HDPSCNT [Ts]	0x0	R/W	0x0 must be set for STN panels.
Vertical Display Start Position Register (LCDC_VDPS)	0x302024 (32 bits)	D31–10	–	reserved	–	–	–	0 when being read.
		D9–0	VDPCNT [9:0]	Vertical display period start position for TFT VT > VDP + VDPS	VDPS = VDPCNT [lines]	0x0	R/W	0x0 must be set for STN panels.
FPLINE Pulse Setting Register (LCDC_FPLINE)	0x302028 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	FPLINE_ST[9:0]	FPLINE pulse start position setup	Start position = FPLINE_ST + 1 [Ts]	0x0	R/W	*1: For TFT 0x0 must be set for STN panels.
		D15–8	–	reserved	–	–	–	0 when being read.
		D7	FPLINE_POL	FPLINE pulse polarity setup	1 Active high 0 Active low	0	R/W	(*1)
FPLINE Pulse Setting Register (LCDC_FPLINE)	0x302028 (32 bits)	D6–0	FPLINE_WD[6:0]	FPLINE pulse width setup	Pulse width = FPLINE_WD + 1 [Ts]	0x0	R/W	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FPFRAME Pulse Setting Register (LCDC_FPFR)	0x30202c (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	FPFRAME_ST[9:0]	FPFRAME pulse start position setup	Start position = FPFRAME_ST × HT [Ts]	0x0	R/W	*1: For TFT 0x0 must be set for STN panels.
		D15–8	–	reserved	–	–	–	0 when being read.
		D7	FPFRAME_POL	FPFRAME pulse polarity setup	1 Active high 0 Active low	0	R/W	(*1)
		D6–0	FPFRAME_WD[6:0]	FPFRAME pulse width setup	Pulse width = (FPFRAME_WD+1) × HT [Ts]	0x0	R/W	(*1)
FPFRAME Pulse Offset Register (LCDC_FPFROFS)	0x302030 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	FPFRAME_STOFS[9:0]	FPFRAME pulse stop offset	Stop offset = FPFRAME_STOFS [Ts]	0x0	R/W	*1: For TFT 0x0 must be set for STN panels.
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	FPFRAME_STOFS[9:0]	FPFRAME pulse start offset	Start offset = FPFRAME_STOFS [Ts]	0x0	R/W	(*1)
TFT Special Output Register (LCDC_TFTSO)	0x302040 (32 bits)	D31–4	–	reserved	–	–	–	0 when being read.
		D3	CTL1CTL	TFT_CTL1 control	1 Program 0 Toggle/line	0	R/W	For TFT
		D2	CTLCNT_RUN	TFT_CTL0–2 control counter run/stop	1 Run 0 Stop	0	R/W	0x0 must be set for STN panels.
		D1	FPSHIFT_POL	FPSHIFT polarity	1 Falling 0 Rising	0	R/W	
		D0	CTL01SWAP	TFT_CTL0/TFT_CTL1 swap	1 Swap 0 Not swap	0	R/W	
TFT_CTL1 Pulse Register (LCDC_TFT_CTL1)	0x302044 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	CTL1STP[9:0]	TFT_CTL1 pulse stop offset TFT_CTL1 pulse width = (CTL1STP - CTL1ST + 1) Ts	Stop offset = CTL1STP + 1 [Ts]	0x0	R/W	*2: For TFT This register is enabled when CTLCNT_RUN = 1.
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	CTL1ST[9:0]	TFT_CTL1 pulse start offset	Start offset = CTL1ST [Ts]	0x0	R/W	(*2)
TFT_CTL0 Pulse Register (LCDC_TFT_CTL0)	0x302048 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.
		D25–16	CTL0STP[9:0]	TFT_CTL0 pulse stop offset TFT_CTL0 pulse width = (CTL0STP - CTL0ST + 1) Ts	Stop offset = CTL0STP + 1 [Ts]	0x0	R/W	*2: For TFT This register is enabled when CTLCNT_RUN = 1.
		D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	CTL0ST[9:0]	TFT_CTL0 pulse start offset	Start offset = CTL0ST [Ts]	0x0	R/W	(*2)
TFT_CTL2 Register (LCDC_TFT_CTL2)	0x30204c (32 bits)	D31–10	–	reserved	–	–	–	0 when being read.
		D9–0	CTL2DLY[9:0]	TFT_CTL2 delay setup	Delay = CTL2DLY [Ts]	0x0	R/W	*2: For TFT This register is enabled when CTLCNT_RUN = 1.
LCDC Reload Control Register (LCDC_RLDCTL)	0x302050 (32 bits)	D31–2	–	reserved	–	–	–	0 when being read.
		D1	LUTRLD	LUT reload trigger	1 Trigger 0 Ignored	0	W	
					1 Reloading 0 Finished		R	
		D0	CTABRLD	Control table reload trigger	1 Trigger 0 Ignored	0	W	
1 Reloading 0 Finished					R			
LCDC Reload Table Base Address Register (LCDC_RLDADR)	0x302054 (32 bits)	D31–10	RTBL_BADR[31:10]	Reload table base address (1KB boundary address, A[9:0] = 0x0)	Areas 3*–5, 7–10, 13–16, and 19–22	0x0	R/W	* DSTRAM cannot be used.
		D9–0	–	reserved	–	–	–	0 when being read.

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
LCDC Display Mode Register (LCDC_DISPMD)	0x302060 (32 bits)	D31	PANSELSEL	Panel type select	1 TFT 0 STN	0	R/W		
		D30	COLOR	Color/mono select	1 Color 0 Mono	0	R/W		
		D29	FPSHIFT_MSK	FPSHIFT mask enable	1 Enable 0 Disable	0	R/W		
		D28	–	reserved	–	–	–	–	0 when being read.
		D27–26	DWD[1:0]	LCD panel data width select	DWD[1:0] Data width	0x0	R/W		
						0x3 8 bits (fmt2)			
						0x2 reserved			
						0x1 8 bits (fmt1)			
						0x0 4 bits			
		D25	SWINV	Software video invert	1 Invert 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 Repeat 0 Not repeat	0	R/W		
D6–5	–	reserved	–	–	–	–	0 when being read.		
D4	LUTPASS	LUT bypass mode select	1 Bypass 0 Use	1	R/W				
D3	–	reserved	–	–	–	–	0 when being read.		
D2–0	BPP[2:0]	Bit-per-pixel select	BPP[2:0] bpp	0x0	R/W				
				0x7 reserved					
				0x6 24 bpp					
				0x5 16 bpp					
				0x4 12 bpp					
				0x3 8 bpp					
				0x2 4 bpp					
				0x1 2 bpp					
				0x0 1 bpp					
Main Window Display Start Address Register (LCDC_MAINADR)	0x302070 (32 bits)	D31–0	MW_START [31:0]	Main window start address MW_START31 = MSB MW_START0 = LSB	0x0 to 0xfffffff (Areas 3–5, 7–10, 13–16, and 19–22)	0x0	R/W		
Main Screen Address Offset Register (LCDC_MAINOFS)	0x302074 (32 bits)	D31–12	–	reserved	–	–	–	0 when being read.	
		D11–0	MW_OFS [11:0]	Main screen address offset	Main screen width (pixels) × bpp/32	0x0	R/W		
Sub-window Display Start Address Register (LCDC_SUBADR)	0x302080 (32 bits)	D31–0	SW_START [31:0]	Sub-window start address SW_START31 = MSB SW_START0 = LSB	0x0 to 0xfffffff (Areas 3–5, 7–10, 13–16, and 19–22)	0x0	R/W		
Sub-screen Address Offset Register (LCDC_SUBOFS)	0x302084 (32 bits)	D31–12	–	reserved	–	–	–	0 when being read.	
		D11–0	SW_OFS [11:0]	Sub-screen address offset	Sub-screen width (pixels) × bpp/32	0x0	R/W		
Sub-window Start Position Register (LCDC_SUBSP)	0x302088 (32 bits)	D31	PIP_EN	PIP enable	1 Enable 0 Disable	0	R/W		
		D30–26	–	reserved	–	–	–	0 when being read.	
		D25–16	PIP_YSTART [9:0]	Sub-window vertical (Y) start position	Y start position = PIP_YSTART (lines) from the origin	0x0	R/W	*3: This register is enabled when PIP_EN = 1.	
		D15–10	–	reserved	–	–	–	0 when being read.	
		D9–0	PIP_XSTART [9:0]	Sub-window horizontal (X) start position	X start position = PIP_XSTART (pixels) from the origin (word units)	0x0	R/W	(*3)	
Sub-window End Position Register (LCDC_SUBEP)	0x30208c (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.	
		D25–16	PIP_YEND [9:0]	Sub-window vertical (Y) end position	Y end position = PIP_YEND (lines) from the origin	0x0	R/W	*3: This register is enabled when PIP_EN = 1.	
		D15–10	–	reserved	–	–	–	0 when being read.	
		D9–0	PIP_XEND [9:0]	Sub-window horizontal (X) end position	X end position = PIP_XEND (pixels) from the origin (word units)	0x0	R/W	(*3)	
Monochrome Look-up Table Register 0 (LCDC_MLUT0)	0x302090 (32 bits)	D31–28	MLUT7[3:0]	Monochrome LUT entry 7 data	0x0 to 0xf	0x0	R/W		
		D27–24	MLUT6[3:0]	Monochrome LUT entry 6 data	0x0 to 0xf	0x0	R/W		
		D23–20	MLUT5[3:0]	Monochrome LUT entry 5 data	0x0 to 0xf	0x0	R/W		
		D19–16	MLUT4[3:0]	Monochrome LUT entry 4 data	0x0 to 0xf	0x0	R/W		
		D15–12	MLUT3[3:0]	Monochrome LUT entry 3 data	0x0 to 0xf	0x0	R/W		
		D11–8	MLUT2[3:0]	Monochrome LUT entry 2 data	0x0 to 0xf	0x0	R/W		
		D7–4	MLUT1[3:0]	Monochrome LUT entry 1 data	0x0 to 0xf	0x0	R/W		
		D3–0	MLUT0[3:0]	Monochrome LUT entry 0 data	0x0 to 0xf	0x0	R/W		

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Monochrome Look-up Table Register 1 (LCD_C_MLUT1)	0x302094 (32 bits)	D31–28	MLUT15[3:0]	Monochrome LUT entry 15 data	0x0 to 0xf	0x0	R/W	
		D27–24	MLUT14[3:0]	Monochrome LUT entry 14 data	0x0 to 0xf	0x0	R/W	
		D23–20	MLUT13[3:0]	Monochrome LUT entry 13 data	0x0 to 0xf	0x0	R/W	
		D19–16	MLUT12[3:0]	Monochrome LUT entry 12 data	0x0 to 0xf	0x0	R/W	
		D15–12	MLUT11[3:0]	Monochrome LUT entry 11 data	0x0 to 0xf	0x0	R/W	
		D11–8	MLUT10[3:0]	Monochrome LUT entry 10 data	0x0 to 0xf	0x0	R/W	
		D7–4	MLUT9[3:0]	Monochrome LUT entry 9 data	0x0 to 0xf	0x0	R/W	
		D3–0	MLUT8[3:0]	Monochrome LUT entry 8 data	0x0 to 0xf	0x0	R/W	

0x302100–0x30211c

DMA Controller (DMAC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
DMAC Channel Enable Register (DMAC_CH_EN)	0x302100 (32 bits)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	DMAON7	DMAC Ch.7 enable	1 Enable 0 Disable	0	R/W		
		D6	DMAON6	DMAC Ch.6 enable	1 Enable 0 Disable	0	R/W		
		D5	DMAON5	DMAC Ch.5 enable	1 Enable 0 Disable	0	R/W		
		D4	DMAON4	DMAC Ch.4 enable	1 Enable 0 Disable	0	R/W		
		D3	DMAON3	DMAC Ch.3 enable	1 Enable 0 Disable	0	R/W		
		D2	DMAON2	DMAC Ch.2 enable	1 Enable 0 Disable	0	R/W		
		D1	DMAON1	DMAC Ch.1 enable	1 Enable 0 Disable	0	R/W		
		D0	DMAON0	DMAC Ch.0 enable	1 Enable 0 Disable	0	R/W		
DMAC Control Table Base Address Register (DMAC_TBL_BASE)	0x302104 (32 bits)	D31–10	TBL_BASE [31:10]	DMAC control table base address	0x0 to 0xfffffc00 (1,024-byte boundary address within a RAM)	0x80000	R/W		
		D9–0	TBL_BASE [9:0]	Fixed at 0x0 (Cannot be altered.)			R		
DMAC Interrupt Enable Register (DMAC_IE)	0x302108 (32 bits)	D31–8	–	reserved	–	–	–	0 when being read.	
		D7	DMAIE7	DMAC Ch.7 interrupt enable	1 Enable 0 Disable	0	R/W		
		D6	DMAIE6	DMAC Ch.6 interrupt enable	1 Enable 0 Disable	0	R/W		
		D5	DMAIE5	DMAC Ch.5 interrupt enable	1 Enable 0 Disable	0	R/W		
		D4	DMAIE4	DMAC Ch.4 interrupt enable	1 Enable 0 Disable	0	R/W		
		D3	DMAIE3	DMAC Ch.3 interrupt enable	1 Enable 0 Disable	0	R/W		
		D2	DMAIE2	DMAC Ch.2 interrupt enable	1 Enable 0 Disable	0	R/W		
		D1	DMAIE1	DMAC Ch.1 interrupt enable	1 Enable 0 Disable	0	R/W		
		D0	DMAIE0	DMAC Ch.0 interrupt enable	1 Enable 0 Disable	0	R/W		
DMAC Trigger Select Register (DMAC_TRG_SEL)	0x30210c (32 bits)	D31–16	–	reserved	–	–	–		0 when being read.
		D15–14	TRG_SEL7 [1:0]	Ch.7 trigger select	TRG_SEL7[1:0]	Trigger source	0x0	R/W	
					0x3 0x2 0x1 0x0	ADC complete reserved USIL Tx No hard trigger			
		D13–12	TRG_SEL6 [1:0]	Ch.6 trigger select	TRG_SEL6[1:0]	Trigger source	0x0	R/W	
					0x3 0x2 0x1 0x0	USB reserved USIL Rx No hard trigger			
		D11–10	TRG_SEL5 [1:0]	Ch.5 trigger select	TRG_SEL5[1:0]	Trigger source	0x0	R/W	
					0x3 0x2 0x1 0x0	T16A5 Ch.x A FSIO Ch.1 Tx reserved No hard trigger			
		D9–8	TRG_SEL4 [1:0]	Ch.4 trigger select	TRG_SEL4[1:0]	Trigger source	0x0	R/W	
					0x3 0x2 0x1 0x0	T16A5 Ch.x B FSIO Ch.1 Rx reserved No hard trigger			
		D7–6	TRG_SEL3 [1:0]	Ch.3 trigger select	TRG_SEL3[1:0]	Trigger source	0x0	R/W	
0x3 0x2 0x1 0x0	T16A5 Ch.x A FSIO Ch.0 Tx USI Tx No hard trigger								
D5–4	TRG_SEL2 [1:0]	Ch.2 trigger select	TRG_SEL2[1:0]	Trigger source	0x0	R/W			
			0x3 0x2 0x1 0x0	T16A5 Ch.x B FSIO Ch.0 Rx USI Rx No hard trigger					
D3–2	TRG_SEL1 [1:0]	Ch.1 trigger select	TRG_SEL1[1:0]	Trigger source	0x0	R/W			
			0x3 0x2 0x1 0x0	USB Port I ² S R No hard trigger					
D1–0	TRG_SEL0 [1:0]	Ch.0 trigger select	TRG_SEL0[1:0]	Trigger source	0x0	R/W			
			0x3 0x2 0x1 0x0	ADC complete T16P I ² S L No hard trigger					

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks		
DMAC Trigger Flag Register (DMAC_TRG_FLG)	0x302110 (32 bits)	D31-8	-	reserved	-	-	-	0 when being read.		
		D7	TRG7	Ch.7 software trigger/trigger status	1 Soft trigger	0 (W)	0 (W)	0	R/W	
		D6	TRG6	Ch.6 software trigger/trigger status				0	R/W	
		D5	TRG5	Ch.5 software trigger/trigger status				0	R/W	
		D4	TRG4	Ch.4 software trigger/trigger status	1 Triggered	0 (R)	0 Not triggered	0	R/W	
		D3	TRG3	Ch.3 software trigger/trigger status				0	R/W	
		D2	TRG2	Ch.2 software trigger/trigger status				0	R/W	
		D1	TRG1	Ch.1 software trigger/trigger status				0	R/W	
		D0	TRG0	Ch.0 software trigger/trigger status				0	R/W	
DMAC End-of-Transfer Flag Register (DMAC_END_FLG)	0x302114 (32 bits)	D31-8	-	reserved	-	-	-	0 when being read.		
		D7	ENDF7	Ch.7 end-of-transfer flag	1	Finished	0	Not finished	0	R/W
		D6	ENDF6	Ch.6 end-of-transfer flag	1	Finished	0	Not finished	0	R/W
		D5	ENDF5	Ch.5 end-of-transfer flag	1	Finished	0	Not finished	0	R/W
		D4	ENDF4	Ch.4 end-of-transfer flag	1	Finished	0	Not finished	0	R/W
		D3	ENDF3	Ch.3 end-of-transfer flag	1	Finished	0	Not finished	0	R/W
		D2	ENDF2	Ch.2 end-of-transfer flag	1	Finished	0	Not finished	0	R/W
		D1	ENDF1	Ch.1 end-of-transfer flag	1	Finished	0	Not finished	0	R/W
		D0	ENDF0	Ch.0 end-of-transfer flag	1	Finished	0	Not finished	0	R/W
DMAC Running Status Register (DMAC_RUN_STAT)	0x302118 (32 bits)	D31-8	-	reserved	-	-	-	0 when being read.		
		D7	RUN7	Ch.7 running status	1	Running	0	Idle/paused	0	R
		D6	RUN6	Ch.6 running status	1	Running	0	Idle/paused	0	R
		D5	RUN5	Ch.5 running status	1	Running	0	Idle/paused	0	R
		D4	RUN4	Ch.4 running status	1	Running	0	Idle/paused	0	R
		D3	RUN3	Ch.3 running status	1	Running	0	Idle/paused	0	R
		D2	RUN2	Ch.2 running status	1	Running	0	Idle/paused	0	R
		D1	RUN1	Ch.1 running status	1	Running	0	Idle/paused	0	R
		D0	RUN0	Ch.0 running status	1	Running	0	Idle/paused	0	R
DMAC Pause Status Register (DMAC_PAUSE_STAT)	0x30211c (32 bits)	D31-8	-	reserved	-	-	-	0 when being read.		
		D7	PAUSE7	Ch.7 pause status	1	Paused	0	Not paused	0	R
		D6	PAUSE6	Ch.6 pause status	1	Paused	0	Not paused	0	R
		D5	PAUSE5	Ch.5 pause status	1	Paused	0	Not paused	0	R
		D4	PAUSE4	Ch.4 pause status	1	Paused	0	Not paused	0	R
		D3	PAUSE3	Ch.3 pause status	1	Paused	0	Not paused	0	R
		D2	PAUSE2	Ch.2 pause status	1	Paused	0	Not paused	0	R
		D1	PAUSE1	Ch.1 pause status	1	Paused	0	Not paused	0	R
		D0	PAUSE0	Ch.0 pause status	1	Paused	0	Not paused	0	R

0x302200-0x302210

SDRAM Controller (SDRAMC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks								
SDRAM Initialization Register (SDRAMC_INIT)	0x302200 (32 bits)	D31-5	-	reserved	-	-	-	0 when being read.								
		D4	SDON	SDRAM controller enable	1	Enable	0	Disable	0	R/W						
		D3	INIDO	SDRAM initialization status	1	Finished	0	Busy	0	R						
		D2	INIMRS	MRS command enable for init.	1	Enable	0	Disable	0	R/W						
		D1	INIPRE	PALL command enable for init.	1	Enable	0	Disable	0	R/W						
		D0	INIREF	REF command enable for init.	1	Enable	0	Disable	0	R/W						
SDRAM Configuration Register (SDRAMC_CFG)	0x302204 (32 bits)	D31-14	-	reserved	-	-	-	0 when being read.								
		D13-12	T24NS[1:0]	Number of SDRAM tRP and tRCD cycles	T24NS[1:0] # of cycles		0x0	R/W								
					0x3	4 cycles										
					0x2	3 cycles										
					0x1	2 cycles										
					0x0	1 cycle										
		D11	-	reserved	-	-	-	-	0 when being read.							
										D10-8	T60NS[2:0]	Number of SDRAM tRAS cycles	T60NS[2:0] # of cycles		0x0	R/W
													0x7	8 cycles		
													0x6	7 cycles		
:	:															
0x1	2 cycles															
0x0	1 cycle															
D7-4	T80NS[3:0]	Number of SDRAM tRC, tRFC and tBSR cycles	T80NS[3:0] # of cycles		0xe	R/W										
			0xf	16 cycles												
			0xe	15 cycles												
			:	:												
0x1	2 cycles															
0x0	1 cycle															
D3	-	reserved	-	-	-	-	0 when being read.									
D2-0	ADDRC[2:0]	SDRAM address configuration	ADDRC[2:0] Configuration		0x0	R/W										
			0x7	512M bits												
			0x6	128M bits x 2												
			0x5	64M bits x 2												
			0x4	reserved												
			0x3	256M bits												
			0x2	128M bits												
			0x1	64M bits												
0x0	16M bits															

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
SDRAM Refresh Control Register (SDRAMC_REF)	0x302208 (32 bits)	D31–26	–	reserved	–	–	–	0 when being read.	
		D25	SREFDO	SDRAM self-refresh status	1 Finished 0 Busy	0	R		
		D24	SCKON	SDRAM clock during self-refresh	1 Enable 0 Disable	0	R/W		
		D23	SELEN	SDRAM self-refresh enable	1 Enable 0 Disable	0	R/W		
		D22–16	SELCO[6:0]	SDRAM self-refresh counter	0x0 to 0x7f	0x7f	R/W		
		D15–12	–	reserved	–	–	–	–	0 when being read.
	D11–0	AURCO[11:0]	SDRAM auto-refresh counter	0x0 to 0xffff	0x8c	R/W			
SDRAM Application Configuration Register (SDRAMC_APP)	0x302210 (32 bits)	D31–6	–	reserved	–	–	–	0 when being read.	
		D5	DBF	Double frequency mode enable	1 Enable 0 Disable	0	W		
		D4	–	reserved	–	–	–	–	
		D3–2	CAS[1:0]	CAS latency setup	CAS[1:0]	CAS latency	0x2	R/W	
					0x3	3			
					0x2	2			
					0x1	1			
	0x0	reserved							
D1	–	reserved	–	–	–	0 when being read.			
D0	–	reserved	–	–	0	R/W	Do not set to 1.		

0x302220–0x302228

SRAM Controller (SRAMC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
#CE[7:4] Access Timing Configuration Register (SRAMC_TM47)	0x302220 (32 bits)	D31–28	CE7SETUP [1:0]	#CE7 setup cycle	CE7SETUP[1:0]	Setup cycle	0x3	R/W	
					0x3	4 cycles			
					0x2	3 cycles			
					0x1	2 cycles			
					0x0	1 cycle			
		D29–28	CE7HOLD [1:0]	#CE7 hold cycle	CE7HOLD[1:0]	Hold cycle	0x3	R/W	
					0x3	4 cycles			
					0x2	3 cycles			
					0x1	2 cycles			
					0x0	1 cycle			
		D27–24	CE7WAIT [3:0]	#CE7 static wait cycle	CE7WAIT[3:0]	Wait cycle	0xf	R/W	
					0xf	15 cycles			
					0xe	14 cycles			
					:	:			
					0x1	1 cycle			
			0x0	0 cycles					
D23–16	–	reserved	–	–	–	–	1 when being read.		
D15–14	CE5SETUP [1:0]	#CE5 setup cycle	CE5SETUP[1:0]	Setup cycle	0x3	R/W			
			0x3	4 cycles					
			:	:					
	0x0	1 cycle							
D13–12	CE5HOLD [1:0]	#CE5 hold cycle	CE5HOLD[1:0]	Hold cycle	0x3	R/W			
			0x3	4 cycles					
			:	:					
	0x0	1 cycle							
D11–8	CE5WAIT [3:0]	#CE5 static wait cycle	CE5WAIT[3:0]	Wait cycle	0xf	R/W			
			0xf	15 cycles					
			:	:					
	0x0	0 cycles							
D7–6	CE4SETUP [1:0]	#CE4 setup cycle	CE4SETUP[1:0]	Setup cycle	0x3	R/W			
			0x3	4 cycles					
			:	:					
	0x0	1 cycle							
D5–4	CE4HOLD [1:0]	#CE4 hold cycle	CE4HOLD[1:0]	Hold cycle	0x3	R/W			
			0x3	4 cycles					
			:	:					
	0x0	1 cycle							
D3–0	CE4WAIT [3:0]	#CE4 static wait cycle	CE4WAIT[3:0]	Wait cycle	0xf	R/W			
			0xf	15 cycles					
			:	:					
	0x0	0 cycles							

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks			
#CE[10:8] Access Timing Configuration Register (SRAMC_ TMG810)	0x302224 (32 bits)	D31-24	-	reserved	-	-	-	-	1 when being read.		
		D23-22	CE10SETUP [1:0]	#CE10 setup cycle	CE10SETUP[1:0]	Setup cycle 0x3 4 cycles 0x2 3 cycles 0x1 2 cycles 0x0 1 cycle	0x3	R/W			
		D21-20	CE10HOLD [1:0]	#CE10 hold cycle	CE10HOLD[1:0]	Hold cycle 0x3 4 cycles 0x2 3 cycles 0x1 2 cycles 0x0 1 cycle	0x3	R/W			
		D19-16	CE10WAIT [3:0]	#CE10 static wait cycle	CE10WAIT[3:0]	Wait cycle 0xf 15 cycles 0xe 14 cycles : : 0x1 1 cycle 0x0 0 cycles	0xf	R/W			
		D15-14	CE9SETUP [1:0]	#CE9 setup cycle	CE9SETUP[1:0]	Setup cycle 0x3 4 cycles : : 0x0 1 cycle	0x3	R/W			
		D13-12	CE9HOLD [1:0]	#CE9 hold cycle	CE9HOLD[1:0]	Hold cycle 0x3 4 cycles : : 0x0 1 cycle	0x3	R/W			
		D11-8	CE9WAIT [3:0]	#CE9 static wait cycle	CE9WAIT[3:0]	Wait cycle 0xf 15 cycles : : 0x0 0 cycles	0xf	R/W			
		D7-6	CE8SETUP [1:0]	#CE8 setup cycle	CE8SETUP[1:0]	Setup cycle 0x3 4 cycles : : 0x0 1 cycle	0x3	R/W			
		D5-4	CE8HOLD [1:0]	#CE8 hold cycle	CE8HOLD[1:0]	Hold cycle 0x3 4 cycles : : 0x0 1 cycle	0x3	R/W			
		D3-0	CE8WAIT [3:0]	#CE8 static wait cycle	CE8WAIT[3:0]	Wait cycle 0xf 15 cycles : : 0x0 0 cycles	0xf	R/W			
		#CE[10:4] Device Configuration Register (SRAMC_TYPE)	0x302228 (32 bits)	D31-14	-	reserved	-	-	-	-	0 when being read.
				D13-12	CE10TYPE [1:0]	#CE10 device type	CExTYPE[1:0]	Device type 0x3-0x2 8-bit device 0x1 16-bit BSL type 0x0 16-bit A0 type	0x0	R/W	
D11-10	CE9TYPE [1:0]			#CE9 device type			0x0	R/W			
D9-8	CE8TYPE [1:0]			#CE8 device type			0x0	R/W			
D7-6	CE7TYPE [1:0]			#CE7 device type			0x0	R/W			
D5-4	-			reserved	-	-	-	-	-	0 when being read.	
D3-2	CE5TYPE [1:0]			#CE5 device type	CExTYPE[1:0]	Device type 0x3-0x2 8-bit device 0x1 16-bit BSL type 0x0 16-bit A0 type	0x0	R/W			
D1-0	CE4TYPE [1:0]			#CE4 device type			0x0	R/W			

0x302300-0x302360

Cache Controller (CCU)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Cache Configuration Register (CCU_CFG)	0x302300 (32 bits)	D31-9	-	reserved	-	-	-	-	0 when being read.
		D8	WBEN	Write buffer enable	1 Enable 0 Disable	1	R/W		
		D7-4	-	reserved	-	-	-	-	0 when being read.
		D3	-	reserved	-	-	-	-	Do not set to 1.
		D2	SBRK	Software break enable	1 Enable 0 Disable	1	R/W		
		D1	IC	Instruction cache enable	1 Enable 0 Disable	0	R/W		
		D0	DC	Data cache enable	1 Enable 0 Disable	0	R/W		

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Cacheable Area Select Register (CCU_AREA)	0x302304 (32 bits)	D31-7	–	reserved	–	–	–	0 when being read.
		D6-4	ARIC[2:0]	Instruction cache area select	ARIC[2:0] Area 0x7 Area 22 0x6 Area 21 0x5 Area 20 0x4 Area 19 0x3 Area 18 0x2 Area 17 0x1 Areas 15 & 16 0x0 Area 14	0x0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2-0	ARDC[2:0]	Data cache area select	ARDC[2:0] Area 0x7 Area 22 0x6 Area 21 0x5 Area 20 0x4 Area 19 0x3 Area 18 0x2 Area 17 0x1 Areas 15 & 16 0x0 Area 14	0x0	R/W	
Cache Lock Register (CCU_LK)	0x302308 (32 bits)	D31-8	–	reserved	–	–	–	0 when being read.
		D7	LKPRI7	Interrupt level 7 cache-lock enable	1 Lock 0 Unlock	0 R/W	0	R/W
		D6	LKPRI6	Interrupt level 6 cache-lock enable	1 Lock 0 Unlock	0 R/W	0	R/W
		D5	LKPRI5	Interrupt level 5 cache-lock enable	1 Lock 0 Unlock	0 R/W	0	R/W
		D4	LKPRI4	Interrupt level 4 cache-lock enable	1 Lock 0 Unlock	0 R/W	0	R/W
		D3	LKPRI3	Interrupt level 3 cache-lock enable	1 Lock 0 Unlock	0 R/W	0	R/W
		D2	LKPRI2	Interrupt level 2 cache-lock enable	1 Lock 0 Unlock	0 R/W	0	R/W
		D1	LKPRI1	Interrupt level 1 cache-lock enable	1 Lock 0 Unlock	0 R/W	0	R/W
		D0	LKPRI0	Interrupt level 0 cache-lock enable	1 Lock 0 Unlock	0 R/W	0	R/W
Cache Status Register (CCU_STAT)	0x30230c (32 bits)	D31-4	–	reserved	–	–	–	0 when being read.
		D3	ICLKS	Instruction cache lock status	1 Locked 0 Not locked	X R	X	R
		D2	DCLKS	Data cache lock status	1 Locked 0 Not locked	X R	X	R
		D1	ICS	Instruction cache operating status	1 Active 0 Inactive	X R	X	R
		D0	DCS	Data cache operating status	1 Active 0 Inactive	X R	X	R
Cache Write Buffer Status Register (CCU_WB_STAT)	0x302318 (32 bits)	D31-10	–	reserved	–	–	–	0 when being read.
		D9	WEFINISH	Write-finish flag	1 Finished 0 Writing	1 R	1	R
		D8	WBEMPTY	Write buffer empty flag	1 Empty 0 Full	1 R	1	R
		D7-0	–	reserved	–	–	X	–
CCLK Division Ratio Select Register (CCU_CCLKDV)	0x302360 (32 bits)	D31-2	–	reserved	–	–	–	0 when being read.
		D1-0	CLK_DOWN[1:0]	CCLK division ratio select	CLK_DOWN[1:0] Division ratio 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	Source clock: MCLK

0x30240c–0x302925

Graphics Engine (GE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
GE Command Address Register (GE_CMD_ADDR)	0x30240c (32 bits)	D31-2	CMD_ADDR[31:2]	GE command address (Word boundary address)	0x0 to 0xfffffc	0x0	R/W		
		D1-0	CMD_ADDR[1:0]	Fixed at 0x0 (Cannot be altered.)	–	–	–	R	
GE Control Register (GE_CTL)	0x302440 (32 bits)	D31-17	–	reserved	–	–	–	0 when being read.	
		D16	GE_STS	GE operation status	1 Busy 0 Idle	0 R	0	R	
		D15-11	–	reserved	–	–	–	–	0 when being read.
		D10	BUS_STS	Bus operation status	1 Running 0 Stop	0 R	0	R	
		D9	CALC_STS	Calculator operation status	1 Running 0 Stop	0 R	0	R	
		D8	DRAW_STS	Pixel drawing status	1 Running 0 Stop	0 R	0	R	
		D7-4	–	reserved	–	–	–	–	0 when being read.
		D3	GE_STOP	GE stop control	1 Stop 0 –	0 R/W	0	R/W	
		D2	GE_RUN	GE run control	1 Run trigger 0 –	0 R/W	0	R/W	
		D1	GE_HRST	GE hot reset control	1 Reset 0 Normal mode	0 R/W	0	R/W	
D0	GE_CRST	GE cold reset control	1 Cold reset 0 Normal mode	1 R/W	0	R/W			
GE Interrupt Enable Register (GE_IE)	0x302444 (32 bits)	D31-17	–	reserved	–	–	–	0 when being read.	
		D16	GE_END_IE	GE end-of-execution interrupt enable	1 Enable 0 Disable	0 R/W	0	R/W	
		D15-9	–	reserved	–	–	–	–	0 when being read.
		D8	GE_ERR_IE0	Drawing error interrupt enable	1 Enable 0 Disable	0 R/W	0	R/W	
		D7-1	–	reserved	–	–	–	–	0 when being read.
D0	GE_ERR_IE1	Calculation error interrupt enable	1 Enable 0 Disable	0 R/W	0	R/W			

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
GE Interrupt Flag Register 1 (GE_IF1)	0x302448 (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.
		D3-0	CALC_ERR[3:0]	Calculation error status	CALC_ERR[3:0] Error 0xf-0x9 reserved 0x8 Decompress 0x7 Picture header 0x6 Font size 0x5 Radius 0x4 Circle location 0x3 Concave 0x2 Coordinates 0x1 Trace width 0x0 No error	0x0	R/W	Reset by writing 0x0.
GE Interrupt Flag Register 2 (GE_IF2)	0x302449 (8 bits)	D7-4	-	reserved	-	-	-	0 when being read.
		D3	DRAW_ERR3	No VRAM write error flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.
		D2	DRAW_ERR2	Color depth over error flag	1 Occurred 0 Not occurred	0	R/W	
		D1	DRAW_ERR1	Clipping area over error flag	1 Occurred 0 Not occurred	0	R/W	
D0	DRAW_ERR0	Work area over error flag	1 Occurred 0 Not occurred	0	R/W			
GE Interrupt Flag Register 3 (GE_IF3)	0x30244a (8 bits)	D7-3	-	reserved	-	-	-	0 when being read.
		D2-0	EXE_END[2:0]	Cause of termination	EXE_END[2:0] Cause 0x7 Unexpected end 0x6 Undefined CMD 0x5 Software 0x4 STOP4 CMD 0x3 STOP3 CMD 0x2 STOP2 CMD 0x1 STOP1 CMD 0x0 Not completed	0x0	R/W	Reset by writing 0x0.
VRAM Work Area Width Register (GE_REAL_W)	0x30244c (32 bits)	D31-13	-	reserved	-	-	-	0 when being read.
		D12-0	REAL_WIDTH[12:0]	Rotated work area width	Width = REAL_WIDTH (pixels)	0x20	R	
VRAM Work Area Start Address Register (GE_WK_ADDR)	0x302450 (32 bits)	D31-10	VWIN_ADDR[31:10]	Work area start address (1KB boundary address)	Areas 3-5, 7-10, 13-16, and 19-22	0x0	R/W	
		D9-0	VWIN_ADDR[9:0]	Fixed at 0x0 (Cannot be altered.)			R	
VRAM Work Area Size Register (GE_WK_SIZE)	0x302454 (32 bits)	D31-28	-	reserved	-	-	-	0 when being read.
		D27-16	VWIN_H[11:0]	Work area height	Height = VWIN_H + 1 (pixels)	0x0	R/W	
		D15-12	-	reserved	-	-	-	0 when being read.
		D11-0	VWIN_W[11:0]	Work area width	Width = VWIN_W + 1 (pixels)	0x1f	R/W	
Display Configuration Register (GE_DISP_CFG)	0x302458 (32 bits)	D31-17	-	reserved	-	-	-	0 when being read.
		D16	TF_TYPE	Block transfer type select	1 Pixel to Byte 0 Byte to Byte	0	R/W	
		D15-5	-	reserved	-	-	-	0 when being read.
		D4	SYNC_TYPE	LCDC horizontal/vertical sync type select	1 Vertical 0 Horizontal	0	R/W	
		D3	-	reserved	-	-	-	0 when being read.
		D2-0	DISP_BPP[2:0]	Color depth (Display data bit per pixel)	DISP_BPP[2:0] Color depth 0x7-0x5 reserved 0x4 16 bpp 0x3 8 bpp 0x2 4 bpp 0x1 2 bpp 0x0 1 bpp	0x0	R/W	
VRAM Rotation Control Register (GE_ROTATE)	0x30245c (32 bits)	D31-2	-	reserved	-	-	-	0 when being read.
		D1-0	VWIN_ROT[1:0]	VRAM rotation select	VWIN_ROT[1:0] Rotation angle 0x3 270° 0x2 180° 0x1 90° 0x0 0°	0x0	R/W	
Clipping Area Start Position Register (GE_CLIP_ST)	0x302460 (32 bits)	D31-28	-	reserved	-	-	-	0 when being read.
		D27-16	CLIP_UPL_Y[11:0]	Clipping area upper left corner Y coordinate	0 to 4,095	0x0	R/W	
		D15-12	-	reserved	-	-	-	0 when being read.
		D11-0	CLIP_UPL_X[11:0]	Clipping area upper left corner X coordinate	0 to 4,095	0x0	R/W	
Clipping Area End Position Register (GE_CLIP_END)	0x302464 (32 bits)	D31-28	-	reserved	-	-	-	0 when being read.
		D27-16	CLIP_LWR_Y[11:0]	Clipping area lower right corner Y coordinate	0 to 4,095	0x0	R/W	
		D15-12	-	reserved	-	-	-	0 when being read.
		D11-0	CLIP_LWR_X[11:0]	Clipping area lower right corner X coordinate	0 to 4,095	0x0	R/W	

APPENDIX A LIST OF I/O REGISTERS

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks	
Mesh Configuration Register (GE_MESH)	0x302468 (32 bits)	D31-20	-	reserved	-	-	-	0 when being read.	
		D19-18	MESH_RW [1:0]	Mesh row width	MESH_RW[1:0]	Width	0x0	R/W	
					0x3	4 pixels reserved			
		D17-16	MESH_CW [1:0]	Mesh column width	MESH_CW[1:0]	Width	0x0	R/W	
0x3	4 pixels reserved								
D15-0	MESH_COL [15:0]	Mesh color	0x0 to 0xffff	0x0	R/W				
Transparent Color Register (GE_MAGIC)	0x30246c (32 bits)	D31-16	-	reserved	-	-	-	0 when being read.	
		D15-0	MAGIC_COL [15:0]	Transparent color	0x0 to 0xffff	0x0	R/W		
Updated Area Start Position Register (GE_UPDT_ST)	0x302470 (32 bits)	D31-28	-	reserved	-	-	-	0 when being read.	
		D27-16	UPDT_UPL_Y [11:0]	Updated area upper left corner Y coordinate	0 to 4,095	0xff	R/W	Cleared by writing any data.	
					-	-	-	0 when being read.	
		D15-12	-	reserved	-	-	-	0 when being read.	
D11-0	UPDT_UPL_X [11:0]	Updated area upper left corner X coordinate	0 to 4,095	0xff	R/W	Cleared by writing any data.			
Updated Area End Position Register (GE_UPDT_END)	0x302474 (32 bits)	D31-28	-	reserved	-	-	-	0 when being read.	
		D27-16	UPDT_LWR_Y [11:0]	Updated area lower right corner Y coordinate	0 to 4,095	0x0	R/W	Cleared by writing any data.	
					-	-	-	0 when being read.	
		D15-12	-	reserved	-	-	-	0 when being read.	
D11-0	UPDT_LWR_X [11:0]	Updated area lower right corner X coordinate	0 to 4,095	0x0	R/W	Cleared by writing any data.			
Palette 1 (GE_PALETTE1)	0x302800 0x3028ff (8 bits)	D7-0	-	1/2/4/8-bit color data	0x0 to 0xff	X	R/W		
CCT1 4-bit Entries (GE_CCT1_4BIT)	0x302910 0x30291f (8 bits)	D7-0	-	CCT1 data (User defined 4 to 8 bpp conversion data)	0x0 to 0xff	X	R/W		
CCT1 2-bit Entries (GE_CCT1_2BIT)	0x302920 0x302923 (8 bits)	D7-0	-	CCT1 data (User defined 2 to 4/8 bpp conversion data)	0x0 to 0xff	X	R/W		
CCT1 1-bit Entries (GE_CCT1_1BIT)	0x302924 0x302925 (8 bits)	D7-0	-	CCT1 data (User defined 1 to 2/4/8 bpp conversion data)	0x0 to 0xff	X	R/W		

Appendix B Power Saving

Current consumption depends, to a large degree, on the CPU operating mode, operating clock frequency, and the peripheral circuits to be activated. This chapter summarizes the control to save power.

The following shows the clock systems that can be controlled with software and power saving control methods. For details of control registers and control methods, see the chapter for each module.

System sleep (disabling all clocks)

- Executing the slp instruction
Execute the slp instruction if all of the system can be stopped. In SLEEP mode, the CPU stops operating and the CMU stops supplying a clock to each functional module. Therefore, all peripheral circuits (except the OSC1 oscillator circuit and RTC) stop operating.
The CPU is reawaken from SLEEP mode by initial reset, RTC interrupt, #NMI signal, or other interrupt from an external device (port input interrupt).

System clock

- Selecting the clock source (CMU module)
Either OSC3, PLL, or OSC1 can be selected as the system clock source. If the application can process the task with a low-speed clock, select OSC1 as the system clock source to reduce current consumption.
- Disabling the OSC3 oscillator circuit (CMU module)
Using OSC1 for the system clock and disabling the OSC3 oscillator circuit achieves more reduction of current consumed.
- Selecting a low system clock (CMU module)
The CMU module provides a clock divider to set the system clock speed to 1/1 to 1/32 of the source clock (OSC3, PLL, OSC1). By running the S1C33L26 with the lowest speed required for the application's task, current consumption can be reduced.

CPU clock (CCLK)

- Executing the halt instruction
Execute the halt instruction if there is no task to be processed by the CPU such as when the display on the LCD is only required or when the CPU is waiting an interrupt. Although the CPU enters HALT mode and stops operating, the peripheral modules keep the status when the halt instruction is executed. So the LCD controller and the peripheral modules used to generate an interrupt can be made to be run. Power saving effect will be enhanced by disabling the unnecessary oscillator and peripheral modules before executing the halt instruction. The CPU reactivates from HALT mode by an interrupt from the ports or peripheral modules that are being operated in HALT mode.

Peripheral clocks

- Disabling peripheral clocks (CMU and PSC modules)
The peripheral clock supply can be disabled if the peripheral modules listed below can be placed in standby state.

Table B.1 Peripheral Modules and Operating Clocks

Clock	Clock enable bit	Peripheral modules
PCLK1	PCLK1_EN/CMU_CLKCTL register	<ul style="list-style-type: none"> • Prescaler (PSC Ch.0) • 8-bit programmable timer Ch.0, 2, 4, 6 (T8 Ch.0, 2, 4, 6) • 16-bit PWM timer Ch.0, 1 (T16A5 Ch.0, 1) • 16-bit audio PWM timer (T16P) • Universal serial interface (USI) • Serial interface Ch.0 (FSIO Ch.0) • A/D converter (ADC10) • I²S (I2S) • Misc registers (MISC)

APPENDIX B POWER SAVING

Clock	Clock enable bit	Peripheral modules
PCLK2	PCLK2_EN/CMU_CLKCTL register	<ul style="list-style-type: none"> • Prescaler (PSC Ch.1) • 8-bit programmable timer Ch.1, 3, 5, 7 (T8 Ch.1, 3, 5, 7) • Universal serial interface with LCD interface (USIL) • Serial interface Ch.1 (FSIO Ch.1) • Watchdog timer (WDT) • Remote controller (REMC) • Interrupt controller (ITC) • I/O ports (GPIO) • BBRAM • Cache controller (CCU) registers • Real-time clock (RTC) registers • SRAM controller (SRAMC) registers • SDRAM controller (SDRAMC) registers • LCD controller (LCDC) registers
GCLK	GCLK_EN/CMU_CLKCTL register	Graphics engine (GE)
LCLK	LCLK_EN/CMU_CLKCTL register	LCD controller (LCDC)
USBCLK	USBCLK_EN/CMU_CLKCTL register	USB function controller (USB)
USBREGCLK	USBREGCLK_EN/CMU_CLKCTL register	USB function controller (USB) registers
SDCLK	SDCLK_EN/CMU_CLKCTL register	<ul style="list-style-type: none"> • SRAM controller (SRAMC) • SDRAM controller (SDRAMC)
BCLK	BCLK_EN/CMU_CLKCTL register	<ul style="list-style-type: none"> • IVRAM (Area 3) • DSTRAM (Area 3) • SRAM controller (SRAMC) • SDRAM controller (SDRAMC) • DMA controller (DMAC) • LCD controller (LCDC) bus interface • Clock management unit (CMU) registers • Bus arbiters (Can be stopped in HALT mode.)

Table B.2 lists the clock control conditions and how to suspend/resume the CPU operation.

Table B.2 List of Clock Control Conditions

Current consumption	OSC1	OSC3/PLL	CPU (CCLK)	Peripherals	CPU suspending method	CPU resuming method
↑ Low	Oscillating	Stop	Stop	Stop	slp instruction	1
	Oscillating	Stop	Stop	Stop (only RTC is running)	slp instruction	1, 2
	Oscillating (System clock)	Stop	Stop	Stop (only RTC is running)	halt instruction	1, 2
	Oscillating (System clock)	Stop	Stop	Run	halt instruction	1, 2, 3
	Oscillating (System clock)	Stop	Run	Run		
	Stop	Oscillating (System clock)	Stop	Run	halt instruction	1, 3
	Oscillating	Oscillating (System clock)	Stop	Run	halt instruction	1, 2, 3
	Stop	Oscillating (System clock)	Run (with low-speed clock)	Run		
High ↓	Oscillating	Oscillating (System clock)	Run (with low-speed clock)	Run		
	Oscillating	Oscillating (System clock)	Run (OSC3•1/1) Run (PLL•1/1)	Run		

Clearing HALT and SLEEP modes (CPU resuming methods)

1. Resuming by a port input interrupt, #RESET or #NMI
The CPU resumes operating by occurrence of a cause of port input interrupt, #RESET, or #NMI.
2. Resuming by the RTC
The CPU resumes operating by occurrence of a cause of RTC interrupt.

3. Resuming by a peripheral or a debug interrupt (issuing an ICD forced break)

The CPU resumes operating by occurrence of a cause of interrupt in a peripheral whose interrupt is enabled. If the IE flag in the CPU has been set to 0, the CPU does not accept the interrupt request and starts executing the instructions that follow the halt instruction. If the IE flag has been set to 1, the CPU executes the interrupt handler.

Battery backup mode

- Turning the system power (LVDD, HVDD, AVDD, PLLVDD) off

If the system uses separated the system power and RTCVDD power sources, it is possible to operate only the RTCVDD system circuits (RTC, OSC1, and BBRAM) with the system power turned off to reduce current consumption. Turning the system power off reduces leakage current that cannot be reduced in SLEEP mode. The #STBY and WAKEUP pins that have been provided in the RTC module are used for controlling this function. Refer to the “Real-Time Clock (RTC)” chapter for more information on the control.

Note: The battery backup mode can help reduce current consumption when many parts are used in the external circuit or if the HVDD/AVDD system circuits will be deactivated for a relatively long time. Depending on the system configuration, the SLEEP mode may be efficient for saving power. Take these conditions into consideration at the system design stage.

Appendix C Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

Oscillator circuit

- Oscillation characteristics depend on factors such as components used (resonator, R_f , C_G , C_D) and circuit board patterns. In particular, with ceramic or crystal resonators, select the appropriate external resistor (R_f) and capacitors (C_G , C_D) only after fully evaluating components actually mounted on the circuit board.
- Oscillator clock disturbances caused by noise may cause malfunctions. To prevent such disturbances, consider the following points. The latest devices, in particular, are manufactured by microscopic processes, making them especially susceptible to noise.

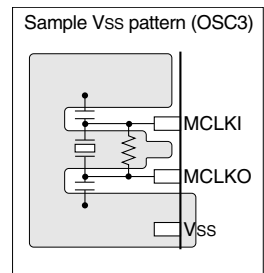
Areas in which noise countermeasures are especially important include the RTCCLKO pin and related circuit components and wiring. RTCCLKI pin handling is equally important. The noise precautions required for the RTCCLKI and RTCCLKO pins are described below. We also recommend applying similar noise countermeasures to the high-speed oscillator circuit, such as the MCLKI and MCKO pins and wiring.

- (1) Components such as a resonator, resistors, and capacitors connected to the RTCCLKI (MCLKI) and RTCCLKO (MCKO) pins should have the shortest connections possible.
- (2) Wherever possible, avoid locating digital signal lines within 3 mm of the RTCCLKI (MCLKI) and RTCCLKO (MCKO) pins or related circuit components and wiring. Rapidly-switching signals, in particular, should be kept at a distance from these components. Since the spacing between layers of multi-layer printed circuit boards is a mere 0.1 mm to 0.2 mm, the above precautions also apply when positioning digital signal lines on other layers.

Never place digital signal lines alongside such components or wiring, even if more than 3 mm distance or located on other layers. Avoid crossing wires.

- (3) Use Vss to shield RTCCLKI (MCLKI) and RTCCLKO (MCKO) pins and related wiring (including wiring for adjacent circuit board layers). Layers wired should be adequately shielded as shown to the right. Fully ground adjacent layers, where possible. At minimum, shield the area at least 5 mm around the above pins and wiring.

Even after implementing these precautions, avoid configuring digital signal lines in parallel, as described in (2) above. Avoid crossing even on discrete layers, except for lines carrying signals with low switching frequencies.



- (4) When an external clock is supplied to the RTCCLKI or MCLKI pin, the clock source should be connected to the RTCCLKI or MCLKI pin in the shortest line. Furthermore, do not connect anything else to the RTCCLKO or MCKO pin.
- (5) After implementing these precautions, check the output clock waveform by running the actual application program within the product. Use an oscilloscope to check outputs from the CMU_CLK pin.

You can check the quality of the OSC3 or PLL output waveform via the CMU_CLK output. Confirm that the frequency is as designed, is free of noise, and has minimal jitter.

You can also check the quality of the OSC1 waveform via the CMU_CLK output. In particular, enlarge the areas before and after the clock rising and falling edges and take special care to confirm that the regions approximately 100 ns to either side are free of clock or spiking noise.

Failure to observe precautions (1) to (3) adequately may lead to jitter in the OSC3 or PLL output and noise in the OSC1 output. Jitter in the OSC3 output will reduce operating frequencies, while noise in the OSC1 output will destabilize timers operated by the OSC1 clock as well as CPU Core operations when the system clock switches to OSC1.

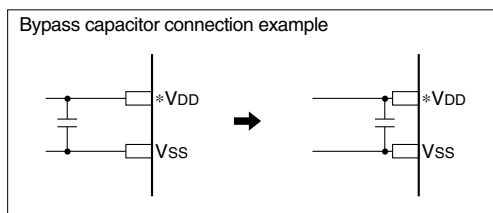
Reset circuit

- The reset signal input to the #RESET pin when power is turned on will vary, depending on various factors, such as power supply start-up time, components used, and circuit board patterns. Constants such as capacitance and resistance should be determined through testing with real-world products.
- Components such as capacitors and resistors connected to the #RESET pin should have the shortest connections possible to prevent noise-induced resets.

Power supply circuit

Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

- (1) Connections from the power supply to the LVDD, HVDD, PLLVDD, AVDD, PLLVSS, and VSS pins should be implemented via the shortest, thickest patterns possible. In particular, the power supply for AVDD affects A/D conversion precision.
- (2) If a bypass capacitor is connected between *VDD and VSS, connections between the *VDD and VSS pins should be as short as possible.

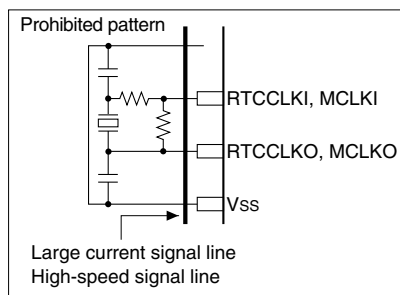


A/D Converter

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

Signal line location

- To prevent electromagnetically-induced noise arising from mutual induction, large-current signal lines should not be positioned close to circuits susceptible to noise, such as oscillators.
- Locating signal lines in parallel over significant distances or crossing signal lines operating at high speed will cause malfunctions due to noise generated by mutual interference. Specifically, avoid positioning crossing signal lines operating at high speed close to circuits susceptible to noise, such as oscillators and analog inputs.



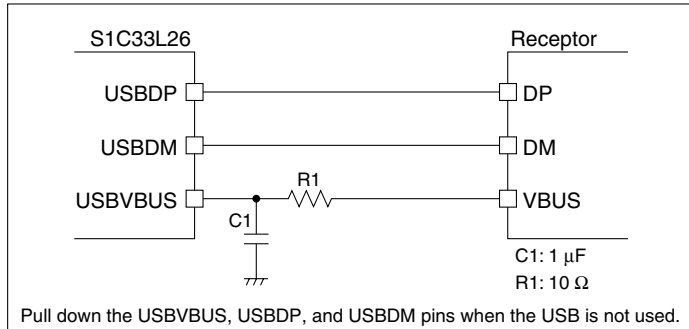
USB

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. The figure below shows an example of external connection.



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.

Noise-induced malfunctions

Check the following five points if you suspect the presence of noise-induced IC malfunctions.

(1) TEST 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 TEST 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

Low-level noise to this pin will cause a switch to debug mode. The switch to debug mode can be confirmed by the clock output from DCLK and a High signal from the DST2 pin.

For the product version, we recommend connecting the DSIO pin directly to HVDD or pulling up the DISO pin using a resistor not exceeding 10 k Ω . The IC includes an internal pull-up resistor. The resistor has a relatively high impedance of 100 k Ω to 500 k Ω and is not noise-resistant.

(3) #RESET pin

Low-level noise to this pin will reset the IC. Depending on the input waveform, the reset may not proceed correctly. This is more likely to occur if, due to circuit design choices, the impedance is high when the reset input is high.

(4) #NMI pin

Low-level noise to 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} and V_{SS} power supply

The IC will malfunction at the instant when noise falling below the rated voltage is input. Incorporate countermeasures on the circuit board, including close patterns for circuit board power supply circuits, noise-filtering decoupling capacitors, and surge/noise prevention components on the power supply line.

Perform the inspections described above using an oscilloscope capable of observing waveforms of at least 200 MHz. It may not be possible to observe high-speed noise events with a low-speed oscilloscope.

If you detect potential noise-induced malfunctions while observing the waveform with an oscilloscope, recheck with a low-impedance (less than 1 k Ω) resistor connecting the relevant pin to GND or to the power supply. Malfunctions at that pin are likely if changes are visible, such as the malfunction disappearing, becoming less frequent, or the phenomena changing.

The TEST, DSIO, #RESET, and #NMI input circuits described above detect input signal edges and are susceptible to malfunctions induced by spike noise. This makes these digital signal pins the most susceptible to noise. To reduce potential noise, keep the following two points in mind when designing circuit boards:

- (A) It is important to lower the signal-driving impedance, as described above. Connect pins to the power supply or GND, with impedance of 1 k Ω or less, preferably 0 Ω . The signal lines connected should be no longer than approximately 5 mm.
- (B) Parallel routing of 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 the digital lines. 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).

Handling of light (for bare chip mounting)

The characteristics of semiconductor components can vary when exposed to light. ICs may malfunction or non-volatile memory data may be corrupted if ICs are exposed to light.

Consider the following precautions for circuit boards and products in which this IC is mounted to prevent IC malfunctions attributable to light exposure.

- (1) Design and mount the product so that the IC is shielded from light during use.
- (2) Shield the IC from light during inspection processes.
- (3) Shield the IC on the upper, underside, and side faces of the IC chip.
- (4) Mount the IC chip within one week of opening the package. If the IC chip must be stored before mounting, take measures to ensure light shielding.
- (5) Adequate evaluations are required to assess nonvolatile memory data retention characteristics before product delivery if the product is subjected to heat stress exceeding regular reflow conditions during mounting processes.

Miscellaneous

This product series is manufactured using microscopic processes.

Although it is designed to ensure basic IC reliability meeting EIAJ and MIL standards, minor variations over time may result in electrical damage arising from disturbances in the form of voltages exceeding the absolute maximum rating when mounting the product in addition to physical damage. The following factors can give rise to these variations:

- (1) Electromagnetically-induced noise from industrial power supplies used in mounting reflow, reworking after mounting, and individual characteristic evaluation (testing) processes
- (2) Electromagnetically-induced noise from a solder iron when soldering

In particular, during soldering, take care to ensure that the soldering iron GND (tip potential) has the same potential as the IC GND.

Appendix D Boot

D.1 Boot Mode

The S1C33L26 supports the three boot modes listed below.

- NOR Flash/external ROM boot (Either 8 bits or 16 bits)
- SPI-EEPROM boot
- PC RS232C boot

The S1C33L26 boots up in the boot mode that can be selected with the BOOT and #CE10 pin configuration at initial reset.

Table D.1.1 Boot Mode Setting

BOOT pin	#CE10 pin	Boot mode	Program execution start address
0	Output	NOR Flash/external ROM	The system starts executing from the address written at address 0x20000000.
1	1 (Input)	SPI-EEPROM	The system loads MBR to IRAM (from address 0x100) and starts executing the code loaded.
	0 (Input)	PC RS232C	

Note: The #CE10 pin includes a pull-up resistor and it is enabled at initial reset. Note, however, that the #CE10 pin is configured as an input pin and its pull-up resistor is disabled in the initial process by the boot sequencer when the BOOT pin is set to 1. Therefore, connect an external pull-up or pull-down resistor to set the #CE10 pin input level to 1 or 0.

D.2 NOR Flash/External ROM Boot

D.2.1 Configuration of NOR Flash/External ROM Boot System

When the S1C33L26 is turned on or reset with the BOOT pin set to 0 (Vss), the S1C33L26 reads the reset vector from address 0x20000000 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 S1C33L26 supports booting from an 8-bit NOR Flash in contrast to the C33 PE Core that supports only a 16-bit device. The S1C33L26 reads the reset vector using the internal boot sequencer, 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.2.1.1 shows a NOR Flash/external ROM boot system connection diagram.

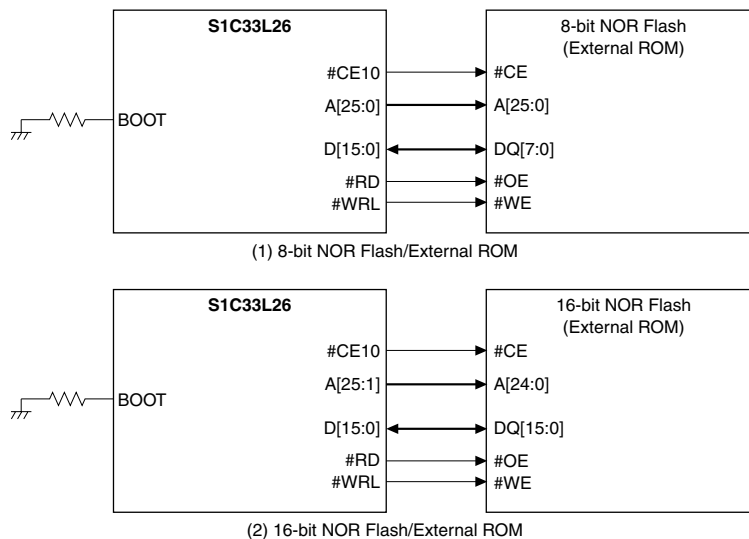


Figure D.2.1.1 NOR Flash/External ROM Boot System

This system uses only the external bus signals for #CE10 that are configured by default.

D.2.2 NOR Flash/External ROM Boot Sequence

Figure D.2.2.1 shows the NOR Flash/external ROM boot flowchart.

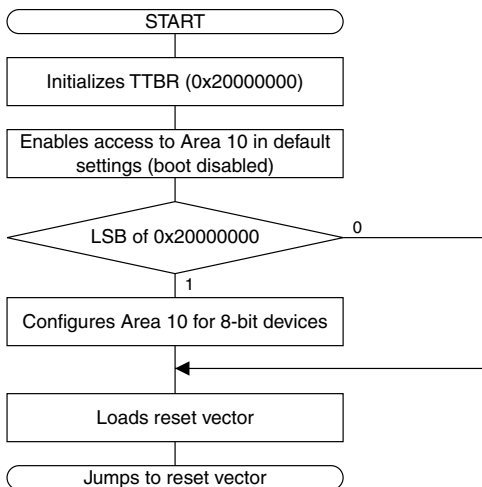


Figure D.2.2.1 NOR Flash/External ROM Boot Flowchart

- (1) When the BOOT pin is set to 0 at power-on or reset, the NOR Flash/external ROM boot sequence is executed.
- (2) Checks the LSB of the reset vector written at address 0x20000000.
- (3) Sets the #CE10 area device size to 8 bits if the LSB of the reset vector is 1. Leaves it unchanged (16 bits) if the LSB of the reset vector is 0.
- (4) Reads the reset vector again and jumps to that address.

Figures D.2.2.2 and D.2.2.3 show 16-bit and 8-bit NOR Flash boot sequences.

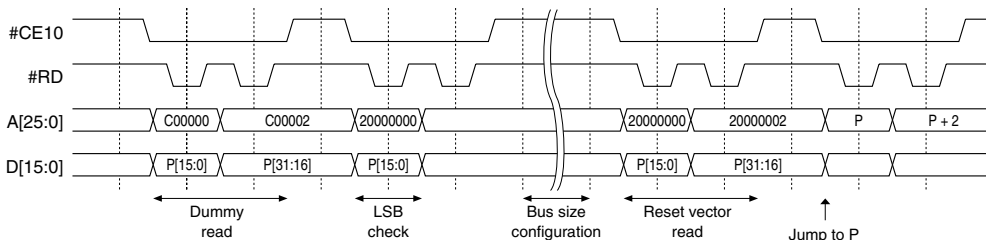


Figure D.2.2.2 16-bit NOR Flash Boot

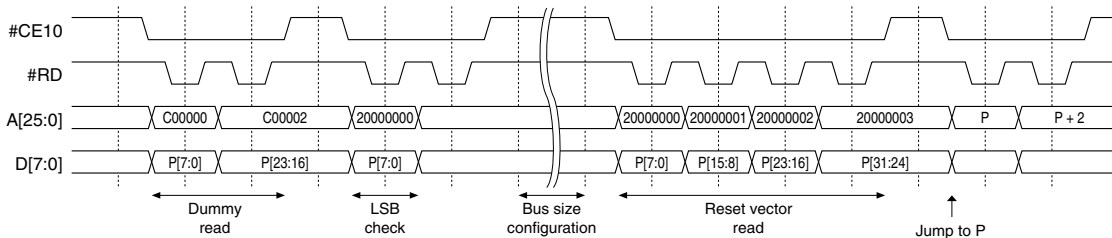


Figure D.2.2.3 8-bit NOR Flash Boot

D.2.3 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 0x20000000.

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 0x20000000.

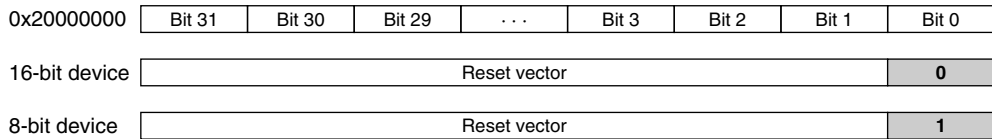


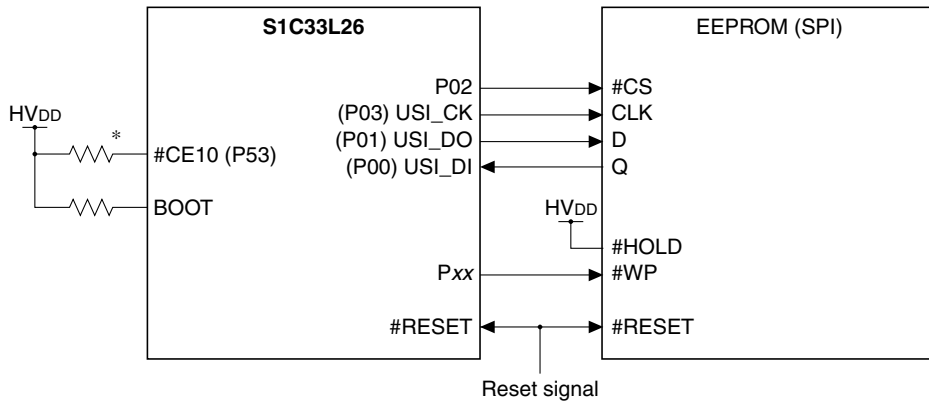
Figure D.2.3.1 Reset Vector for NOR Flash/External ROM Boot

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.3 SPI-EEPROM Boot

D.3.1 Configuration of SPI-EEPROM Boot System

When the S1C33L26 is turned on or reset with the BOOT and #CE10 pins set to 1 (HVDD), the S1C33L26 boots up by executing the MBR after loading it from the EEPROM, FRAM, or Serial Flash connected to the SPI bus to IRAM. Figure D.3.1.1 shows an SPI-EEPROM boot system connection diagram.



* The BOOT and #CE10 pins must be pulled up with external resistors, as the #CE10 (P53) internal pull-up resistor is disabled.

Figure D.3.1.1 SPI-EEPROM Boot System

Table D.3.1.1 Pins Used for SPI-EEPROM Boot

S1C33L26 pin	EEPROM pin	Pin status before booting	Pin status after booting
P02/USI_CS/SCLK1/REMC_O	#CS	Input	High output
P00/USI_DI/SIN1/#NAND_WR	Q	Input	Input
P01/USI_DO/SOUT1/#NAND_RD	D	Input	Output
P03/USI_CK/#SRDY1/REMC_I	CLK	Input	Low output
#RESET	#RESET	Input	Input
—	#HOLD*3	—	—
Pxx*1	#WP	(Input)	(Input)
#CE10/P53*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 pin.

*3 Fix at high.

The pins listed in the table are configured for USI (SPI master mode) (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 / 32 (e.g., 1.5 MHz when OSC3 = 48 MHz)

SPI mode: SCPOL = 0, SCPHA = 0

Data bit length: 8 bits

Master/slave mode: Master mode

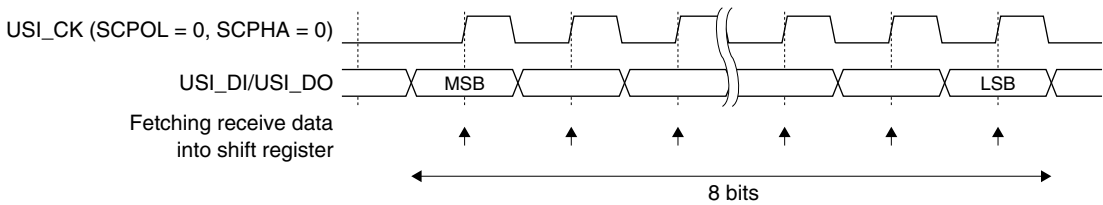


Figure D.3.1.2 SPI Mode

D.3.2 SPI-EEPROM Boot Sequence

Figure D.3.2.1 shows the SPI-EEPROM boot flowchart.

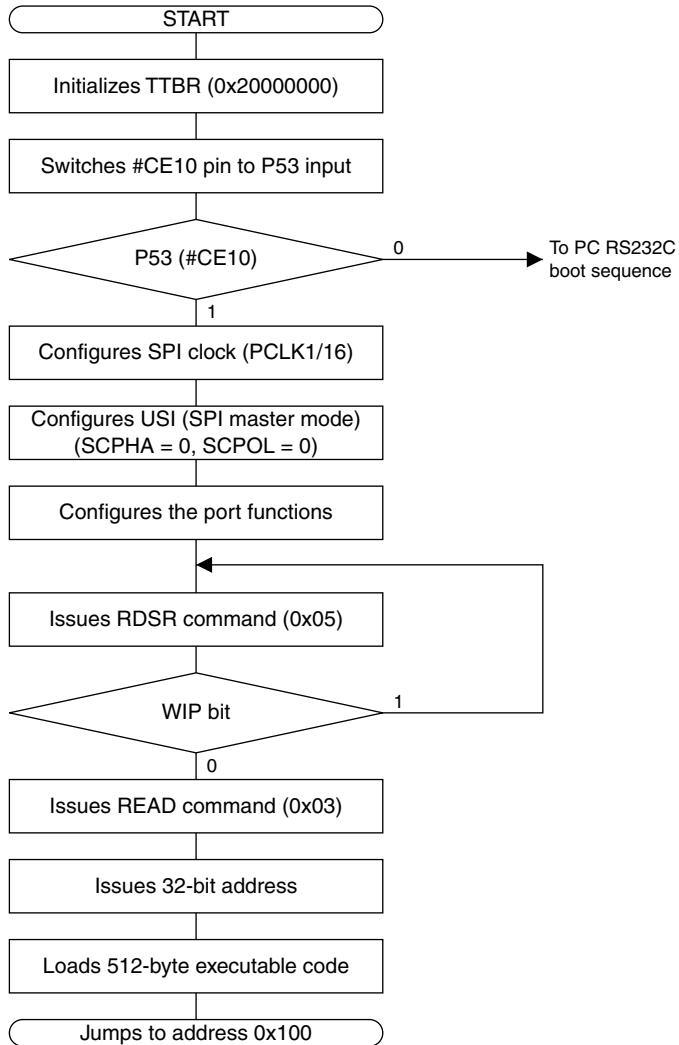


Figure D.3.2.1 SPI-EEPROM Boot Flowchart

- (1) When the BOOT and P53 (#CE10) pins are set to 1 at power-on or reset, the SPI-EEPROM boot sequence is executed.
- (2) The boot sequence configures the port and the USI 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 IRAM (0x100–).
- (6) Jumps to address 0x100 to execute the loaded codes.

Figure D.3.2.2 shows MBR reading start sequences for an SPI-EEPROM.

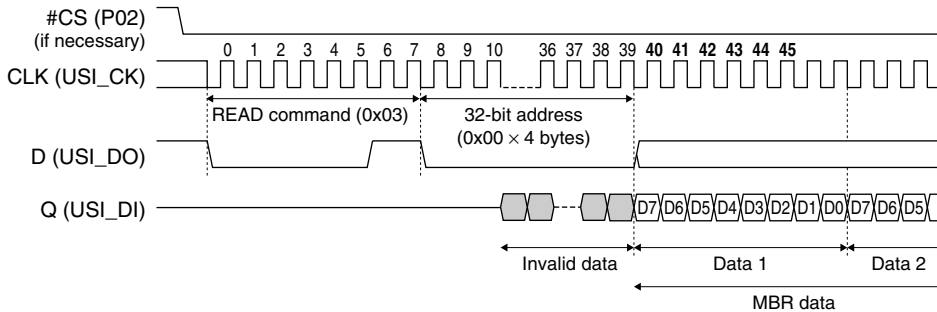


Figure D.3.2.2 EEPROM Read

D.3.3 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.3.3.1 and Figure D.3.3.1 show the data locations according to the EEPROM size.

Table D.3.3.1 EEPROM Size and Address Size

EEPROM size	Address size	MBR data location
1 to 256 bytes	1 byte	4th byte to 256th byte
0.25K to 64K bytes	2 bytes	3rd byte to 514th byte
64K to 16M bytes	3 bytes	2nd byte to 513th byte
16M to 4G bytes	4 bytes	1st byte to 512th byte

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.3.3.1 Data Location According to the EEPROM Size

D.4 PC RS232C Boot

D.4.1 Configuration of PC RS232C Boot System

When the S1C33L26 is turned on or reset with the BOOT pin set to 1 (HV_{DD}) and the #CE10 pin set to 0 (V_{SS}), the S1C33L26 boots up by executing the MBR after loading it from the PC (RS232C) to IRAM via FSIO Ch.1. Figure D.4.1.1 shows a PC RS232C boot system connection diagram.

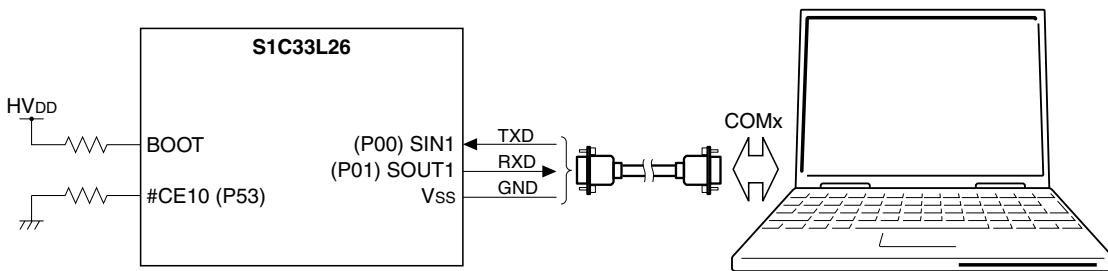


Figure D.4.1.1 PC RS232C Boot System

Table D.4.1.1 Pins Used for PC RS232C Boot

S1C33L26 pin	RS232C pin	Pin status before booting	Pin status after booting
P00/USI_DI/SIN1/#NAND_WR	TXD	Input	Input
P01/USI_DO/SOUT1/#NAND_RD	RXD	Input	Output
#CE10/P53 *1	–	Input (pulled up internally)	Input

*1 Used to select SPI-EEPROM boot or PC RS232C boot during boot mode configuration with the BOOT pin.

The pins listed in the table are configured for FSIO Ch.1 (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: 2 bit

Parity: None

Note: Table D.4.1.2 shows the maximum baud rate that can be set according to the S1C33L26 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.4.1.2 System Clock Frequency and Baud Rate

OSC3 frequency	> 500 kHz	> 4 MHz	> 7 MHz	> 15 MHz	> 29 MHz	> 40 MHz
Maximum baud rate (bps)	1200	9600	19200	38400	57600	115200

D.4.2 PC RS232C Boot Sequence

Figure D.4.2.1 shows the PC RS232C boot flowchart.

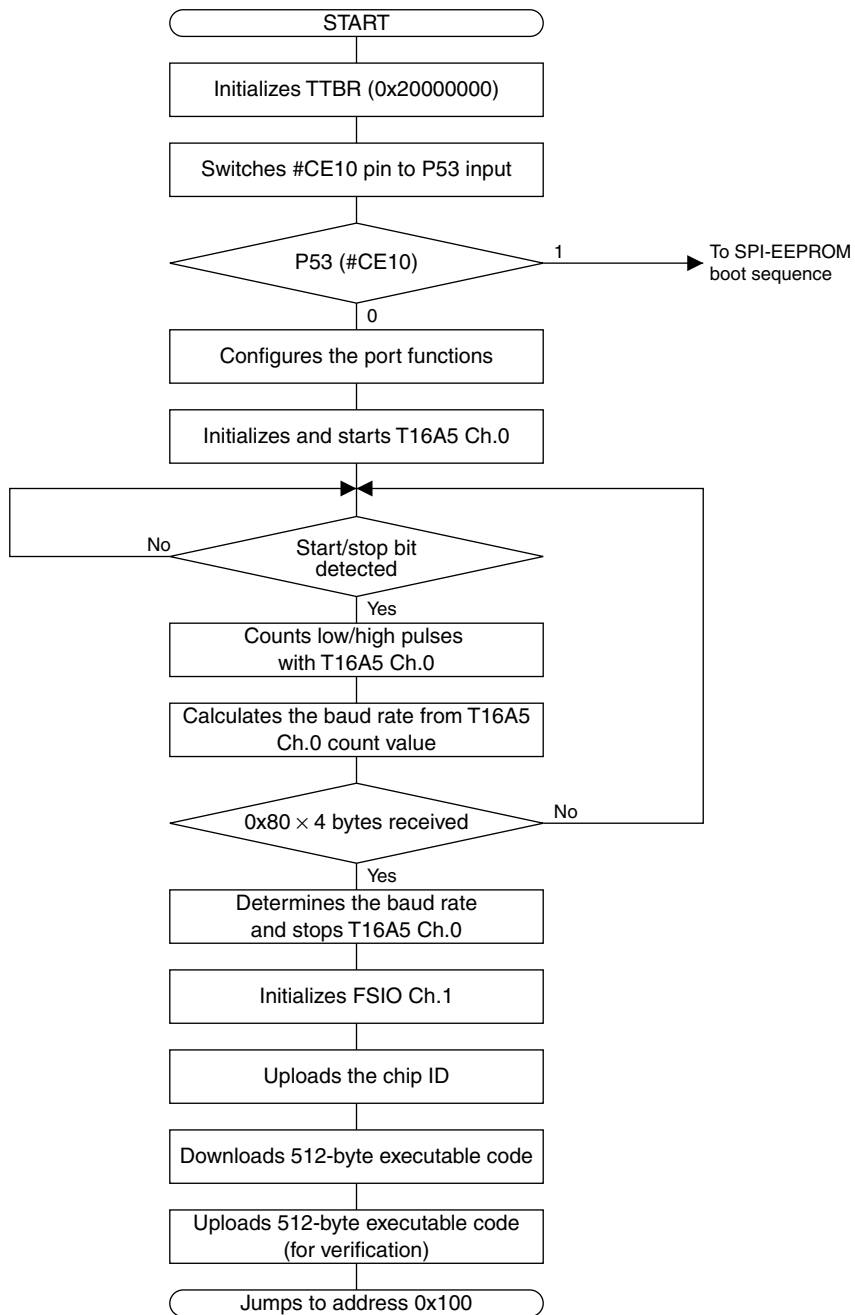


Figure D.4.2.1 PC RS232C Boot Flowchart

- (1) When the BOOT pin is set to 1 and the P53 (#CE10) pin is set to 0 at power-on or reset, the PC RS232C boot sequence is executed.
- (2) The boot sequence configures the port and the T16A5 Ch.0.
- (3) Waits for the P00 (SIN1) input pulled-down to 0 (start bit). When a start bit is input, the boot sequence starts T16A5 Ch.0 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 T16A5 Ch.0 count value, and FSIO Ch.1 is enabled with the calculated baud rate.
- (5) The S1C33L26 uploads a 4-byte chip ID code to the PC.
The PC sends 512 bytes of executable codes to the S1C33L26 after the chip ID is verified.

- (6) The S1C33L26 downloads the 512 bytes of executable codes.
The executable codes are loaded from the beginning of IRAM (0x100–).
- (7) The S1C33L26 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 0x100 to execute the loaded codes.

D.4.3 Transfer Data

First the PC sends 4 bytes of 0x80 to the S1C33L26. Then the PC sends 512-byte MBR data after verifying the 4-byte chip ID code received from the S1C33L26.

The S1C33L26 calculates the baud rate by counting the 4 bytes of 0x80 received from the PC and configures its serial interface. Then the S1C33L26 sends the 4-byte chip ID code to the PC. After the 512-byte MBR data is received, the S1C33L26 returns it to the PC for verification.

Figure D.4.3.1 shows the transfer data.

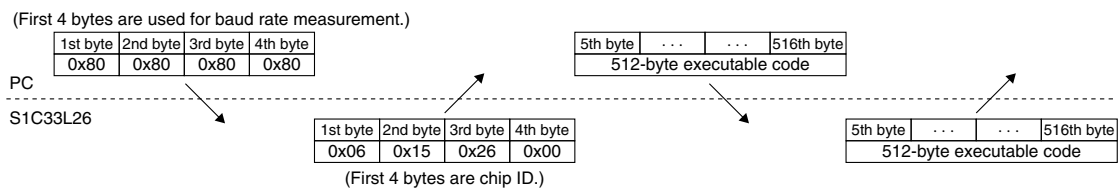


Figure D.4.3.1 Transfer Data for PC RS232C Boot

Revision History

Code No.	Page	Contents
411900100	All	New establishment
411900101	1-3	Features: SDRAMC (Old) • Equipped with a two-stage × 32-bit DQB (Data Queue Buffer). (New) • Equipped with a four-stage × 16-bit DQB (Data Queue Buffer).
	1-5	Features: USI/USIL (SPI mode) (Old) - Receive data mask function is available (master mode only). (New) Deleted
	1-6	Features: USB (Old) • Scratchable variable number of bulk end points (New) • Supports four general-purpose endpoints and endpoint 0 (control).
	1-12	Pin functions: Notes on pin name (Old) No description (New) - Be sure to avoid assigning the same signal to more than one pin.
	1-14	List of external bus pins: #CE10 (Old) * The #CE10 pull-up resistor enable/disable ... when the #CE10 pin is configured as an input pin. (New) * The #CE10 pull-up resistor is enabled (en) ... or disabled (dis) when the BOOT pin is set to 1.
	1-19	List of Other Pins: BOOT Modified Table 1.3.2.7
		USI/USIL pin configuration: USI_DO, USIL_DI, USIL_DO Modified Tables 1.3.2.8 and 1.3.2.9
	3-2	Memory map: Boot address (Old) Note: ... Note, however, ... Connect a pull-down resistor to the #CE10 pin to set the pin level to 0. (New) Note: ... Note, however, ... pull-up or pull-down resistor to set the #CE10 pin input level to 1 or 0.
	3-3	Memory map: IVRAM (Old) IVRAM located in Area 3 is accessed in five or six cycles, according to the ... register setting. (New) IVRAM located in Area 3 is accessed in four or five cycles, according to the ... register setting.
	4-3	Power supply: Power-on sequence (Old) (2) tPR: Power-on-reset time ... See “Electrical Characteristics” for the power-on-reset time. (New) (2) tPR: Power-on-reset time ... See the “#RESET Pin” section in the “Reset and NMI” chapter for the power-on-reset time.
		Power supply: Power-off sequence (Old) Power-off: ... 3. LVDD, PLLVDD (and RTCVDD) (May be turned off with 1 above at the same time.) Notes: • Applying only HVDD makes a diode circuit on the path from HVDD to AVDD ... • Be sure to avoid applying AVDD ... degraded due to flow-through current of the AVDD. (New) Power-off: ... 3. LVDD, PLLVDD (and RTCVDD) (May be turned off with 2 above at the same time.) Note: Be sure to avoid applying HVDD or AVDD ... due to flow-through current of the HVDD or AVDD.
	6-6	CMU: PLL setting examples Modified Table 6.4.4.4
	8-9	RTC: WAKEUP and #STBY pins (Old) Note that leakage currents flow from the RTCVDD system ... the #STBY pin is set to a high level. (New) Note that leakage current flows to the RTCVDD system ... the #STBY pin is set to a high level.
	9-3	SRAMC: Chip enable signals (Old) The SRAMC provides maximum 26 bits of ... access to a maximum 382M-byte address space. (New) The SRAMC provides maximum 26 bits of ... access to a maximum 336M-byte address space. Modified Table 9.4.1.1
9-10	SRAMC: SRAM read/write timings with external #WAIT (Old) Note: Figures 9.6.2.1 and 9.6.2.2 assume a very low operating speed ... • When a #CE _x signal is used to generate a #WAIT signal, set the conditions ... Operating clock frequency < 60 MHz: CE _x SETUP[1:0] = 0x1 and CE _x WAIT[3:0] = 0x2, or ... • When a #RD/#WRH/#WRL signal is used to generate a #WAIT signal, set the conditions ... Operating clock frequency < 60 MHz: CE _x WAIT[3:0] = 0x3 ... (New) Note: Figures 9.6.2.1 and 9.6.2.2 assume a very low operating speed ... • When a #CE _x signal is used to generate a #WAIT signal, set the conditions ... CE _x SETUP[1:0] = 0x1 and CE _x WAIT[3:0] = 0x2, or ... • When a #RD/#WRH/#WRL signal is used to generate a #WAIT signal, set the condition ... CE _x WAIT[3:0] = 0x3 ...	
	10-3	SDRAMC: SDRAM size selections and SDRAM address Modified Table 10.4.2.2

REVISION HISTORY

Code No.	Page	Contents
411900101	10-7	SDRAMC: Initializing SDRAM (Old) 1. Initializing the SDRAMC registers ... (4) SDRAMC_APP register Set the CAS latency. Also enable the double frequency mode and queue buffer if necessary. (New) 1. Initializing the SDRAMC registers ... (4) SDRAMC_APP register Set the CAS latency. Also enable double frequency mode if necessary.
	10-8	SDRAMC: Initializing SDRAM Modified Figure 10.5.1.1
	10-12, 10-13	SDRAMC: SDRAM refresh Modified Figures 10.5.5.1 and 10.5.5.2
	11-3, 11-8	CCU: Selecting area to be cached Modified Tables 11.3.2.1 and 11.7.2
	13-7, 13-13, 13-14, AP-A-54	DMAC: DMAC trigger source (Old) No description (New) Modified Tables 13.4.1 and 13.7.2 * Set the T16A5 channel for invoking the DMAC using DMASEL[1:0]/T16A_CTLx register. Modified the DMAC_TRG_SEL register table
	14-6	T8: T8 interrupt circuit Modified Figure 14.9.1
	15-5, 15-12	T16A5: Reading counter values (Old) Note: T16SEL[1:0]/T16A_CTLx register ... counter value will be read from the T16A_TCO register. (New) Note: The counter value must be read from ... selected using T16SEL[1:0]/T16A_CTLx register.
	18-1	USI: USI module overview - SPI master/slave mode (Old) - Receive data mask function is available (master mode only). (New) Deleted
	18-5	USI: Settings for SPI mode (Old) When used in SPI master mode, ... and enable or disable the receive data mask function. (New) When used in SPI master mode, select the clock mode and data length.
	18-6	USI: Receive data mask function (Old) Receive data mask function (master mode) The USI in SPI master mode provides a receive data mask (data retransmission) function. For normal data transfer, set SMSKEN to 0 (default) to disable the receive data mask function. (New) Deleted
	18-7	USI: Data transfer in UART mode - Data reception (Old) If the subsequent receive data is written to the receive data buffer when URDIF is 1, an overrun error occurs. (New) If the next reception is completed ... an overrun error occurs (at the time stop bit has been received).
	18-8	USI: Data receiving timing chart (UART mode) Modified Figure 18.5.1.2 USI: Data transmission timing chart (SPI mode) Modified Figure 18.5.2.1
	18-9	USI: Data transfer in SPI mode - Data reception (Old) If the subsequent receive data is written to the receive data buffer when SRDIF is 1, an overrun error occurs. (New) While SRDIF is set to 1, ... overrun error occurs at the time the first bit of the third byte is fetched). USI: Data receiving timing chart (SPI mode) Modified Figure 18.5.2.2
	18-11	USI: I ² C master data transmission timing chart Modified Figure 18.5.3.2
	18-11, 18-12	USI: Sending slave address and transfer direction bit (Old) ... In 10-bit mode, data is sent twice under software control. ... To send a 10-bit address, execute this procedure twice as shown in Figure 18.5.3.4. ... (New) ... In 10-bit mode, data is sent twice or three times under software control. ... Modified Figure 18.5.3.4 ... To send a 10-bit address, execute this procedure twice or three times as shown in Figure 18.5.3.4. ...
	18-14	USI: I ² C master data receiving timing chart Modified Figure 18.5.3.9
	18-15	USI: Control method in I ² C slave mode (Old) ... After an interrupt occurs, ... (ISSTA[2:0]/USI_ISIF register) to check the operation finished. (New) ... After an interrupt occurs, ... This also automatically reset ISSTA[2:0] to 0x0.
	18-16	USI: I ² C slave data transmission timing chart Modified Figure 18.5.3.12
	18-18	USI: I ² C slave data receiving timing chart Modified Figure 18.5.3.14

Code No.	Page	Contents
411900101	18-19, 18-20	<p>USI: Receive errors - Overrun error (Old) Overrun error (all interface modes) If data is received before the previously received data in the receive data buffer has not been read, ... The overrun error flag is reset to 0 by writing 1. (New) Overrun error (all interface modes) UART mode An overrun error occurs if the next reception is completed when URDIF is 1 and the receive data ... I²C master/slave mode ... To reset an overrun error, write 1 to IMEIF/ISEIF and then read the receive data buffer (USI_RD register) twice.</p>
	18-21	<p>USI: Interrupts in UART mode - Receive error interrupt (Old) ... If any of the error flags has the value 1, ... proceed with error recovery. (New) ... If any of the error flags has the value 1, ... proceed with error recovery. To reset an overrun error, ... USIMOD[2:0]/USI_GCFG register) to initialize USI.</p>
	18-22	<p>USI: Interrupts in SPI mode - Receive error interrupt (Old) ... If SEIF is 1, the interrupt handler routine will proceed with error recovery. (New) ... If SEIF is 1, the interrupt handler routine will proceed with error recovery. To reset an overrun error, clear SEIF ... then read the receive data buffer (USI_RD register) twice.</p>
	18-22, 18-23	<p>USI: Interrupts in I²C master/slave mode - Receive error interrupt (Old) To use this interrupt, ... interrupt requests for this cause will not be sent to the ITC. If IMEIF is 1, the interrupt handler routine will proceed with error recovery. (New) To use this interrupt, ... interrupt requests for this cause will not be sent to the ITC. An overrun error occurs ... two-byte data has been received without reading the receive data buffer. The USI module sets ... the interrupt handler routine will proceed with error recovery. To reset an overrun error, ... read the receive data buffer (USI_RD register) twice.</p>
	18-23, 18-31, 18-32, AP-A-3, AP-A-26	<p>USI: 0x30045f USI SPI Master Mode Receive Data Mask Register (USI_SMSK) (New) Deleted</p>
	18-25	<p>USI: USI Receive Data Buffer Register (USI_RD) - (D[7:0]) RD[7:0]: USI Receive Data Buffer Bits (Old) If receiving the subsequent data is completed ... the new received data overwrites the contents. (New) Deleted</p>
	18-27	<p>USI: USI UART Mode Interrupt Flag Register (USI_UIF) - (D2) UOEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... UOEIF is reset by writing 1. (New) An overrun error occurs ... (write 0x0 to USIMOD[2:0]/USI_GCFG register) to initialize USI.</p>
	18-28, 18-29, AP-A-26	<p>USI: USI SPI Master/Slave Mode Configuration Register (USI_SCFG) (Old) D1 SMSKEN: Receive Data Mask Enable Bit (New) D1 Reserved (Do not set to 1.)</p>
	18-30	<p>USI: USI SPI Master/Slave Mode Interrupt Flag Register (USI_SIF) - (D2) SEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... SEIF is reset by writing 1. (New) An overrun error occurs if data are received successively ... USI_RD register twice can be reversed.</p>
	18-33	<p>USI: USI I²C Master Mode Interrupt Flag Register (USI_IMIF) - (D[4:2]) IMSTA[2:0]: I²C Master Status Bits (Old) When an operation completion interrupt occurs, ... the operation that has been finished. (New) ... the operation that has been finished. IMSTA[2:0] is automatically reset to 0x0 by writing 1 to IMIF. USI: USI I²C Master Mode Interrupt Flag Register (USI_IMIF) - (D1) IMEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... IMEIF is reset by writing 1. (New) An overrun error occurs ... and then read the receive data buffer (USI_RD register) twice.</p>
	18-35	<p>USI: USI I²C Slave Mode Interrupt Flag Register (USI_ISIF) - (D[4:2]) ISSTA[2:0]: I²C Slave Status Bits (Old) When an operation completion interrupt occurs, ... the operation that has been finished. (New) ... the operation that has been finished. ISSTA[2:0] is automatically reset to 0x0 by writing 1 to ISIF. USI: USI I²C Slave Mode Interrupt Flag Register (USI_ISIF) - (D1) ISEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... ISEIF is reset by writing 1. (New) An overrun error occurs ... and then read the receive data buffer (USI_RD register) twice.</p>
	19-1	<p>USIL: USIL module overview - SPI master/slave mode (Old) - Receive data mask function is available (master mode only). (New) Deleted</p>
	19-3	<p>USIL: Transfer clock (Old) When the USIL is configured to a UART, SPI master (normal mode), I²C master, or LCD SPI, ... When the USIL is configured to an SPI master (fast mode) or LCD parallel interface, PCLK2 is used as the source clock. (New) When the USIL is configured to a UART, SPI master (normal mode), I²C master, LCD SPI, or LCD parallel interface, the source clock for transfer is supplied by the 8-bit programmable timer ... When the USIL is configured to an SPI master (fast mode), PCLK2 is used as the source clock.</p>
	19-6	<p>USIL: Settings for SPI mode (Old) When used in SPI master mode, select the clock mode and enable or disable the receive data mask function. (New) When used in SPI master mode, select the clock mode.</p>

REVISION HISTORY

Code No.	Page	Contents
411900101	19-7	USIL: Receive data mask function (Old) Receive data mask function (master mode) The USIL in SPI master mode provides a receive data mask (data retransmission) function. For normal data transfer, set SMSKEN to 0 (default) to disable the receive data mask function. (New) Deleted
	19-11	USIL: Data transfer in UART mode - Data reception (Old) If the subsequent receive data is written to the receive data buffer when URDIF is 1, an overrun error occurs. (New) If the next reception is completed ... an overrun error occurs (at the time stop bit has been received). USIL: Data receiving timing chart (UART mode) Modified Figure 19.5.1.2
	19-12	USIL: Data transmission timing chart (SPI mode) Modified Figure 19.5.2.1 USIL: Data transfer in SPI mode - Data reception (Old) If the subsequent receive data is written to the receive data buffer when SRDIF is 1, an overrun error occurs. (New) While SRDIF is set to 1, ... overrun error occurs at the time the first bit of the third byte is fetched).
	19-13	USIL: Data receiving timing chart (SPI mode) Modified Figure 19.5.2.2
	19-14, 19-15	USIL: I ² C master data transmission timing chart Modified Figure 19.5.3.2
	19-15, 19-16	USIL: Sending slave address and transfer direction bit (Old) ... In 10-bit mode, data is sent twice under software control. ... To send a 10-bit address, execute this procedure twice as shown in Figure 19.5.3.4. ... (New) ... In 10-bit mode, data is sent twice or three times under software control. ... Modified Figure 19.5.3.4 ... To send a 10-bit address, execute this procedure twice or three times as shown in Figure 19.5.3.4. ...
	19-17, 19-18	USIL: I ² C master data receiving timing chart Modified Figure 19.5.3.9
	19-19	USIL: Control method in I ² C slave mode (Old) ... After an interrupt occurs, ... (ISSTA[2:0]/USIL_ISIF register) to check the operation finished. (New) ... After an interrupt occurs, ... This also automatically reset ISSTA[2:0] to 0x0.
	19-20	USIL: I ² C slave data transmission timing chart Modified Figure 19.5.3.12
	19-22	USIL: I ² C slave data receiving timing chart Modified Figure 19.5.3.14
	19-23	USIL: Data Transmission in LCD SPI Mode (Old) The LSBSY flag indicates the USIL status ... the LCD SPI controller is operating or at standby. (New) The LSBSY flag indicates the USIL status ... LSDMOD[1:0]/USIL_LSDCFG register has completed.
	19-24	USIL: Figure 19.5.4.1 (Old) Data Transmission Timing Chart (LCD SPI mode) (New) Data Transmission Timing Chart (LCD SPI mode, 16-bit data format) USIL: Data Transfer in LCD Parallel Mode - Data write (Old) To write data to the LCD driver/panel ... the command bit status (LPCMD/USIL_LPCFG register). The LCD parallel interface asserts the chip enable and write signals and outputs the buffer data ... The command bit status is output from the USIL_DI pin. The transmitter circuit includes ... (New) To write data to the LCD driver/panel ... the command bit status (LPCMD/USIL_LPCFG register). The command bit must be set before ... the USIL_DI pin immediately after it is written to the register. The LCD parallel interface asserts the chip enable signal and outputs the buffer data ... The transmitter circuit includes ... USIL: Data-Write Timing Chart (LCD parallel mode) Modified Figure 19.5.5.1
	19-25, 19-48	USIL: Data Transfer in LCD Parallel Mode - Data read (Old) The LCD parallel interface asserts the chip enable and read signals ... via the USIL_DI pin. (New) Set the command bit ... output from the USIL_DI pin immediately after it is written to the register.
	19-27	USIL: Receive errors - Overrun error (Old) Overrun error Overrun error (UART, SPI, I ² C modes) If data is received before the previously received data in the receive data buffer has not been read, ... The overrun error flag is reset to 0 by writing 1. (New) Overrun error (UART, SPI, I ² C master/slave modes) UART mode An overrun error occurs if the next reception is completed when URDIF is 1 and the receive data ... I ² C master/slave mode ... To reset an overrun error, write 1 to IMEIF/ISEIF and then read the receive data buffer (USIL_RD register) twice.

Code No.	Page	Contents
411900101	19-28	USIL: Interrupts in UART mode - Receive error interrupt (Old) ... If any of the error flags has the value 1, ... proceed with error recovery. (New) ... If any of the error flags has the value 1, ... proceed with error recovery. To reset an overrun error, ... USILMOD[2:0]/USIL_GCFG register) to initialize USIL.
	19-29	USIL: Interrupts in SPI mode - Receive error interrupt (Old) ... If SEIF is 1, the interrupt handler routine will proceed with error recovery. (New) ... If SEIF is 1, the interrupt handler routine will proceed with error recovery. To reset an overrun error, clear SEIF ... then read the receive data buffer (USIL_RD register) twice.
	19-30	USIL: Interrupts in I ² C master/slave mode - Receive error interrupt (Old) To use this interrupt, ... interrupt requests for this cause will not be sent to the ITC. If IMEIF is 1, the interrupt handler routine will proceed with error recovery. (New) To use this interrupt, ... interrupt requests for this cause will not be sent to the ITC. An overrun error occurs ... two-byte data has been received without reading the receive data buffer. The USIL module sets ... the interrupt handler routine will proceed with error recovery. To reset an overrun error, ... read the receive data buffer (USIL_RD register) twice.
	19-32, 19-38, AP-A-3, AP-A-28	USIL: 0x30065f USIL SPI Master Mode Receive Data Mask Register (USIL_SMSK) (New) Deleted
	19-33	USIL: USIL Receive Data Buffer Register (USIL_RD) - (D[7:0]) RD[7:0]: USIL Receive Data Buffer Bits (Old) If receiving the subsequent data is completed ... the new received data overwrites the contents. (New) Deleted
	19-36	USIL: USIL UART Mode Interrupt Flag Register (USIL_UIF) - (D2) UOEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... UOEIF is reset by writing 1. (New) An overrun error occurs ... (write 0x0 to USILMOD[2:0]/USIL_GCFG register) to initialize USIL.
	19-36, 19-37, AP-A-28	USIL: USIL SPI Master/Slave Mode Configuration Register (USIL_SCFG) (Old) D1 SMSKEN: Receive Data Mask Enable Bit (New) D1 Reserved (Do not set to 1.)
	19-38, 19-39	USIL: USIL SPI Master/Slave Mode Interrupt Flag Register (USIL_SIF) - (D2) SEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... SEIF is reset by writing 1. (New) An overrun error occurs if data are received successively when SRDIF is 1. While SRDIF is set ... The procedure that writes 1 to SEIF and reads USIL_RD register twice can be reversed.
	19-41	USIL: USIL I ² C Master Mode Interrupt Flag Register (USIL_IMIF) - (D[4:2]) IMSTA[2:0]: I ² C Master Status Bits (Old) When an operation completion interrupt occurs, ... the operation that has been finished. (New)... the operation that has been finished. IMSTA[2:0] is automatically reset to 0x0 by writing 1 to IMIF. USIL: USIL I ² C Master Mode Interrupt Flag Register (USIL_IMIF) - (D1) IMEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... IMEIF is reset by writing 1. (New) An overrun error occurs ... and then read the receive data buffer (USIL_RD register) twice.
	19-43	USIL: USIL I ² C Slave Mode Interrupt Flag Register (USIL_ISIF) - (D[4:2]) ISSTA[2:0]: I ² C Slave Status Bits (Old) When an operation completion interrupt occurs, ... the operation that has been finished. (New)... the operation that has been finished. ISSTA[2:0] is automatically reset to 0x0 by writing 1 to ISIF. USIL: USIL I ² C Slave Mode Interrupt Flag Register (USIL_ISIF) - (D1) ISEIF: Overrun Error Flag Bit (Old) An overrun errors occurs when the previous received data ... ISEIF is reset by writing 1. (New) An overrun error occurs ... and then read the receive data buffer (USIL_RD register) twice.
	19-45	USIL: USIL LCD SPI Mode Interrupt Flag Register (USIL_LSIF) - (D1) LSBYS: Transfer Busy Flag Bit (Old) It is cleared to 0 once the transfer is completed. (New) It is cleared to 0 after data transfer ... LSDMOD[1:0]/USIL_LSDCFG register has completed.
	21-4, 21-17	I ² S: Sample clock period Modified Figures 21.4.2 and 21.7.8
	21-10	I ² S: Data output timing chart Modified Figure 21.5.2
	21-18	I ² S: I ² S Start/Stop Register (I2S_START) - (D7) I2SBUSY: I ² S Busy Flag Bit (Old) I2SBUSY is set to 1 when the I ² S starts data output and stays 1 while data is being output. (New) I2SBUSY is set to 1 when 1 is written to I2SSTART and stays 1 while data is being output.
24-7, 24-17	GPIO: Interrupt mode and polarity selection (Old) In SLEEP mode, the CMU senses the port interrupt ... if edge trigger mode is selected. (New) SLEEP mode can be canceled ... how the GPIO interrupt mode (edge trigger/level trigger) is set.	
26-2	LCDC: Block diagram Modified Figure 26.2.1 LCDC: Block diagram (Old) SAPB bus interface The C33 PE Core accesses the LCDC registers and look-up table through ... Sequence controller The sequence controller controls data flow ... through the look-up table. ... (New) SAPB bus interface The C33 PE Core accesses the LCDC registers and monochrome look-up table through ... Sequence controller The sequence controller controls data flow ... through the color look-up table. ...	

REVISION HISTORY

Code No.	Page	Contents
411900101	26-13	LCDC: HR-TFT Panel Timing Parameters (Old) LCFC_FPFR register (New) LCDC_FPFR register
	26-31	LCDC: Inverting and blanking the display (Old) 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. (New) This is accomplished by inverting the display data output from the LCDC, rather than by inverting the pixel data in the display memory.
	26-46	LCDC: LCDC Display Mode Register (LCDC_DISPMD) - (D25) SWINV: Software Video Invert Bit (Old) Inverse operation is applied to output of the look-up tables, and does not affect the display memory. (New) Inverse operation is applied to the LCDC output, and does not affect the display memory.
	27-4	GE: Relationship between LCD display and work area (Old) For correspondence between GE and LCDC settings, see Section 27.8. (New) For correspondence between GE and LCDC settings, see Section 27.7.
	28-2	USB: USB Operating Clocks (Old) This clock can be stopped after USB control register settings have been finished. (New) Deleted
	28-27	USB: SIE_IntStat (SIE Interrupt Status) - (D6) NonJ (Old) This bit is valid when the InSUSPEND bit of the USB_Control register is set to 1. (New) This bit is valid when the InSUSPEND bit of the USB_Control register is set to 1. This bit is valid during snooze as well.
	31-1	Electrical characteristics: Absolute maximum rating (Old) No description (New) *2) The maximum input voltage range of the #STBY pin is Vss - 0.3 V to 4.0 V.
		Electrical characteristics: Recommended operating conditions (Old) No description (New) *1) $HV_{DD}/AV_{DD} \geq LV_{DD}/RTC_{VDD}/PLL_{VDD}$ $LV_{DD} = RTC_{VDD} = PLL_{VDD}$ *2) The recommended input voltage range of the #STBY pin is Vss - 0.3 V to 3.6 V.
	32-1	Basic external connection diagram: #STBY Modified the figure
	AP-B-2	Power Saving: List of clock control conditions Modified Table B.2
	AP-C-3	Mounting precautions: Precautions on VBUS Modified the figure
	AP-D-1	Boot: Boot mode (Old) Note: ... Note, however, ... Connect a pull-down resistor to the #CE10 pin to set the pin level to 0. (New) Note: ... Note, however, ... pull-up or pull-down resistor to set the #CE10 pin input level to 1 or 0.
	AP-D-3	Boot: Configuration of SPI-EEPROM boot system (Old) When the S1C33L26 is turned on or reset with both the BOOT and #CE10 pins left open (or set to 1), ... (New) When the S1C33L26 is turned on or reset with the BOOT and #CE10 pins set to 1 (HVDD), ... Modified Figure D.3.1.1
AP-D-6	Boot: Configuration of PC RS232C boot system (Old) When the S1C33L26 is turned on or reset with the BOOT pin left open (or set to 1) ... (New) When the S1C33L26 is turned on or reset with the BOOT pin set to 1 (HVDD) ... Modified Figure D.4.1.1	

AMERICA

EPSON ELECTRONICS AMERICA, INC.

2580 Orchard Parkway,
San Jose, CA 95131, USA
Phone: +1-800-228-3964 Fax: +1-408-922-0238

EUROPE

EPSON EUROPE ELECTRONICS GmbH

Riesstrasse 15, 80992 Munich,
GERMANY
Phone: +49-89-14005-0 Fax: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.

7F, Jinbao Bldg., No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, CHINA
Phone: +86-10-8522-1199 Fax: +86-10-8522-1125

SHANGHAI BRANCH

7F, Block B, Hi-Tech Bldg., 900 Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5577 Fax: +86-21-5423-4677

SHENZHEN BRANCH

12F, Dawning Mansion, Keji South 12th Road,
Hi-Tech Park, Shenzhen 518057, CHINA
Phone: +86-755-2699-3828 Fax: +86-755-2699-3838

EPSON HONG KONG LTD.

20/F, Harbour Centre, 25 Harbour Road,
Wanchai, Hong Kong
Phone: +852-2585-4600 Fax: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON TAIWAN TECHNOLOGY & TRADING LTD.

14F, No. 7, Song Ren Road,
Taipei 110, TAIWAN
Phone: +886-2-8786-6688 Fax: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.

1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 Fax: +65-6271-3182

SEIKO EPSON CORP.**KOREA OFFICE**

5F, KLI 63 Bldg., 60 Yoido-dong,
Youngdeungpo-Ku, Seoul 150-763, KOREA
Phone: +82-2-784-6027 Fax: +82-2-767-3677

SEIKO EPSON CORP.**MICRODEVICES OPERATIONS DIVISION****Device Sales & Marketing Dept.**

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 Fax: +81-42-587-5117