

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17W11

Technical Manual

NOTICE: PLEASE READ THE FOLLOWING NOTICE CAREFULLY BEFORE USING THIS DOCUMENT

The contents of this document are subject to change without notice.

1. This document may not be copied, reproduced, or used for any other purpose, in whole or in part, without the consent of the Seiko Epson Corporation (“Epson”).
2. Before purchasing or using Epson products, please contact our sales representative for the latest information and always be sure to check the latest information published on Epson’s official web sites and other sources.
3. Information provided in this document such as application circuits, programs, usage, etc., are for reference purposes only. Using the application circuits, programs, usage, etc. in the design of your equipment or systems is your own responsibility. Epson makes no guarantees against any infringements or damages to any third parties’ intellectual property rights or any other rights resulting from the information. This document does not grant you any licenses, intellectual property rights or any other rights with respect to Epson products owned by Epson or any third parties.
4. Epson is committed to constantly improving quality and reliability, but semiconductor products in general are subject to malfunction and failure. By using Epson products, you shall be responsible for your hardware. Software and systems must be designed well enough to prevent death or injury as well as any property damage even if any of the malfunctions or failures might be caused by Epson products. When designing your products using Epson products, please be sure to check and comply with the latest information regarding Epson products (this document, specifications, data sheets, manuals, Epson’s web site, etc.). When using the information included above materials such as product data, charts, technical contents, programs, algorithms and application circuit examples, you shall evaluate your products both on a stand-alone basis as well as within your overall systems. You shall be solely responsible for deciding whether or not to adopt and use Epson products.
5. Epson has prepared this document and programs provided in this document carefully to be accurate and dependable, but Epson does not guarantee that the information and the programs are always accurate and complete. Epson assumes no responsibility for any damages which you incur due to misinformation in this document and the programs.
6. No dismantling, analysis, reverse engineering, modification, alteration, adaptation, reproduction, etc., of Epson products is allowed.
7. Epson products have been designed, developed and manufactured to be used in general electronic applications (office equipment, communications equipment, measuring instruments, home electronics, etc.) (“General Purpose”) and applications which is individually listed in this document or designated by Epson (“Designated Purpose”). Epson products are NOT intended for any use beyond the General Purpose and Designated Purpose uses that requires particular/higher quality or reliability in order to refrain from causing any malfunction or failure leading to death, injury, serious property damage or severe impact on society, including, but not limited to those listed below (“Particular Purpose”). Therefore, you are advised to use Epson products only for General Purpose and Designated Purpose uses. Should you desire to buy and use Epson products for a Particular Purpose, Epson makes no warranty and disclaims with respect to Epson products, whether express or implied, including without limitation any implied warranty of merchantability or fitness for any Particular Purpose. Please be sure to contact our sales representative and obtain approval in advance.
 【Examples of Particular Purpose】
 Space equipment (artificial satellites, rockets, etc.) /
 Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.) /
 Medical equipment / Relay equipment to be placed on ocean floor /
 Power station control equipment / Disaster or crime prevention equipment / Traffic control equipment / Financial equipment
 Other applications requiring similar levels of reliability as those listed above. Please be sure to contact our sales representative for details of the other applications.
8. Epson products listed in this document and our associated technologies shall not be used in any equipment or systems that laws and regulations in Japan or any other countries prohibit to manufacture, use or sell. Furthermore, Epson products and our associated technologies shall not be used for developing weapons of mass destruction, or any other military purposes or applications. If exporting Epson products or our associated technologies, you shall comply with the Foreign Exchange and Foreign Trade Control Act in Japan, Export Administration Regulations in the U.S.A. (EAR) and other export-related laws and regulations in Japan and any other countries and follow the required procedures as provided by the relevant laws and regulations.
9. Epson assumes no responsibility for any damages (whether direct or indirect) caused by or in relation with your non-compliance with the terms and conditions in this document.
10. Epson assumes no responsibility for any damages (whether direct or indirect) incurred by any third party that you assign, transfer, loan, etc., Epson products to.
11. For more details or other concerns about this document, please contact our sales representative.
12. Company names and product names listed in this document are trademarks or registered trademarks of their respective companies.

Evaluation board/kit and Development tool important notice

1. Epson evaluation board/kit or development tool is designed for use for engineering evaluation, demonstration, or development purposes only. Do not use it for other purposes. It is not intended to meet the requirements of design for finished products.
2. Epson evaluation board/kit or development tool is intended for use by an electronic engineer and is not a consumer product. The user should use it properly and in a safe manner. Epson does not assume any responsibility or liability of any kind of damage and/or fire caused by the use of it. The user should cease to use it when any abnormal issue occurs even during proper and safe use.
3. The part used for Epson evaluation board/kit or development tool may be changed without any notice.

Rev. e1.5, 2025.7

Preface

This is a technical manual for designers and programmers who develop a product using the S1C17W11. This document describes the functions of the IC, embedded peripheral circuit operations, and their control methods.

For the CPU functions and instructions, refer to the “S1C17 Family S1C17 Core Manual.” For the functions and operations of the debugging tools, refer to the respective tool manuals. (Our “Products: Document Downloads” website provides the downloadable manuals.)

Notational conventions and symbols in this manual

Register address

Peripheral circuit chapters do not provide control register addresses. Refer to “Peripheral Circuit Area” in the “Memory and Bus” chapter or “List of Peripheral Circuit Control Registers” in the Appendix.

Register and control bit names

In this manual, the register and control bit names are described as shown below to distinguish from signal and pin names.

XXX register: Represents a register including its all bits.

XXX.YYY bit: Represents the one control bit YYY in the XXX register.

XXX.ZZZ[1:0] bits: Represents the two control bits ZZZ1 and ZZZ0 in the XXX register.

Register table contents and symbols

Initial: Value set at initialization

Reset: Initialization condition. The initialization condition depends on the reset group (H0, H1, or S0). For more information on the reset groups, refer to “Initialization Conditions (Reset Groups)” in the “Power Supply, Reset, and Clocks” chapter.

R/W: R = Read only bit

W = Write only bit

WP = Write only bit with a write protection using the MSCPROT.PROT[15:0] bits

R/W = Read/write bit

R/WP = Read/write bit with a write protection using the MSCPROT.PROT[15:0] bits

Control bit read/write values

This manual describes control bit values in a hexadecimal notation except for one-bit values (and except when decimal or binary notation is required in terms of explanation). The values are described as shown below according to the control bit width.

1 bit: 0 or 1

2 to 4 bits: 0x0 to 0xf

5 to 8 bits: 0x00 to 0xff

9 to 12 bits: 0x000 to 0xfff

13 to 16 bits: 0x0000 to 0xffff

Decimal: 0 to 9999...

Binary: 0b0000... to 0b1111...

Channel number

Multiple channels may be implemented in some peripheral circuits (e.g., 16-bit timer, etc.). The peripheral circuit chapters use ‘n’ as the value that represents the channel number in the register and pin names regardless of the number of channel actually implemented. Normally, the descriptions are applied to all channels. If there is a channel that has different functions from others, the channel number is specified clearly.

Example) T16_nCTL register of the 16-bit timer

If one channel is implemented (Ch.0 only): T16_nCTL = T16_0CTL only

If two channels are implemented (Ch.0 and Ch.1): T16_nCTL = T16_0CTL and T16_1CTL

For the number of channels implemented in the peripheral circuits of this IC, refer to “Features” in the “Overview” chapter.

Table of Contents

Preface	i
Notational conventions and symbols in this manual	i
1 Overview	1-1
1.1 Features	1-1
1.2 Block Diagram	1-5
1.3 Pins	1-6
1.3.1 Pin Configuration Diagram (SQFN7-48pin)	1-6
1.3.2 Pad Configuration Diagram (Chip)	1-7
1.3.3 Pin Descriptions	1-8
2 Power Supply, Reset, and Clocks	2-1
2.1 Power Generator (PWG2)	2-1
2.1.1 Overview	2-1
2.1.2 Pins	2-1
2.1.3 V _{D1} Regulator Operation Mode	2-2
2.1.4 V _{D1} Regulator/RFC Regulator Voltage Modes	2-2
2.2 System Reset Controller (SRC)	2-3
2.2.1 Overview	2-3
2.2.2 Input Pin	2-3
2.2.3 Reset Sources	2-4
2.2.4 Initialization Conditions (Reset Groups)	2-4
2.3 Clock Generator (CLG)	2-5
2.3.1 Overview	2-5
2.3.2 Input/Output Pins	2-6
2.3.3 Clock Sources	2-6
2.3.4 Operations	2-8
2.4 Operating Mode	2-10
2.4.1 Initial Boot Sequence	2-10
2.4.2 Transition between Operating Modes	2-11
2.5 Interrupts	2-12
2.6 Control Registers	2-13
PWG2 V _{D1} Regulator Control Register	2-13
PWG2 RFC Regulator Control Register	2-13
CLG System Clock Control Register	2-14
CLG Oscillation Control Register	2-15
CLG OSC1 Control Register	2-16
CLG OSC3 Control Register	2-16
CLG Interrupt Flag Register	2-17
CLG Interrupt Enable Register	2-17
CLG FOUT Control Register	2-17
CLG Oscillation Frequency Trimming Register 1	2-18
CLG Oscillation Frequency Trimming Register 2	2-19
3 CPU and Debugger	3-1
3.1 Overview	3-1
3.2 CPU Core	3-2
3.2.1 CPU Registers	3-2
3.2.2 Instruction Set	3-2
3.2.3 Reading PSR	3-2
3.2.4 I/O Area Reserved for the S1C17 Core	3-2

3.3	Debugger	3-2
3.3.1	Debugging Functions	3-2
3.3.2	Resource Requirements and Debugging Tools	3-2
3.3.3	List of Debugger Input/Output Pins	3-3
3.3.4	External Connection	3-3
3.3.5	Flash Security Function	3-3
3.4	Control Register	3-4
	MISC PSR Register	3-4
	Debug RAM Base Register	3-4
4	Memory and Bus	4-1
4.1	Overview	4-1
4.2	Bus Access Cycle	4-1
4.3	Flash Memory	4-2
4.3.1	Flash Memory Pin	4-2
4.3.2	Flash Bus Access Cycle Setting	4-2
4.3.3	Flash Programming	4-3
4.4	EEPROM	4-3
4.4.1	EEPROM Pin	4-3
4.4.2	Operations of EEPROM	4-3
4.4.3	Interrupts	4-4
4.5	RAM	4-4
4.6	Display Data RAM	4-5
4.7	Peripheral Circuit Control Registers	4-5
4.7.1	System-Protect Function	4-8
4.8	Control Registers	4-8
	MISC System Protect Register	4-8
	MISC IRAM Size Register	4-8
	FLASHC Flash Read Cycle Register	4-8
	EEPROMC Control Register 0	4-9
	EEPROMC Control Register 1	4-9
	EEPROMC Address Register	4-10
	EEPROMC Write Data Register	4-10
	EEPROMC Interrupt Flag Register	4-10
	EEPROMC Interrupt Enable Register	4-10
5	Interrupt Controller (ITC)	5-1
5.1	Overview	5-1
5.2	Vector Table	5-1
5.2.1	Vector Table Base Address (TTBR)	5-2
5.3	Initialization	5-3
5.4	Maskable Interrupt Control and Operations	5-3
5.4.1	Peripheral Circuit Interrupt Control	5-3
5.4.2	ITC Interrupt Request Processing	5-3
5.4.3	Conditions to Accept Interrupt Requests by the CPU	5-4
5.5	NMI	5-4
5.6	Software Interrupts	5-4
5.7	Interrupt Processing by the CPU	5-4
5.8	Control Registers	5-5
	MISC Vector Table Address Low Register	5-5
	MISC Vector Table Address High Register	5-5

Table of Contents

ITC Interrupt Level Setup Register x	5-5
6 I/O Ports (PPORT)	6-1
6.1 Overview	6-1
6.2 I/O Cell Structure and Functions	6-2
6.2.1 Schmitt Input	6-2
6.2.2 Over Voltage Tolerant Fail-Safe Type I/O Cell	6-2
6.2.3 Pull-Up/Pull-Down	6-2
6.2.4 CMOS Output and High Impedance State	6-3
6.2.5 LED Drive Output Pins	6-3
6.3 Clock Settings	6-3
6.3.1 PPORT Operating Clock	6-3
6.3.2 Clock Supply in SLEEP Mode	6-3
6.3.3 Clock Supply in DEBUG Mode	6-3
6.4 Operations	6-4
6.4.1 Initialization	6-4
6.4.2 Port Input/Output Control	6-5
6.5 Interrupts	6-6
6.6 Control Registers	6-6
Px Port Data Register	6-6
Px Port Enable Register	6-7
Px Port Pull-up/down Control Register	6-7
Px Port Interrupt Flag Register	6-8
Px Port Interrupt Control Register	6-8
Px Port Chattering Filter Enable Register	6-8
Px Port Mode Select Register	6-8
Px Port Function Select Register	6-9
P Port Clock Control Register	6-9
P Port Interrupt Flag Group Register	6-10
6.7 Control Register and Port Function Configuration of this IC	6-11
6.7.1 P0 Port Group	6-11
6.7.2 P1 Port Group	6-12
6.7.3 P2 Port Group	6-13
6.7.4 Pd Port Group	6-14
6.7.5 Common Registers between Port Groups	6-15
7 Universal Port Multiplexer (UPMUX)	7-1
7.1 Overview	7-1
7.2 Peripheral Circuit I/O Function Assignment	7-1
7.3 Control Registers	7-2
Pxy–xz Universal Port Multiplexer Setting Register	7-2
8 Watchdog Timer (WDT2)	8-1
8.1 Overview	8-1
8.2 Clock Settings	8-1
8.2.1 WDT2 Operating Clock	8-1
8.2.2 Clock Supply in DEBUG Mode	8-1
8.3 Operations	8-2
8.3.1 WDT2 Control	8-2
8.3.2 Operations in HALT and SLEEP Modes	8-3
8.4 Control Registers	8-3
WDT2 Clock Control Register	8-3

WDT2 Control Register	8-4
WDT2 Counter Compare Match Register	8-4
9 Supply Voltage Detector (SVD5)	9-1
9.1 Overview	9-1
9.2 Input Pins and External Connection	9-2
9.2.1 Input Pins	9-2
9.2.2 External Connection	9-2
9.3 Clock Settings	9-2
9.3.1 SVD5 Operating Clock	9-2
9.3.2 Clock Supply in SLEEP Mode	9-2
9.3.3 Clock Supply in DEBUG Mode	9-3
9.4 Operations	9-3
9.4.1 SVD5 Control	9-3
9.4.2 SVD5 Operations	9-4
9.5 SVD5 Interrupt and Reset	9-4
9.5.1 SVD5 Interrupt	9-4
9.5.2 SVD5 Reset	9-5
9.6 Control Registers	9-5
SVD5 Clock Control Register	9-5
SVD5 Control Register	9-6
SVD5 Status and Interrupt Flag Register	9-7
SVD5 Interrupt Enable Register	9-8
10 16-bit Timers (T16)	10-1
10.1 Overview	10-1
10.2 Input Pin	10-1
10.3 Clock Settings	10-2
10.3.1 T16 Operating Clock	10-2
10.3.2 Clock Supply in SLEEP Mode	10-2
10.3.3 Clock Supply in DEBUG Mode	10-2
10.3.4 Event Counter Clock	10-2
10.4 Operations	10-2
10.4.1 Initialization	10-2
10.4.2 Counter Underflow	10-3
10.4.3 Operations in Repeat Mode	10-3
10.4.4 Operations in One-shot Mode	10-3
10.4.5 Counter Value Read	10-4
10.5 Interrupt	10-4
10.6 Control Registers	10-4
T16 Ch.n Clock Control Register	10-4
T16 Ch.n Mode Register	10-5
T16 Ch.n Control Register	10-5
T16 Ch.n Reload Data Register	10-6
T16 Ch.n Counter Data Register	10-6
T16 Ch.n Interrupt Flag Register	10-6
T16 Ch.n Interrupt Enable Register	10-6
11 UART (UART3)	11-1
11.1 Overview	11-1
11.2 Input/Output Pins and External Connections	11-2
11.2.1 List of Input/Output Pins	11-2

Table of Contents

11.2.2	External Connections.....	11-2
11.2.3	Input Pin Pull-Up Function.....	11-2
11.2.4	Output Pin Open-Drain Output Function.....	11-2
11.2.5	Input/Output Signal Inverting Function.....	11-2
11.3	Clock Settings.....	11-2
11.3.1	UART3 Operating Clock.....	11-2
11.3.2	Clock Supply in SLEEP Mode.....	11-3
11.3.3	Clock Supply in DEBUG Mode.....	11-3
11.3.4	Baud Rate Generator.....	11-3
11.4	Data Format.....	11-3
11.5	Operations.....	11-4
11.5.1	Initialization.....	11-4
11.5.2	Data Transmission.....	11-4
11.5.3	Data Reception.....	11-6
11.5.4	IrDA Interface.....	11-7
11.5.5	Carrier Modulation.....	11-7
11.6	Receive Errors.....	11-8
11.6.1	Framing Error.....	11-8
11.6.2	Parity Error.....	11-8
11.6.3	Overrun Error.....	11-8
11.7	Interrupts.....	11-9
11.8	Control Registers.....	11-9
	UART3 Ch. <i>n</i> Clock Control Register.....	11-9
	UART3 Ch. <i>n</i> Mode Register.....	11-10
	UART3 Ch. <i>n</i> Baud-Rate Register.....	11-11
	UART3 Ch. <i>n</i> Control Register.....	11-11
	UART3 Ch. <i>n</i> Transmit Data Register.....	11-12
	UART3 Ch. <i>n</i> Receive Data Register.....	11-12
	UART3 Ch. <i>n</i> Status and Interrupt Flag Register.....	11-12
	UART3 Ch. <i>n</i> Interrupt Enable Register.....	11-13
	UART3 Ch. <i>n</i> Carrier Waveform Register.....	11-14
12	Synchronous Serial Interface (SPIA).....	12-1
12.1	Overview.....	12-1
12.2	Input/Output Pins and External Connections.....	12-2
12.2.1	List of Input/Output Pins.....	12-2
12.2.2	External Connections.....	12-2
12.2.3	Pin Functions in Master Mode and Slave Mode.....	12-3
12.2.4	Input Pin Pull-Up/Pull-Down Function.....	12-3
12.3	Clock Settings.....	12-3
12.3.1	SPIA Operating Clock.....	12-3
12.3.2	Clock Supply in DEBUG Mode.....	12-4
12.3.3	SPI Clock (SPICLK _{<i>n</i>}) Phase and Polarity.....	12-4
12.4	Data Format.....	12-4
12.5	Operations.....	12-5
12.5.1	Initialization.....	12-5
12.5.2	Data Transmission in Master Mode.....	12-5
12.5.3	Data Reception in Master Mode.....	12-7
12.5.4	Terminating Data Transfer in Master Mode.....	12-8
12.5.5	Data Transfer in Slave Mode.....	12-8
12.5.6	Terminating Data Transfer in Slave Mode.....	12-10
12.6	Interrupts.....	12-10

12.7	Control Registers	12-11
	SPIA Ch. <i>n</i> Mode Register.....	12-11
	SPIA Ch. <i>n</i> Control Register.....	12-12
	SPIA Ch. <i>n</i> Transmit Data Register	12-13
	SPIA Ch. <i>n</i> Receive Data Register	12-13
	SPIA Ch. <i>n</i> Interrupt Flag Register	12-13
	SPIA Ch. <i>n</i> Interrupt Enable Register	12-14
13	I²C (I2C)	13-1
13.1	Overview	13-1
13.2	Input/Output Pins and External Connections	13-2
	13.2.1 List of Input/Output Pins	13-2
	13.2.2 External Connections.....	13-2
13.3	Clock Settings	13-3
	13.3.1 I2C Operating Clock	13-3
	13.3.2 Clock Supply in DEBUG Mode	13-3
	13.3.3 Baud Rate Generator.....	13-3
13.4	Operations.....	13-4
	13.4.1 Initialization	13-4
	13.4.2 Data Transmission in Master Mode	13-5
	13.4.3 Data Reception in Master Mode	13-7
	13.4.4 10-bit Addressing in Master Mode	13-9
	13.4.5 Data Transmission in Slave Mode	13-10
	13.4.6 Data Reception in Slave Mode	13-12
	13.4.7 Slave Operations in 10-bit Address Mode	13-14
	13.4.8 Automatic Bus Clearing Operation	13-14
	13.4.9 Error Detection.....	13-15
13.5	Interrupts	13-16
13.6	Control Registers	13-17
	I2C Ch. <i>n</i> Clock Control Register	13-17
	I2C Ch. <i>n</i> Mode Register.....	13-18
	I2C Ch. <i>n</i> Baud-Rate Register.....	13-18
	I2C Ch. <i>n</i> Own Address Register.....	13-18
	I2C Ch. <i>n</i> Control Register	13-18
	I2C Ch. <i>n</i> Transmit Data Register.....	13-19
	I2C Ch. <i>n</i> Receive Data Register	13-20
	I2C Ch. <i>n</i> Status and Interrupt Flag Register.....	13-20
	I2C Ch. <i>n</i> Interrupt Enable Register	13-21
14	16-bit PWM Timers (T16B).....	14-1
14.1	Overview	14-1
14.2	Input/Output Pins	14-2
14.3	Clock Settings	14-2
	14.3.1 T16B Operating Clock	14-2
	14.3.2 Clock Supply in SLEEP Mode	14-3
	14.3.3 Clock Supply in DEBUG Mode	14-3
	14.3.4 Event Counter Clock.....	14-3
14.4	Operations.....	14-3
	14.4.1 Initialization	14-3
	14.4.2 Counter Block Operations	14-4
	14.4.3 Comparator/Capture Block Operations	14-7
	14.4.4 TOUT Output Control.....	14-15
14.5	Interrupt.....	14-20

Table of Contents

14.6	Control Registers	14-21
	T16B Ch. <i>n</i> Clock Control Register	14-21
	T16B Ch. <i>n</i> Counter Control Register.....	14-22
	T16B Ch. <i>n</i> Max Counter Data Register	14-23
	T16B Ch. <i>n</i> Timer Counter Data Register.....	14-23
	T16B Ch. <i>n</i> Counter Status Register	14-23
	T16B Ch. <i>n</i> Interrupt Flag Register.....	14-24
	T16B Ch. <i>n</i> Interrupt Enable Register	14-25
	T16B Ch. <i>n</i> Comparator/Capture <i>m</i> Control Register	14-26
	T16B Ch. <i>n</i> Compare/Capture <i>m</i> Data Register	14-29
15	Sound Generator (SNDA2)	15-1
15.1	Overview	15-1
15.2	Output Pins and External Connections	15-2
	15.2.1 List of Output Pins	15-2
	15.2.2 Output Pin Drive Mode	15-2
	15.2.3 External Connections.....	15-2
15.3	Clock Settings	15-3
	15.3.1 SNDA2 Operating Clock	15-3
	15.3.2 Clock Supply in SLEEP Mode	15-3
	15.3.3 Clock Supply in DEBUG Mode	15-3
15.4	Operations.....	15-3
	15.4.1 Initialization	15-3
	15.4.2 Buzzer Output in Normal Buzzer Mode	15-3
	15.4.3 Buzzer Output in One-shot Buzzer Mode.....	15-6
	15.4.4 Output in Melody Mode	15-7
15.5	Interrupts	15-9
15.6	Control Registers	15-9
	SNDA2 Clock Control Register.....	15-9
	SNDA2 Select Register.....	15-10
	SNDA2 Control Register.....	15-11
	SNDA2 Data Register	15-11
	SNDA2 Interrupt Flag Register.....	15-12
	SNDA2 Interrupt Enable Register.....	15-12
16	LCD Driver (LCD4B)	16-1
16.1	Overview	16-1
16.2	Output Pins and External Connections	16-2
	16.2.1 List of Output Pins	16-2
	16.2.2 External Connections.....	16-2
16.3	Clock Settings	16-2
	16.3.1 LCD4B Operating Clock	16-2
	16.3.2 Clock Supply in SLEEP Mode	16-3
	16.3.3 Clock Supply in DEBUG Mode	16-3
	16.3.4 Frame Frequency	16-3
16.4	LCD Power Supply.....	16-3
	16.4.1 Internal Generation Mode.....	16-4
	16.4.2 External Voltage Application Mode 1	16-4
	16.4.3 External Voltage Application Mode 2	16-4
	16.4.4 External Voltage Application Mode 3	16-5
	16.4.5 LCD Power Supply Circuit Settings	16-5
16.5	Operations.....	16-5
	16.5.1 Initialization	16-5

16.5.2	Display On/Off	16-6
16.5.3	Inverted Display	16-7
16.5.4	Drive Duty Switching.....	16-7
16.5.5	Drive Waveforms	16-7
16.5.6	Partial Common Output Drive.....	16-10
16.5.7	n-Segment-Line Inverse AC Drive	16-10
16.6	Display Data RAM.....	16-11
16.6.1	Display Area Selection.....	16-11
16.6.2	Segment Pin Assignment	16-11
16.6.3	Common Pin Assignment	16-11
16.7	Interrupt.....	16-14
16.8	Control Registers	16-14
	LCD4B Clock Control Register	16-14
	LCD4B Control Register.....	16-15
	LCD4B Timing Control Register 1	16-15
	LCD4B Timing Control Register 2	16-16
	LCD4B Power Control Register.....	16-16
	LCD4B Display Control Register	16-17
	LCD4B COM Pin Control Register 0	16-18
	LCD4B Interrupt Flag Register	16-18
	LCD4B Interrupt Enable Register	16-18
17	R/F Converter (RFC2)	17-1
17.1	Overview	17-1
17.2	Input/Output Pins and External Connections.....	17-2
17.2.1	List of Input/Output Pins	17-2
17.2.2	External Connections.....	17-2
17.3	Clock Settings	17-3
17.3.1	RFC2 Operating Clock.....	17-3
17.3.2	Clock Supply in SLEEP Mode	17-3
17.3.3	Clock Supply in DEBUG Mode	17-3
17.4	Operations.....	17-3
17.4.1	Initialization	17-3
17.4.2	Operating Modes	17-4
17.4.3	RFC2 Counters.....	17-4
17.4.4	Converting Operations and Control Procedure	17-5
17.4.5	CR Oscillation Frequency Monitoring Function	17-7
17.5	Interrupts	17-7
17.6	Control Registers	17-7
	RFC2 Ch.n Clock Control Register.....	17-7
	RFC2 Ch.n Control Register.....	17-8
	RFC2 Ch.n Oscillation Trigger Register.....	17-9
	RFC2 Ch.n Measurement Counter Low and High Registers	17-10
	RFC2 Ch.n Time Base Counter Low and High Registers	17-10
	RFC2 Ch.n Interrupt Flag Register	17-10
	RFC2 Ch.n Interrupt Enable Register.....	17-11
18	Charge Pump Regulator (CHREG).....	18-1
18.1	Overview	18-1
18.2	Input/Output Pins and External Connections.....	18-1
18.2.1	List of Input/Output Pins	18-1
18.2.2	External Connections.....	18-1
18.3	Operations.....	18-2

Table of Contents

18.3.1	Charge Pump ON/OFF Procedure	18-2
18.3.2	Charge Pump Settings.....	18-2
18.4	Control Registers	18-2
	CHREG Configuration Register.....	18-2
	CHREG Control Register.....	18-2
19	Multiplier/Divider (COPRO2)	19-1
19.1	Overview	19-1
19.2	Operation Mode and Output Mode	19-1
19.3	Multiplication	19-2
19.4	Division	19-3
19.5	MAC	19-5
19.6	Reading Operation Results	19-7
20	Electrical Characteristics	20-1
20.1	Absolute Maximum Ratings	20-1
20.2	Recommended Operating Conditions	20-1
20.3	Current Consumption	20-2
20.4	System Reset Controller (SRC) Characteristics	20-2
20.5	Clock Generator (CLG) Characteristics	20-3
20.6	Flash Memory Characteristics	20-5
20.7	EEPROM Characteristics.....	20-5
20.8	Input/Output Port (PPORT) Characteristics.....	20-6
20.9	Supply Voltage Detector (SVD5) Characteristics.....	20-7
20.10	UART (UART3) Characteristics	20-9
20.11	Synchronous Serial Interface (SPIA) Characteristics	20-9
20.12	I ² C (I2C) Characteristics	20-10
20.13	LCD Driver (LCD4B) Characteristics	20-11
20.14	R/F Converter (RFC2) Characteristics.....	20-14
20.15	Charge Pump Regulator (CHREG) Characteristics.....	20-16
21	Basic External Connection Diagram	21-1
22	Package	22-1
Appendix A	List of Peripheral Circuit Control Registers	AP-A-1
0x4000–0x4008	Misc Registers (MISC)	AP-A-1
0x4020–0x4022	Power Generator (PWG2).....	AP-A-1
0x4040–0x4054	Clock Generator (CLG)	AP-A-1
0x4080–0x4090	Interrupt Controller (ITC).....	AP-A-2
0x40a0–0x40a4	Watchdog Timer (WDT2).....	AP-A-3
0x4100–0x4106	Supply Voltage Detector (SVD5).....	AP-A-3
0x4160–0x416c	16-bit Timer (T16) Ch.0	AP-A-4
0x41b0	Flash Controller (FLASHC)	AP-A-4
0x41c0–0x41ca	EEPROM Controller (EEPROMC)	AP-A-4
0x4200–0x42e2	I/O Ports (PPORT)	AP-A-5
0x4300–0x430e	Universal Port Multiplexer (UPMUX).....	AP-A-7
0x4380–0x4390	UART (UART3) Ch.0	AP-A-8

0x43a0–0x43ac	16-bit Timer (T16) Ch.1	AP-A-9
0x43b0–0x43ba	Synchronous Serial Interface (SPIA) Ch.0	AP-A-9
0x43c0–0x43d2	I ² C (I2C) Ch.0	AP-A-10
0x5000–0x501a	16-bit PWM Timer (T16B) Ch.0	AP-A-11
0x5040–0x505a	16-bit PWM Timer (T16B) Ch.1	AP-A-12
0x5260–0x526c	16-bit Timer (T16) Ch.2	AP-A-13
0x5300–0x530a	Sound Generator (SNDA2)	AP-A-14
0x5340–0x5342	Charge Pump Regulator (CHREG)	AP-A-14
0x5400–0x5412	LCD Driver (LCD4B).....	AP-A-14
0x5440–0x5450	R/F Converter (RFC2) Ch.0	AP-A-15
0xffff90	Debugger (DBG)	AP-A-16
Appendix B	Power Saving	AP-B-1
B.1	Operating Status Configuration Examples for Power Saving	AP-B-1
B.2	Other Power Saving Methods	AP-B-1
Appendix C	Mounting Precautions	AP-C-1
Appendix D	Measures Against Noise.....	AP-D-1
Appendix E	Initialization Routine.....	AP-E-1
Revision History		

1 Overview

The S1C17W11 is a 16-bit MCU featuring low power consumption. It incorporates various serial interfaces, an LCD driver, and a high-precision CR oscillation-type A/D converter (R/F converter). Combined with the powerful processing capability of a 16-bit CPU, it is best suited for battery-powered applications requiring LCD and resistance measurement functions (such as thermometers, water quality meters, and salinity meters). In addition, it includes a charge pump regulator for driving a buzzer and LEDs, allowing for white LED illumination and increased buzzer volume.

1.1 Features

Table 1.1.1 Features

Model		S1C17W11
CPU		
CPU core	Seiko Epson original 16-bit RISC CPU core S1C17	
Other	On-chip debugger	
Embedded Flash memory		
Capacity (for both instructions and data)	48K bytes	
Erase/program count	1,000 times (min.) * Programming by the debugging tool ICDmini	
Other	Security function to protect from reading/programming by ICDmini	
	On-board programming function using ICDmini	
	* An external smoothing capacitor is required.	
	Flash programming voltage can be generated internally.	
Embedded EEPROM		
Capacity	128 bytes	
Erase/program count	100,000 times (min.)	
Embedded RAM		
Capacity	2K bytes	
Embedded display RAM		
Capacity	20 bytes (4 COM x 20 SEG x 2 screens)	
Clock generator (CLG)		
System clock source	4 sources (IOSC/OSC1/OSC3/EXOSC)	
System clock frequency (operating frequency)	4.2 MHz (max.)	
IOSC oscillator circuit (boot clock source)	700 kHz (typ.) embedded oscillator	
	23 μ s (max.) starting time (time from cancelation of SLEEP state to vector table read by the CPU)	
OSC1 oscillator circuit	32 kHz (typ.) embedded oscillator	
OSC3 oscillator circuit	4 MHz (max.)/2 MHz/1 MHz/500 kHz/384 kHz/250 kHz embedded oscillator	
EXOSC clock input	4.2 MHz (max.) square or sine wave input	
Other	Configurable system clock division ratio	
	Configurable system clock used at wake up from SLEEP state	
	Operating clock frequency for the CPU and all peripheral circuits is selectable.	
I/O port (PPORT)		
Number of general-purpose ports	Input/output port	24 bits (max.), 2 bits can also be used as debug ports (DSIO / DST2).
	Output port	1 bit (max.), Can also be used as a debug port (DCLK).
	Other	Pins are shared with the peripheral I/O.
Number of input interrupt ports	22 bits (max.)	
Number of ports that support universal port	16 bits	

1 Overview

Model	S1C17W11
multiplexer (UPMUX)	A peripheral circuit I/O function selected via software can be assigned to each port. Applicable functions: I2C, SPIA, UART3, T16B
LED drive output ports	2 bits, Nch open-drain, maximum output current 10 mA

Model	S1C17W11
Timers	
Watchdog timer (WDT2)	Generates NMI or hardware reset. Programmable NMI/reset generation cycle
16-bit timer (T16)	3 channels Generates the SPIA master clock.
16-bit PWM timer (T16B)	2 channels Event counter/capture function PWM waveform generation function Number of PWM output or capture input ports: 2 ports (per channel)
Supply voltage detector (SVD5)	
Detection voltage	V _{DD} or external voltage (One external voltage input port is provided, and an external voltage level can be detected even if it exceeds V _{DD} .)
Detection level	32 levels (1.15 to 3.2 V)
Other	Intermittent operation mode Generates an interrupt or reset according to the detection level evaluation.
Serial interfaces	
UART (UART3)	1 channel Baud-rate generator included, IrDA1.0 supported Open drain output, signal polarity, and baud rate division ratio are configurable. Infrared communication carrier modulation output function
Synchronous serial interface (SPIA)	1 channel 2 to 16-bit variable data length The 16-bit timer (T16) can be used for the baud-rate generator in master mode.
I ² C (I2C) ^{*1}	1 channel Baud-rate generator included
Sound generator (SNDA2)	
Buzzer output function	512 Hz to 16 kHz output frequencies One-shot output function
Melody generation function	Pitch: 128 Hz to 16 kHz ≈ C3 to C6 Duration: 7 notes/rests (Half note/rest to thirty-second note/rest) Tempo: 16 tempos (30 to 480) Tie may be specified.
Charge pump regulator (CHREG)	
Output voltage	V _{DD} × 2 ≤ 5.0 V (typ.)
Other	Used for the I/O voltage of the SNDA2 output (BZOUT, #BZOUT) Can be used as the LED drive power supply, 10 mA (max.) output current
LCD driver (LCD4B)	
LCD output (max value.)	20SEG × 1 to 4COM (max.)
LCD power supply	1/3 bias power supply included (internal step-up/resistive divider selectable). An external voltage can be applied. (Internal resistors are provided to divide the external source voltage.)
LCD contrast	29 levels (Only when the internal voltage booster is enabled.)
R/F converter (RFC2)	
Conversion method	CR oscillation type with 24-bit counters
Number of conversion channels	1 channel
Supported sensors	DC-bias resistive sensors AC-bias resistive sensors
Number of connectable sensors	2 (max.) in DC resistive sensor mode 1 (max.) in AC resistive sensor mode
Reference power supply for measurement	Internally generated by dedicated power circuit

1 Overview

Model	S1C17W11
Multiplier/divider (COPRO2)	
Arithmetic functions	16-bit x 16-bit multiplier
	16-bit x 16-bit + 32-bit multiply and accumulation unit
	32-bit ÷ 32-bit divider
Reset	
#RESET pin	Reset when the reset pin is set to low.
Power-on reset	Reset at power on.
Key entry reset	Reset when the P10 to P11/P12/P13 keys are pressed simultaneously (can be enabled/disabled using a register).
Watchdog timer reset	Reset when the watchdog timer overflows (can be enabled/disabled using a register).
Supply voltage detector reset	Reset when the supply voltage detector detects the set voltage level (can be enabled/disabled using a register).
Interrupt	
Non-maskable interrupt	4 systems (Reset, address misaligned interrupt, debug, NMI)
Programmable interrupt	External interrupt: 1 system (8 levels)
	Internal interrupt: 14 systems (8 levels)
Power supply voltage	
V _{DD} operating voltage	1.2 to 3.6 V
Internal logic voltage	Selectable among 1.2 V/1.4 V/1.8 V
V _{DD} operating voltage for Flash programming	2.2 to 3.6 V (Programming voltage V _{PP} : 7.5 V supplied externally or generated internally)
V _{DD} operating voltage for EEPROM programming	2.2 to 3.6 V (Programming voltage V _{PP} : generated internally)
Operating temperature	
Operating temperature range	-40 to 85°C
Current consumption (typ. value)	
SLEEP mode	109 nA (V _{DD} =1.5V), 116 nA (V _{DD} =3.0V) IOSC = OFF, OSC1 = OFF, OSC3 = OFF, V _{D1} =1.2V
HALT mode	1.28 µA OSC1 = 32 kHz, V _{D1} =1.2V
RUN mode	3.82 µA OSC1 = 32 kHz, CPU = OSC1, V _{D1} =1.2V
	75.6 µA IOSC = 700kHz, CPU = IOSC, V _{D1} =1.2V
	156 µA OSC3 = 1MHz, CPU = OSC3, V _{D1} =1.2V
Shipping form	
1	SQFN7-48pin (P-VQFN048-0707-0.50, 7 × 7 mm, t = 1.0 mm, 0.5 mm pitch) *2
2	Bare chip

*1 The input filter in I2C (SDA and SCL inputs) does not comply with the standard for removing noise spikes less than 50 ns.

*2 Shown in parentheses is the JEITA package name.

1.2 Block Diagram

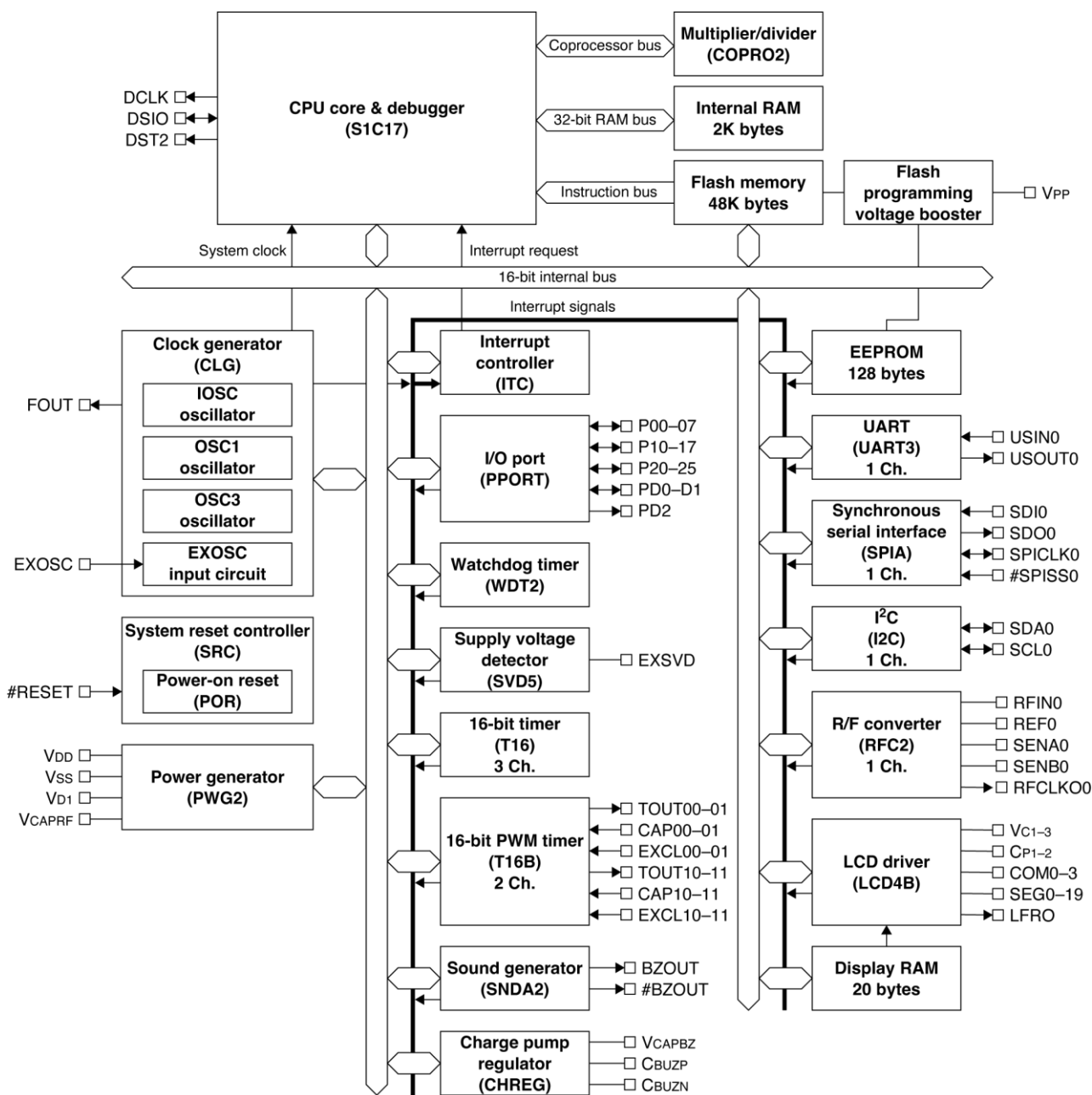


Figure 1.2.1 S1C17W11 Block Diagram

1.3.2 Pad Configuration Diagram (Chip)

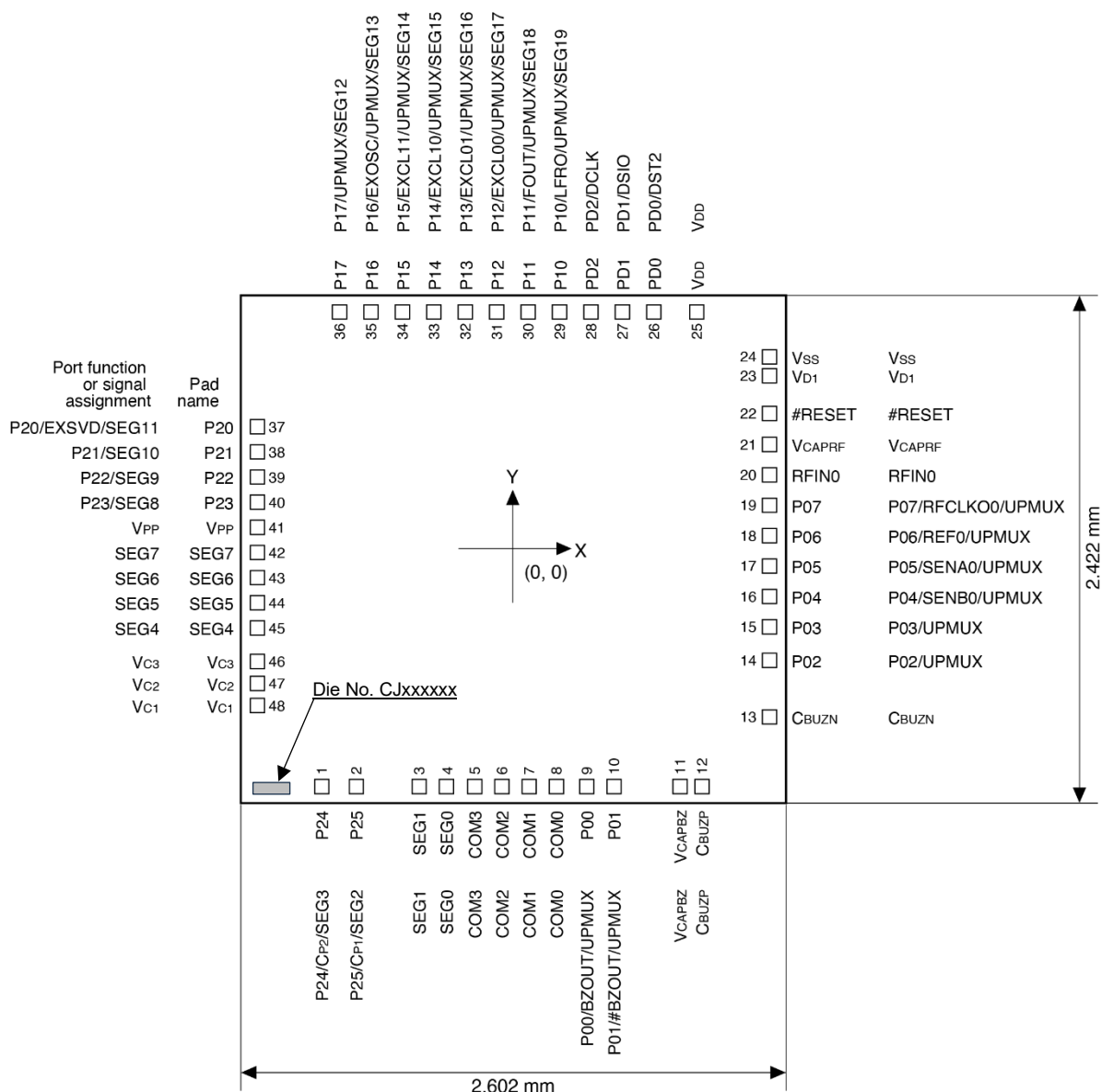


Figure 1.3.2.1 S1C17W11 Pin Configuration Diagram (Chip)

Pad opening: X = 68 μm, Y = 68 μm

Chip thickness: 400 μm

Table 1.3.2.1 S1C17W11 Pad Coordinates

No.	X _{μm}	Y _{μm}	No.	X _{μm}	Y _{μm}	No.	X _{μm}	Y _{μm}	No.	X _{μm}	Y _{μm}
1	-915.0	-111.5	13	1221.5	-800.0	25	875.0	1131.5	37	-1221.5	585.0
2	-750.0	-1131.5	14	1221.5	-530.0	26	670.0	1131.5	38	-1221.5	465.0
3	-450.0	-1131.5	15	1221.5	-370.0	27	520.0	1131.5	39	-1221.5	345.0
4	-320.0	-1131.5	16	1221.5	-225.0	28	370.0	1131.5	40	-1221.5	225.0
5	-185.0	-1131.5	17	1221.5	-80.0	29	220.0	1131.5	41	-1221.5	105.0
6	-55.0	-1131.5	18	1221.5	65.0	30	70.0	1131.5	42	-1221.5	-15.0
7	75.0	-1131.5	19	1221.5	210.0	31	-80.0	1131.5	43	-1221.5	-135.0
8	205.0	-1131.5	20	1221.5	355.0	32	-230.0	1131.5	44	-1221.5	-255.0
9	350.0	-1131.5	21	1221.5	500.0	33	-380.0	1131.5	45	-1221.5	-375.0
10	480.0	-1131.5	22	1221.5	647.5	34	-530.0	1131.5	46	-1221.5	-534.5
11	795.0	-1131.5	23	1221.5	827.5	35	-680.0	1131.5	47	-1221.5	-639.5
12	900.0	-1131.5	24	1221.5	917.5	36	-830.0	1131.5	48	-1221.5	-744.5

1 Overview

1.3.3 Pin Descriptions

Symbol meanings

Assigned signal:

The signal listed at the top of each pin is assigned in the initial state. The pin function must be switched via software to assign another signal (see the “I/O Ports” chapter).

I/O:

I = Input
 O = Output
 I/O = Input/output
 P = Power supply
 A = Analog signal
 Hi-Z = High impedance state

Initial state:

I (Pull-up) = Input with pulled up
 I (Pull-down) = Input with pulled down
 Hi-Z = High impedance state
 O (H) = High level output
 O (L) = Low level output

Tolerant fail-safe structure:

✓ = Over voltage tolerant fail-safe type I/O cell included (see the “I/O Ports” chapter)

Table 1.3.2 Pin description

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function
V _{DD}	V _{DD}	P	-	-	Power supply (+)
V _{SS}	V _{SS}	P	-	-	GND
V _{PP}	V _{PP}	P	-	-	Flash programming power supply
V _{D1}	V _{D1}	A	-	-	DC-DC converter output
V _{C1-3}	V _{C1-3}	P	-	-	LCD panel drive power supply
#RESET	#RESET	I	I (Pull-up)	-	Reset input
V _{CAPBZ}	V _{CAPBZ}	P	Hi-Z	-	Buzzer/LED drive voltage output
C _{BUZP}	C _{BUZP}	A	Hi-Z	-	Buzzer/LED drive power supply boost capacitor connection pin
C _{BUZN}	C _{BUZN}	A	Hi-Z	-	Buzzer/LED drive power supply boost capacitor connection pin
V _{CAPRF}	V _{CAPRF}	A	-	-	R/F converter regulator output
RFIN0	RFIN0	A	Hi-Z	-	R/F converter Ch.0 oscillation input
P00	P00	I/O	Hi-Z	✓	I/O port
	BZOUT	O			Sound generator buzzer output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P01	P01	I/O	Hi-Z	✓	I/O port
	#BZOUT	O			Sound generator inverted buzzer output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P02	P02	I/O	Hi-Z	✓	I/O port (open-drain type)
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P03	P03	I/O	Hi-Z	✓	I/O port (open-drain type)
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P04	P04	I/O	Hi-Z	✓	I/O port
	SEN0	A			R/F converter Ch.0 sensor B oscillation pin
	UPMUX	I/O			User-selected I/O (universal port multiplexer)

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function
P05	P05	I/O	Hi-Z	✓	I/O port
	SENA0	A			R/F converter Ch.0 sensor A oscillation pin
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P06	P06	I/O	Hi-Z	✓	I/O port
	REF0	A			R/F converter Ch.0 reference oscillation pin
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P07	P07	I/O	Hi-Z	✓	I/O port
	RFCLK00	A			R/F converter Ch.0 clock monitor output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P10	P10	I/O	Hi-Z	✓	I/O port
	LFRO	O			LCD frame signal monitor output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG19	A			LCD segment output 19
P11	P11	I/O	Hi-Z	✓	I/O port
	FOUT	O			Clock generator clock external output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG18	A			LCD segment output 18
P12	P12	I/O	Hi-Z	✓	I/O port
	EXCL00	I			16-bit PWM timer Ch.0 event counter input 0
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG17	A			LCD segment output 17
P13	P13	I/O	Hi-Z	✓	I/O port
	EXCL01	I			16-bit PWM timer Ch.0 event counter input 1
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG16	A			LCD segment output 16
P14	P14	I/O	Hi-Z	✓	I/O port
	EXCL10	I			16-bit PWM timer Ch.1 event counter input 0
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG15	A			LCD segment output 15
P15	P15	I/O	Hi-Z	✓	I/O port
	EXCL11	I			16-bit PWM timer Ch.1 event counter input 1
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG14	A			LCD segment output 14
P16	P16	I/O	Hi-Z	✓	I/O port
	EXOSC	I			Clock generator external clock input
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG13	A			LCD segment output 13
P17	P17	I/O	Hi-Z	✓	I/O port
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG12	A			LCD segment output 12
P20	P20	I/O	Hi-Z	✓	I/O port
	EXSVD	A			External power supply voltage detection input
	SEG11	A			LCD segment output 11
P21	P21	I/O	Hi-Z	✓	I/O port
	SEG10	A			LCD segment output 10
P22	P22	I/O	Hi-Z	✓	I/O port
	SEG9	A			LCD segment output 9
P23	P23	I/O	Hi-Z	✓	I/O port
	SEG8	A			LCD segment output 8

1 Overview

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function
P24	P24	O	Hi-Z	✓	I/O port
	C _{P2}	A			LCD drive voltage boost capacitor connection pin
	SEG3	A			LCD segment output 3
P25	P25	I/O	Hi-Z	✓	I/O port
	C _{P1}	A			LCD drive voltage boost capacitor connection pin
	SEG2	A			LCD segment output 2
PD0	DST2	O	O (L)	✓	On-chip debugger status output
	PD0	I/O			I/O port
PD1	DSIO	I/O	I (Pull-up)	✓	On-chip debugger data input/output
	PD1	I/O			I/O port
PD2	DCLK	O	O (H)	-	On-chip debugger clock output
	PD2	O			Output port
COM0-3	COM0-3	A	Hi-Z	-	LCD common outputs
SEG0-1	SEG0-1	A	Hi-Z	-	LCD segment outputs
SEG4-7	SEG4-7				

Note: In the peripheral circuit descriptions, the assigned signal name is used as the pin name.

Universal port multiplexer (UPMUX)

The universal port multiplexer (UPMUX) allows software to select the peripheral circuit input/output function to be assigned to each pin from those listed below.

Table 1.3.3 Peripheral Circuit Input/output Function Selectable by UPMUX

Peripheral circuit	Signal to be assigned	I/O	Channel number <i>n</i>	Function
Synchronous serial interface (SPIA)	SDI _{<i>n</i>}	I	<i>n</i> = 0	SPIA Ch. <i>n</i> data input
	SDO _{<i>n</i>}	O		SPIA Ch. <i>n</i> data output
	SPICLK _{<i>n</i>}	I/O		SPIA Ch. <i>n</i> clock input/output
	#SPISS _{<i>n</i>}	I		SPIA Ch. <i>n</i> slave-select input
I ² C (I2C)	SCL _{<i>n</i>}	I/O	<i>n</i> = 0	I2C Ch. <i>n</i> clock input/output
	SDA _{<i>n</i>}	I/O		I2C Ch. <i>n</i> data input/output
UART (UART3)	USIN _{<i>n</i>}	I	<i>n</i> = 0	UART3 Ch. <i>n</i> data input
	USOUT _{<i>n</i>}	O		UART3 Ch. <i>n</i> data output
16-bit PWM timer (T16B)	TOUT _{<i>n</i>0} /CAP _{<i>n</i>0}	I/O	<i>n</i> = 0, 1	T16B Ch. <i>n</i> PWM output/capture input 0
	TOUT _{<i>n</i>1} /CAP _{<i>n</i>1}	I/O		T16B Ch. <i>n</i> PWM output/capture input 1

Note: Do not assign a function to two or more pins simultaneously.

2 Power Supply, Reset, and Clocks

The power supply, reset, and clocks in this IC are managed by the embedded power generator, system reset controller, and clock generator, respectively.

2.1 Power Generator (PWG2)

2.1.1 Overview

PWG2 is the power generator that controls the internal power supply system to drive this IC with stability and low power. The main features of PWG2 are outlined below.

- Embedded V_{D1} regulator
 - The V_{D1} regulator generates the V_{D1} voltage to drive internal circuits (excluding RFC2), this makes it possible to keep current consumption constant independent of the V_{DD} voltage level.
 - The V_{D1} regulator supports two operation modes, normal mode and economy mode, and setting the V_{D1} regulator into economy mode at light loads helps achieve low-power operations.
 - Three selectable voltage levels (1.2 V/1.4 V/1.8 V) are available. The output voltages of the V_{D1} and RFC regulators are always identical.
- Embedded RFC regulator
 - The RFC regulator generates the voltage V_{CAPRF} that operates RFC2, improving the R/F converter measurement characteristics when the V_{DD} voltage is low.

Figure 2.1.1.1 shows the PWG2 configuration.

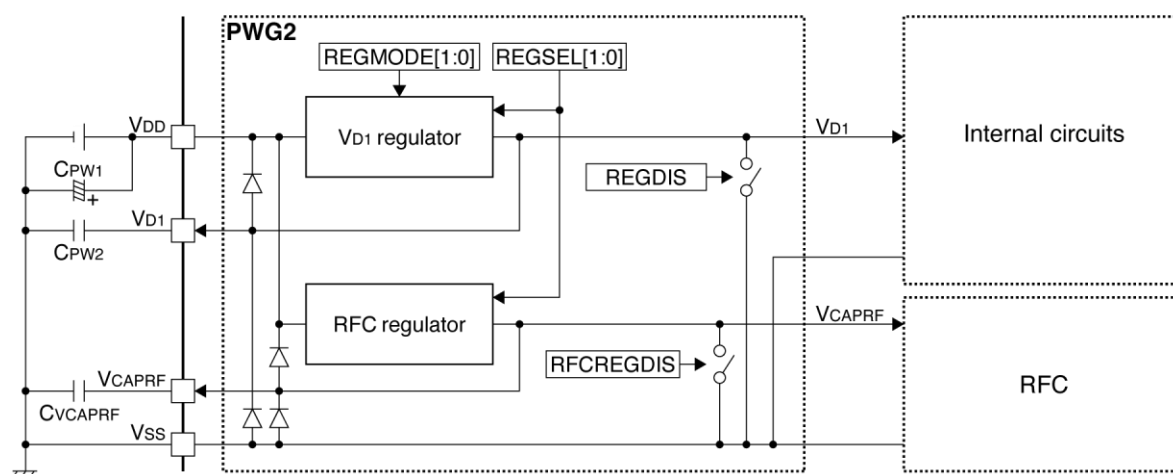


Figure 2.1.1.1 PWG2 Configuration

2.1.2 Pins

Table 2.1.2.1 lists the PWG2 pins.

Table 2.1.2.1 List of PWG2 Pins

Pin name	I/O	Initial status	Function
V_{DD}	P	–	Power supply (+)
V_{SS}	P	–	GND
V_{D1}	A	–	V_{D1} regulator output pin
V_{CAPRF}	A	–	RFC regulator output pin

For the V_{DD} operating voltage range and recommended external parts, refer to “Recommended Operating Conditions, Power supply voltage V_{DD} ” in the “Electrical Characteristics” chapter and the “Basic External Connection Diagram” chapter, respectively.

2 Power Supply, Reset, and Clocks

2.1.3 V_{D1} Regulator Operation Mode

The V_{D1} regulator supports two operation modes: normal mode and economy mode. Setting the V_{D1} regulator into economy mode at light loads helps achieve low-power operations. Table 2.1.3.1 lists examples of light load conditions in which economy mode can be set.

Table 2.1.3.1 Examples of Light Load Conditions in which Economy Mode Can be Set

Light load condition	Exceptions
SLEEP mode (when all oscillators are stopped, or OSC1 only is active)	When a clock source except for OSC1 is active
HALT mode (when OSC1 only is active)	
RUN mode (when OSC1 only is active)	

The V_{D1} regulator also supports automatic mode in which the hardware detects a light load condition and automatically switches between normal mode and economy mode. Use the V_{D1} regulator in automatic mode when no special control is required.

2.1.4 V_{D1} Regulator/RFC Regulator Voltage Modes

The V_{D1} and RFC regulators support three voltage modes: 1.2 V, 1.4 V, and 1.8 V. The default voltage mode is 1.4 V. When operating at a low-speed clock, switching to 1.2 V reduces power consumption. For Flash memory or EEPROM erase/write operations, the voltage mode must be set to 1.8 V.

Procedure for switching to a lower voltage

1. Set the MODEN bit of each peripheral circuit to 0. (Disable peripheral operation)
2. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
3. Switch the system clock to a low-speed clock. (OSC1 or IOSC)
4. Stop OSC3 and EXOSC.
5. Configure the following PWGVD1CTL and PWGRFCCTL register bits:
 - Change the PWGVD1CTL.REGSEL[1:0] bits to the desired value. (Switch V_{D1}/RFC regulator voltage)
 - Set the PWGVD1CTL.REGMODE[1:0] bits to 0x2. (Set V_{D1} regulator to normal mode)
 - Set the PWGVD1CTL.REGDIS bit to 1. (Discharge V_{D1} pin)
 - Set the PWGRFCCTL.RFCREGDIS bit to 1. (Discharge V_{CAPRF} pin)
6. 500μs after changing voltage, configure the following PWGVD1CTL and PWGRFCCTL register bits:
 - Set the PWGVD1CTL.REGMODE[1:0] bits to 0x0. (Set V_{D1} regulator to automatic mode)
 - Set the PWGVD1CTL.REGDIS bit to 0. (Stop V_{D1} pin discharge)
 - Set the PWGRFCCTL.RFCREGDIS bit to 0. (Stop V_{CAPRF} pin discharge)
7. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Procedure for switching to a higher voltage

1. Set the MODEN bit of each peripheral circuit to 0. (Disable peripheral operation)
2. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
3. Configure the following PWGVD1CTL bits:
 - Change the PWGVD1CTL.REGSEL[1:0] bits to the desired value. (Switch V_{D1}/RFC regulator voltage)
 - Set the PWGVD1CTL.REGMODE[1:0] bits to 0x2. (Set V_{D1} regulator to normal mode)
4. 500μs after changing voltage, set the PWGVD1CTL.REGMODE[1:0] bits to 0x0. (Set V_{D1} regulator to automatic mode)
5. Switch the system clock to a high-speed clock if necessary.
6. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

2.2 System Reset Controller (SRC)

2.2.1 Overview

SRC is the system reset controller that resets the internal circuits according to the requests from the reset sources to archive steady IC operations. The main features of SRC are outlined below.

- Embedded reset hold circuit maintains reset state to boot the system safely while the internal power supply is unstable after power on or the oscillation frequency is unstable after the clock source is initiated.
- Supports reset requests from multiple reset sources.
 - #RESET pin
 - POR
 - Key-entry reset
 - Watchdog timer reset
 - Supply voltage detector reset
 - Peripheral circuit software reset (supports some peripheral circuits only)
- The CPU registers and peripheral circuit control bits will be reset with an appropriate initialization condition according to changes in status.

Figure 2.2.1.1 shows the SRC configuration.

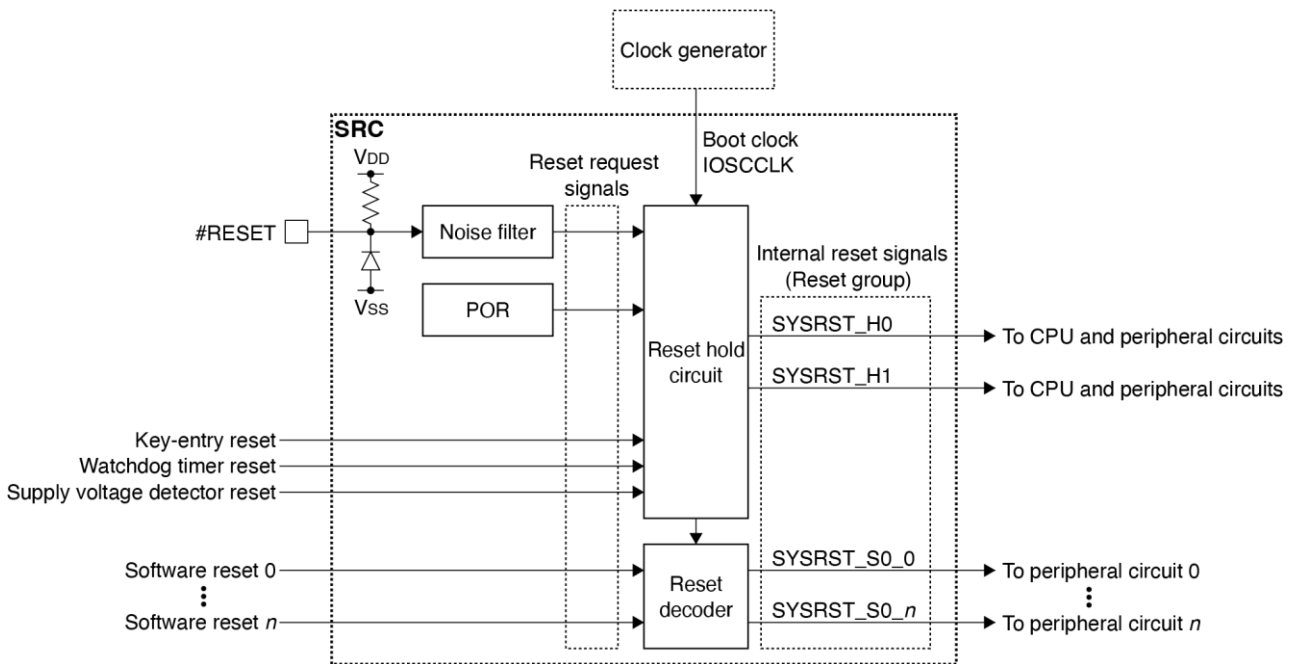


Figure 2.2.1.1 SRC Configuration

2.2.2 Input Pin

Table 2.2.2.1 shows the SRC pin.

Table 2.2.2.1 SRC Pin

Pin name	I/O	Initial status	Function
#RESET	I	I (Pull-up)	Reset input

The #RESET pin is connected to the noise filter that removes pulses not conforming to the requirements. An internal pull-up resistor is connected to the #RESET pin, so the pin can be left open. For the #RESET pin characteristics, refer to “#RESET pin characteristics” in the “Electrical Characteristics” chapter.

2 Power Supply, Reset, and Clocks

2.2.3 Reset Sources

The reset source refers to causes that request system initialization. The following shows the reset sources.

#RESET pin

Inputting a reset signal with a certain low level period to the #RESET pin issues a reset request.

POR

POR (Power On Reset) issues a reset request when the rise of V_{DD} is detected. A reset request from this circuit ensures that the system will be reset properly when the power is turned on. Figure 2.2.3.1 shows an example of POR internal reset operation according to variations in V_{DD} .

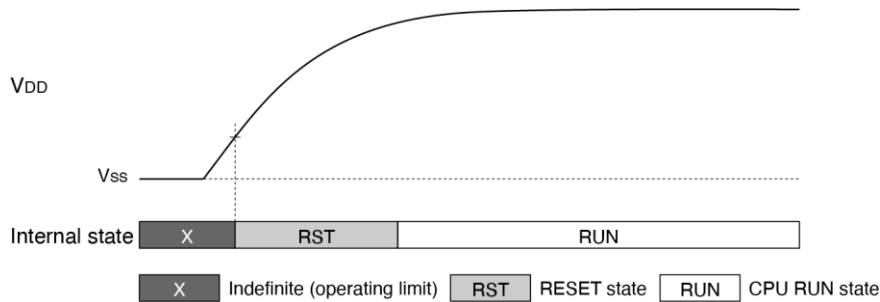


Figure 2.2.3.1 Example of Internal Reset by POR

For the POR electrical specifications, refer to “POR characteristics” in the “Electrical Characteristics” chapter.

Key-entry reset

Inputting a low level signal of a certain period to the I/O port pins configured to a reset input issues a reset request. This function must be enabled using an I/O port register. For more information, refer to the “I/O Ports” chapter.

Watchdog timer reset

Setting the watchdog timer into reset mode will issue a reset request when the counter overflows. This helps return the runaway CPU to a normal operating state. For more information, refer to the “Watchdog timer” chapter.

Supply voltage detector reset

By enabling the low power supply voltage detection reset function, the supply voltage detector will issue a reset request when a drop in the power supply voltage is detected. This makes it possible to put the system into reset state if the IC must be stopped under a low voltage condition. For more information, refer to the “Supply Voltage Detector” chapter.

Peripheral circuit software reset

Some peripheral circuits provide a control bit for software reset (MODEN or SFTRST). Setting this bit initializes the peripheral circuit control bits. Note, however, that the software reset operations depend on the peripheral circuit. For more information, refer to “Control Registers” in each peripheral circuit chapter.

Note: The MODEN bit of some peripheral circuits does not issue a software reset.

2.2.4 Initialization Conditions (Reset Groups)

A different initialization condition is set for the CPU registers and peripheral circuit control bits, individually. The reset group refers to an initialization condition. Initialization is performed when a reset source included in a reset group issues a reset request. Table 2.2.4.1 lists the reset groups. For the reset group to initialize the registers and control bits, refer to the “CPU and Debugger” chapter or “Control Registers” in each peripheral circuit chapter.

Table 2.2.4.1 List of Reset Groups

Reset group	Reset source	Reset cancelation timing
H0	#RESET pin	Reset state is maintained for the reset hold time t_{RSTR} after the reset request is canceled.
	POR	
	Key-entry reset	
	Supply voltage detector reset	
	Watchdog timer reset	
H1	#RESET pin	
	POR	
S0	Peripheral circuit software reset (MODEN and SFTRST) bits. The software reset operations depend on the peripheral circuit.	Reset state is canceled immediately after the reset request is canceled.

2.3 Clock Generator (CLG)

2.3.1 Overview

CLG is the clock generator that controls the clock sources and manages clock supply to the CPU and the peripheral circuits. The main features of CLG are outlined below.

- Supports multiple clock sources.
 - IOSC oscillator circuit supporting high-speed startup without external components
 - OSC1 oscillator circuit supporting low-power operation without external components
 - OSC3 oscillator circuit supporting high-speed operation without external components
 - EXOSC clock input circuit supporting square wave and sine wave inputs
- The system clock (SYSCLK), which is used as the operating clock for the CPU and bus, and the peripheral circuit operating clocks can be configured individually by selecting the suitable clock source and division ratio.
- IOSCCLK output from the IOSC oscillator circuit is used as the boot clock for fast booting.
- Controls the oscillator and clock input circuits to enable/disable according to the operating mode, RUN or SLEEP mode.
- Provides a flexible system clock switching function at SLEEP mode cancelation.
 - The clock sources to be stopped in SLEEP mode can be selected.
 - SYSCLK to be used at SLEEP mode cancelation can be selected from all clock sources.
 - The oscillator and clock input circuit on/off state can be maintained or changed at SLEEP mode cancelation.
- Provides the FOUT function to output an internal clock for driving external ICs or for monitoring the internal state.

Figure 2.3.1.1 shows the CLG configuration.

2 Power Supply, Reset, and Clocks

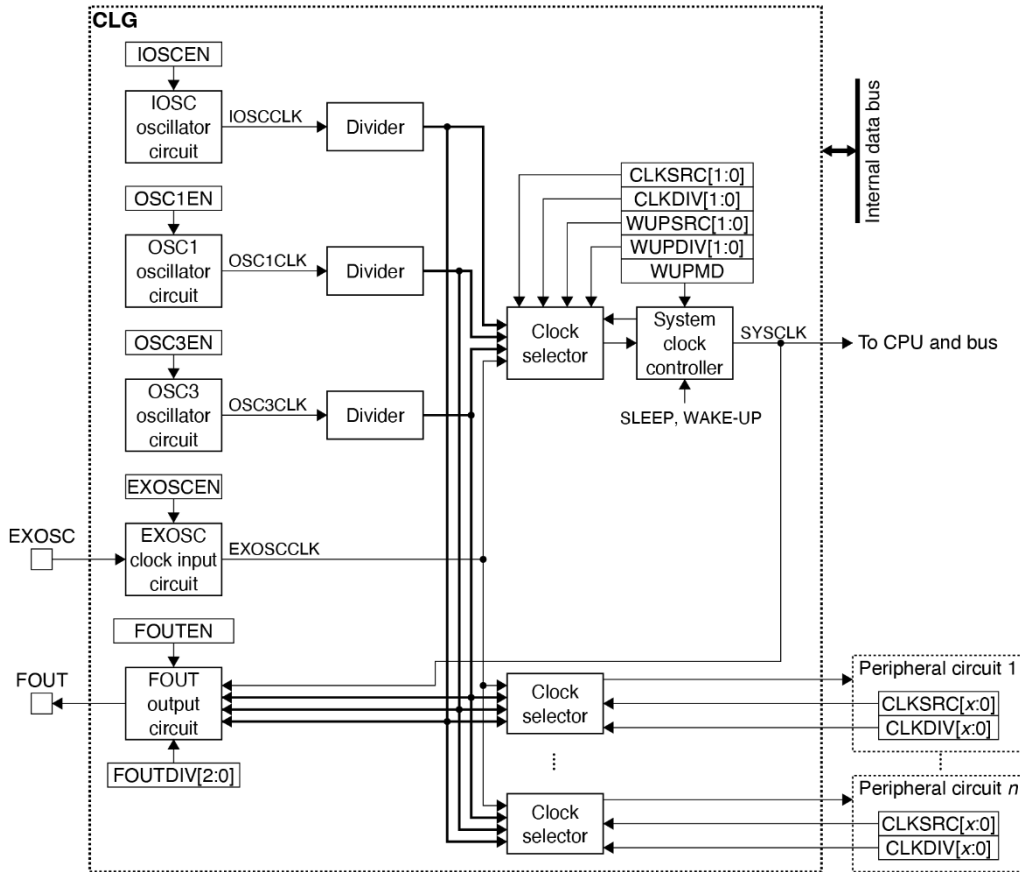


Figure 2.3.1.1 CLG Configuration

2.3.2 Input/Output Pins

Table 2.3.2.1 lists the CLG pins.

Table 2.3.2.1 List of CLG Pins

Pin name	I/O*	Initial status*	Function
EXOSC	I	I	EXOSC clock input
FOUT	O	O (L)	FOUT clock output

* Indicates the status when the pin is configured for CLG.

If the port is shared with the CLG input/output function and other functions, the CLG function must be assigned to the port. For more information, refer to the “I/O Ports” chapter.

2.3.3 Clock Sources

IOSC oscillator circuit

The IOSC oscillator circuit features a fast startup and no external parts are required for oscillating. Figure 2.3.3.1 shows the configuration of the IOSC oscillator circuit.

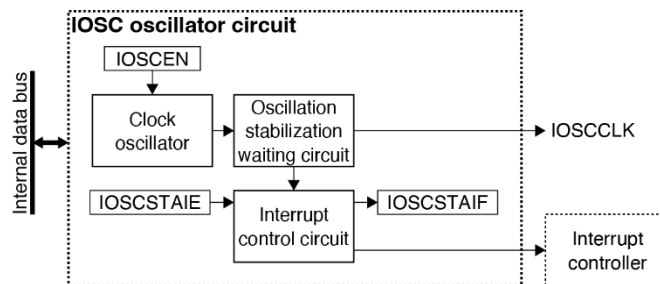


Figure 2.3.3.1 IOSC Oscillator Circuit Configuration

The IOSC oscillator circuit output clock IOSCCLK is used as SYSCLK at booting. For the oscillation characteristics, refer to “IOSC oscillator circuit characteristics” in the “Electrical Characteristics” chapter.

OSC1 oscillator circuit

The OSC1 oscillator circuit is a 32.3 kHz oscillator circuit that operates without any external parts. Figure 2.3.3.2 shows the configuration of the OSC1 oscillator circuit.

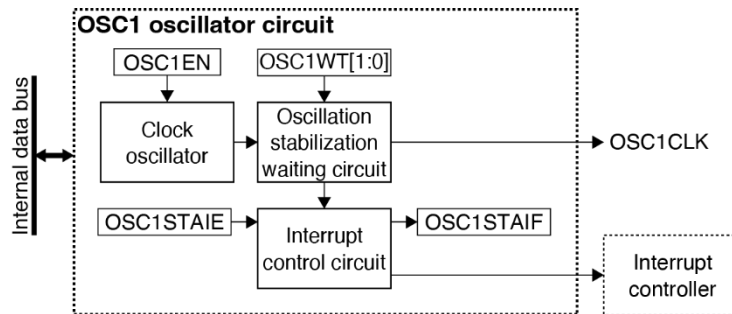


Figure 2.3.3.2 OSC1 Oscillator Circuit Configuration

For the oscillation characteristics, refer to the “OSC1 oscillator circuit characteristics” in the “Electrical Characteristics” chapter.

OSC3 oscillator circuit

The OSC3 oscillator circuit features a fast startup and no external parts are required for oscillating.

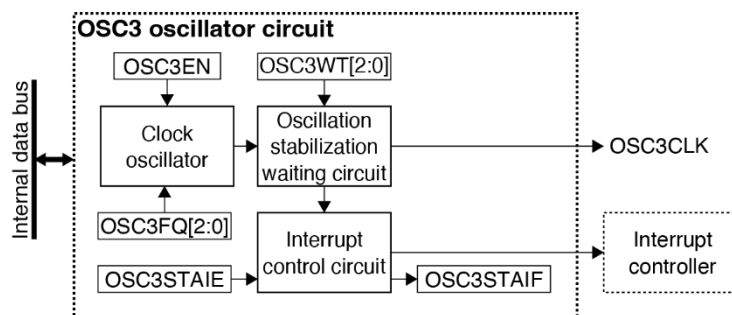


Figure 2.3.3.3 OSC3 Oscillator Circuit Configuration

For the oscillation characteristics, refer to the “OSC3 oscillator circuit characteristics” in the “Electrical Characteristics” chapter.

EXOSC clock input

EXOSC is an external clock input circuit that supports square wave and sine wave clocks. Figure 2.3.3.4 shows the configuration of the EXOSC clock input circuit.

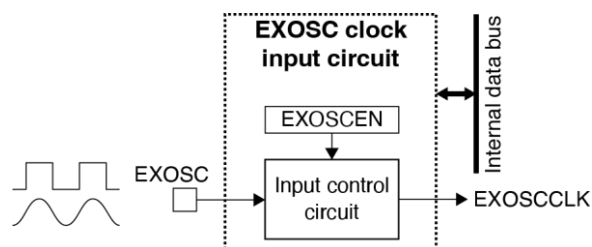


Figure 2.3.3.4 EXOSC Clock Input Circuit

EXOSC has no oscillation stabilization waiting circuit included, therefore, it must be enabled when a stabilized clock is being supplied. For the input clock characteristics, refer to “EXOSC external clock input characteristics” in the “Electrical Characteristics” chapter.

2 Power Supply, Reset, and Clocks

2.3.4 Operations

Oscillation start time and oscillation stabilization waiting time

The oscillation start time refers to the time after the oscillator circuit is enabled until the oscillation signal is actually sent to the internal circuits. The oscillation stabilization waiting time refers to the time it takes the clock to stabilize after the oscillation starts. To avoid malfunctions of the internal circuits due to an unstable clock during this period, the oscillator circuit includes an oscillation stabilization waiting circuit that can disable supplying the clock to the system until the designated time has elapsed. Figure 2.3.4.1 shows the relationship between the oscillation start time and the oscillation stabilization waiting time.

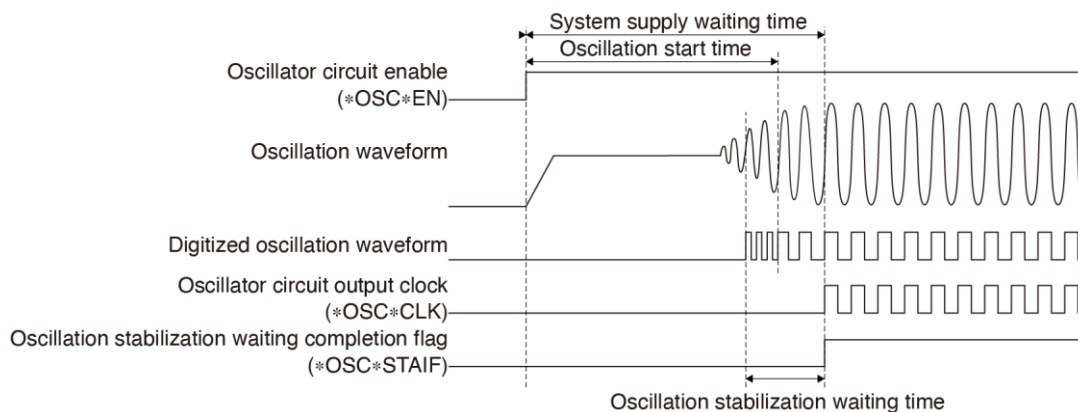


Figure 2.3.4.1 Oscillation Start Time and Oscillation Stabilization Waiting Time

The oscillation stabilization waiting times for the OSC1 and OSC3 oscillator circuits can be set using the CLGOSC1.OSC1WT[1:0] bits and CLGOSC3.OSC3WT[2:0] bits, respectively. To check whether the oscillation stabilization waiting time is set properly and the clock is stabilized immediately after the oscillation starts or not, monitor the oscillation clock using the FOUT output function. The oscillation stabilization waiting time for the IOSC oscillator circuit is fixed at 16 IOSCCLK clocks. The oscillation stabilization waiting time for the OSC1 oscillator circuit should be set to 4,096 OSC1CLK clocks or more. The oscillation stabilization waiting time for the OSC3 oscillator circuit should be set to 4 OSC3CLK clocks or more.

When the oscillation stabilization waiting operation has completed, the oscillator circuit sets the oscillation stabilization waiting completion flag and starts clock supply to the internal circuits.

Note: The oscillation stabilization waiting time is always expended at start of oscillation even if the oscillation stabilization waiting completion flag has not been cleared to 0.

Oscillation start procedure for the IOSC oscillator circuit

Follow the procedure shown below to start oscillation of the IOSC oscillator circuit.

1. Write 1 to the CLGINTE.IOSCSTAIF bit. (Clear interrupt flag)
2. Write 1 to the CLGINTE.IOSCSTAIE bit. (Enable interrupt)
3. Write 1 to the CLGOSC.IOSCEN bit. (Start oscillation)
4. IOSCCLK can be used if the CLGINTE.IOSCSTAIF bit = 1 after an interrupt occurs.

Oscillation start procedure for the OSC1 oscillator circuit

Follow the procedure shown below to start oscillation of the OSC1 oscillator circuit.

1. Write 1 to the CLGINTE.OSC1STAIF bit. (Clear interrupt flag)
2. Write 1 to the CLGINTE.OSC1STAIE bit. (Enable interrupt)
3. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
4. Configure the CLGOSC1.OSC1WT[1:0] bits. (Set oscillation stabilization waiting time)
5. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
6. Write 1 to the CLGOSC.OSC1EN bit. (Start oscillation)
7. OSC1CLK can be used if the CLGINTE.OSC1STAIF bit = 1 after an interrupt occurs.

Oscillation start procedure for the OSC3 oscillator circuit

Follow the procedure shown below to start oscillation of the OSC3 oscillator circuit.

1. Write 1 to the CLGINTF.OSC3STAIF bit. (Clear interrupt flag)
2. Write 1 to the CLGINTE.OSC3STAIE bit. (Enable interrupt)
3. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
4. Configure the following CLGOSC3 register bits:
 - CLGOSC3.OSC3WT[2:0] bits (Set oscillation stabilization waiting time)
 - CLGOSC3.OSC3FQ[2:0] bits (Set oscillation frequency)
5. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
6. Write 1 to the CLGOSC.OSC3EN bit. (Start oscillation)
7. OSC3CLK can be used if the CLGINTF.OSC3STAIF bit = 1 after an interrupt occurs.

System clock switching

The CPU boots using IOSCCLK as SYSCLK. After booting, the clock source of SYSCLK can be switched according to the processing speed required. The SYSCLK frequency can also be set by selecting the clock source division ratio, this makes it possible to run the CPU at the most suitable performance for the process to be executed. The CLGSCLK.CLKSRC[1:0] and CLGSCLK.CLKDIV[1:0] bits are used for this control.

The CLGSCLK register bits are protected against writings by the system protect function, therefore, the system protection must be removed by writing 0x0096 to the MSCPROT.PROT[15:0] bits before the register setting can be altered. For the transition between the operating modes including the system clock switching, refer to “Operating Mode.”

Clock control in SLEEP mode

The CPU enters SLEEP mode when it executes the slp instruction. Whether the clock sources being operated are stopped or not at this point can be selected in each source individually. This allows the CPU to fast switch between SLEEP mode and RUN mode, and the peripheral circuits to continue operating without disabling the clock in SLEEP mode. The CLGOSC.IOSCSLPC, CLGOSC.OSC1SLPC, CLGOSC.OSC3SLPC, and CLGOSC.EXOSCSLPC bits are used for this control. Figure 2.3.4.2 shows a control example.

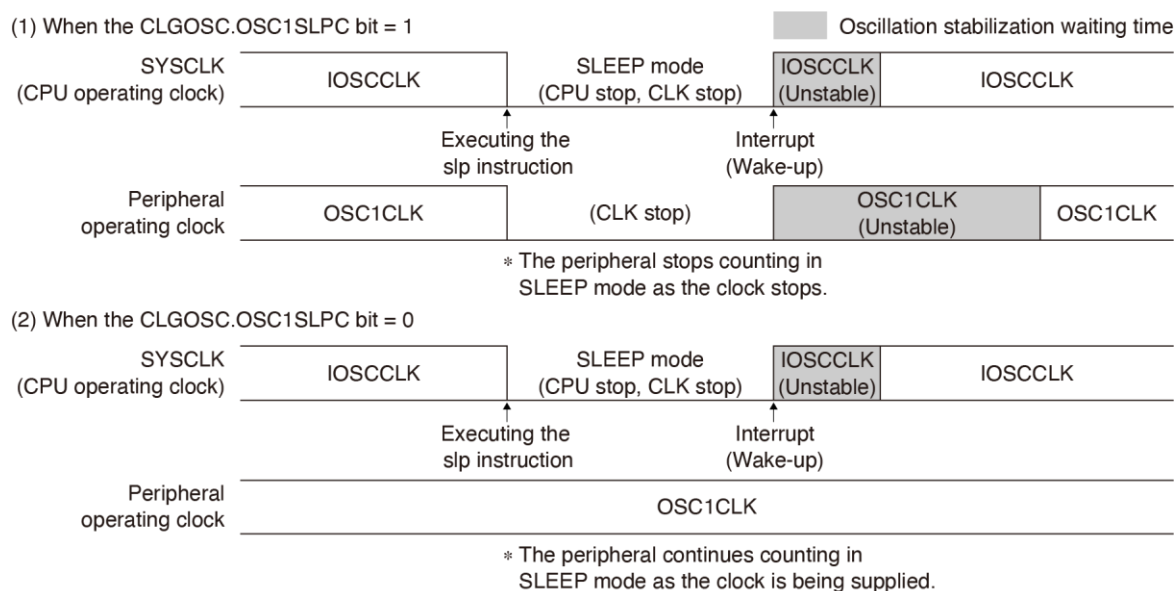


Figure 2.3.4.2 Clock Control Example in SLEEP Mode

The SYSCLK condition (clock source and division ratio) at wake-up from SLEEP mode to RUN mode can also be configured. This allows flexible clock control according to the wake-up process. Configure the clock using the CLGSCLK.WUPSRC[1:0] and CLGSCLK.WUPDIV[1:0] bits, and write 1 to the CLGSCLK.WUPMD bit to enable this function.

2 Power Supply, Reset, and Clocks

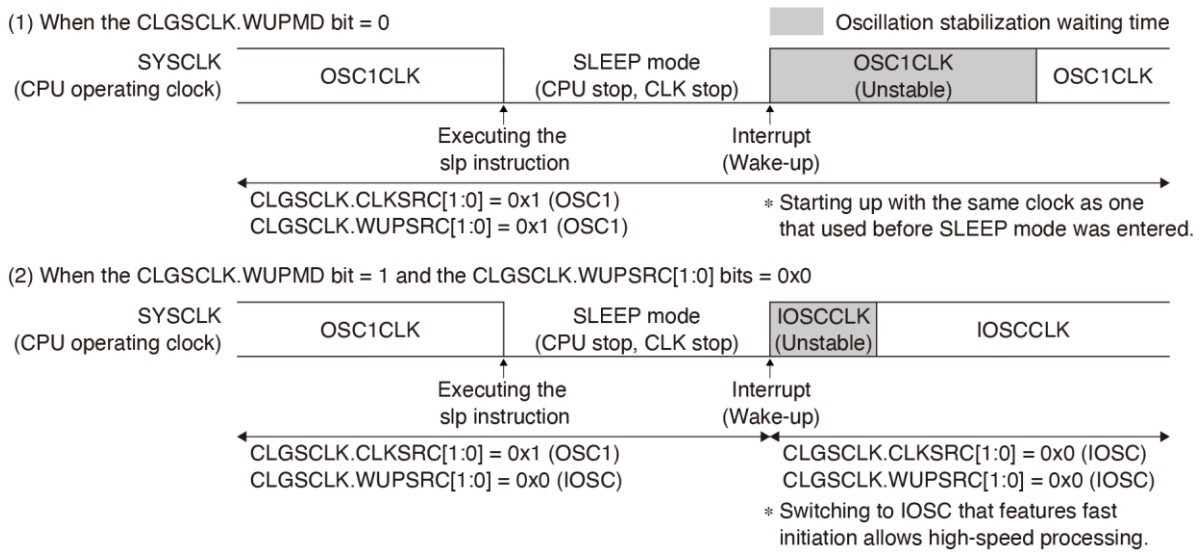


Figure 2.3.4.3 Clock Control Example at SLEEP Cancellation

Clock external output (FOUT)

The FOUT pin can output the clock generated by a clock source or its divided clock to outside the IC. This allows monitoring the oscillation frequency of the oscillator circuit or supplying an operating clock to external ICs. Follow the procedure shown below to start clock external output.

1. Assign the FOUT function to the port. (Refer to the “I/O Ports” chapter.)
2. Configure the following CLGFOUT register bits:
 - CLGFOUT.FOUTSRC[1:0] bits (Select clock source)
 - CLGFOUT.FOUTDIV[2:0] bits (Set clock division ratio)
 - Set the CLGFOUT.FOUTEN bit to 1. (Enable clock external output)

2.4 Operating Mode

2.4.1 Initial Boot Sequence

Figure 2.4.1.1 shows the initial boot sequence after power is turned on.

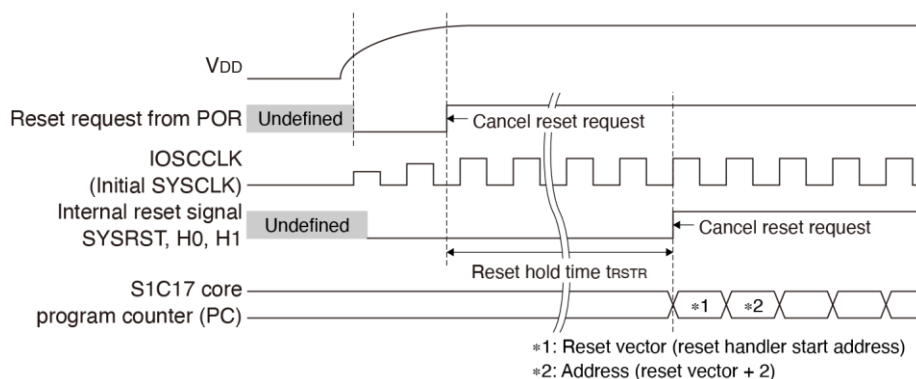


Figure 2.4.1.1 Initial Boot Sequence

Note: The reset cancellation time at power-on varies according to the power rise time and reset request cancellation time.

For the reset hold time t_{RSTR} , refer to “Reset hold circuit characteristics” in the “Electrical Characteristics” chapter.

2.4.2 Transition between Operating Modes

State transitions between operating modes shown in Figure 2.4.2.1 take place in this IC.

RUN mode

RUN mode refers to the state in which the CPU is executing the program. A transition to this mode takes place when the system reset request from the system reset controller is canceled. RUN mode is classified into “IOSC RUN,” “OSC1 RUN,” “OSC3 RUN,” and “EXOSC RUN” by the SYSCLK clock source.

HALT mode

When the CPU executes the halt instruction, it suspends program execution and stops operating. This state is HALT mode. In this mode, the clock sources and peripheral circuits keep operating. This mode can be set while no software processing is required and it reduces power consumption as compared with RUN mode. HALT mode is classified into “IOSC HALT,” “OSC1 HALT,” “OSC3 HALT,” and “EXOSC HALT” by the SYSCLK clock source.

SLEEP mode

When the CPU executes the slp instruction, it suspends program execution and stops operating. This state is SLEEP mode. In this mode, the clock sources stop operating as well. However, the clock source in which the CLGOSC.IOSCSLPC/OSC1SLPC/OSC3SLPC/EXOSCSLPC bit is set to 0 keeps operating, so the peripheral circuits with the clock being supplied can also operate. By setting this mode when no software processing and peripheral circuit operations are required, power consumption can be less than HALT mode.

The RAM retains data even in SLEEP mode.

Note: The current consumption when a clock source is active in SLEEP mode by setting the CLGOSC.IOSCSLPC/OSC1SLPC/OSC3SLPC/EXOSCSLPC bit to 0 is equivalent to the value in HALT mode with the same clock source condition (refer to “Current Consumption, Current consumption in HALT mode IHALT1, IHALT2, and IHALT3” in the “Electrical Characteristics” chapter).

DEBUG mode

When a debug interrupt occurs, the CPU enters DEBUG mode. DEBUG mode is canceled when the ret instruction is executed. For more information on DEBUG mode, refer to “Debugger” in the “CPU and Debugger” chapter.

Canceling HALT or SLEEP mode

The conditions listed below generate the HALT/SLEEP cancelation signal to cancel HALT or SLEEP mode and put the CPU into RUN mode. This transition is executed even if the CPU does not accept the interrupt request.

- Interrupt request from a peripheral circuit
- NMI from the watchdog timer
- Debug interrupt
- Reset request

2 Power Supply, Reset, and Clocks

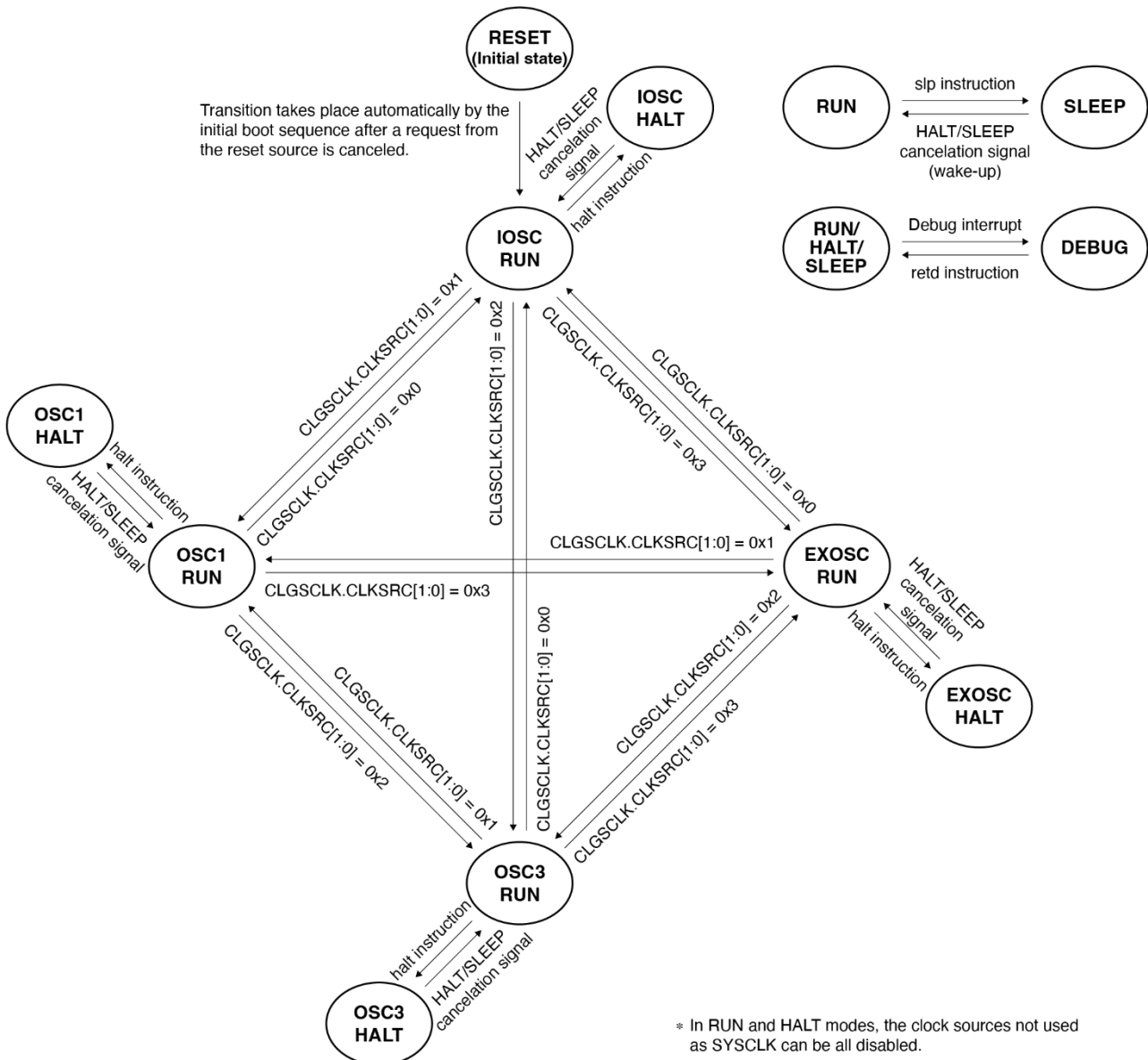


Figure 2.4.2.1 Operating Mode-to-Mode State Transition Diagram

2.5 Interrupts

CLG has a function to generate the interrupts shown in Table 2.5.1.

Table 2.5.1 CLG Interrupt Functions

Interrupt	Interrupt flag	Set condition	Clear condition
IOSC oscillation stabilization waiting completion	CLGINTF.IOSCSTAIF	When the IOSC oscillation stabilization waiting operation has completed after the oscillation starts	Writing 1
OSC1 oscillation stabilization waiting completion	CLGINTF.OSC1STAIF	When the OSC1 oscillation stabilization waiting operation has completed after the oscillation starts	Writing 1
OSC3 oscillation stabilization waiting completion	CLGINTF.OSC3STAIF	When the OSC3 oscillation stabilization waiting operation has completed after the oscillation starts	Writing 1

CLG provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

2.6 Control Registers

PWG2 V_{D1} Regulator Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PWGVD1CTL	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6	REGDIS	0	H0	R/WP	
	5–4	REGSEL[1:0]	0x1	H0	R/WP	
	3–2	–	0x0	–	R	
	1–0	REGMODE[1:0]	0x0	H0	R/WP	

Bits 15–7 Reserved

Bit 6 **REGDIS**

This bit controls the discharge function of the V_{D1} regulator.

1 (R/WP): Discharge enabled

0 (R/WP): Discharge disabled

Bits 5–4 **REGSEL[1:0]**

These bits control the V_{D1} and RFC regulator output voltages.

Table 2.6.1 V_{D1}/RFC Regulator Output Voltage

PWGVD1CTL.REGSEL[1:0] bits	Regulator output voltage
0x3	Reserved
0x2	1.8 V
0x1	1.4 V
0x0	1.2 V

Bits 3–2 Reserved

Bits 1–0 **REGMODE[1:0]**

These bits control the V_{D1} regulator operating mode.

Table 2.6.2 V_{D1} Regulator Operating Mode

PWGVD1CTL.REGMODE[1:0] bits	V _{D1} regulator operating mode
0x3	Economy mode
0x2	Normal mode
0x1	Reserved
0x0	Automatic mode

PWG2 RFC Regulator Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PWGRFCCTL	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6	RFCREGDIS	0	H0	R/WP	
	5–1	–	0x00	–	R	
	0	(reserved)	0	H0	R/WP	

Bits 15–7 Reserved

Bit 6 **RFCREGDIS**

This bit controls the discharge function of the RFC regulator.

1 (R/WP): Discharge enabled

0 (R/WP): Discharge disabled

Bits 5–0 Reserved

2 Power Supply, Reset, and Clocks

CLG System Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGSCLK	15	WUPMD	0	H0	R/WP	-
	14	-	0	-	R	
	13-12	WUPDIV[1:0]	0x0	H0	R/WP	
	11-10	-	0x0	-	R	
	9-8	WUPSRC[1:0]	0x0	H0	R/WP	
	7-6	-	0x0	-	R	
	5-4	CLKDIV[1:0]	0x0	H0	R/WP	
	3-2	-	0x0	-	R	
1-0	CLKSRC[1:0]	0x0	H0	R/WP		

Bit 15 WUPMD

This bit enables the SYSCLK switching function at wake-up.

1 (R/WP): Enable

0 (R/WP): Disable

When the CLGSCLK.WUPMD bit = 1, setting values of the CLGSCLK.WUPSRC[1:0] bits and the CLGSCLK.WUPDIV[1:0] bits are loaded to the CLGSCLK.CLKSRC[1:0] bits and the CLGSCLK.CLKDIV[1:0] bits, respectively, at wake-up from SLEEP mode to switch SYSCLK. When the CLGSCLK.WUPMD bit = 0, the CLGSCLK.CLKSRC[1:0] and CLGSCLK.CLKDIV[1:0] bits are not altered at wake-up.

Note: When the CLGSCLK.WUPMD bit = 1, the clock source enable bits (CLGOSC.EXOSCEN, CLGOSC.OSC1EN, CLGOSC.OSC3EN, CLGOSC.IOSCEN) except for the SYSCLK source selected by the CLGSCLK.CLKSRC[1:0] bits will be cleared to 0 to stop the clocks after a system wake-up. However, the enable bit of the clock source being operated during SLEEP mode by setting the CLGOSC.****SLPC bit retains 1 after a wake-up.

Bit 14 Reserved

Bits 13-12 WUPDIV[1:0]

These bits select the SYSCLK division ratio for resetting the CLGSCLK.CLKDIV[1:0] bits at wake-up. This setting is ineffective when the CLGSCLK.WUPMD bit = 0.

Bits 11-10 Reserved

Bits 9-8 WUPSRC[1:0]

These bits select the SYSCLK clock source for resetting the CLGSCLK.CLKSRC[1:0] bits at wake-up. When a currently stopped clock source is selected, it will automatically start oscillating or clock input at wake-up. However, this setting is ineffective when the CLGSCLK.WUPMD bit = 0.

Table 2.6.3 SYSCLK Clock Source and Division Ratio Settings at Wake-up

CLGSCLK. WUPDIV[1:0] bits	CLGSCLK.WUPSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCCLK	OSC1CLK	OSC3CLK	EXOSCCLK
0x3	1/8	Reserved	1/8	Reserved
0x2	1/4	Reserved	1/4	Reserved
0x1	1/2	1/2	1/2	Reserved
0x0	1/1	1/1	1/1	1/1

Bits 7-6 Reserved

Bits 5-4 CLKDIV[1:0]

These bits set the division ratio of the clock source to determine the SYSCLK frequency.

Bits 3-2 Reserved

Bits 1-0 CLKSRC[1:0]

These bits select the SYSCLK clock source.

When a currently stopped clock source is selected, it will automatically start oscillating or clock input.

Table 2.6.4 SYSCLK Clock Source and Division Ratio Settings

CLGSCLK. CLKDIV[1:0] bits	CLGSCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCCLK	OSC1CLK	OSC3CLK	EXOSCCLK
0x3	1/8	Reserved	1/8	Reserved
0x2	1/4	Reserved	1/4	Reserved
0x1	1/2	1/2	1/2	Reserved
0x0	1/1	1/1	1/1	1/1

CLG Oscillation Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGOSC	15–12	–	0x0	–	R	–
	11	EXOSCSLPC	1	H0	R/W	
	10	OSC3SLPC	1	H0	R/W	
	9	OSC1SLPC	1	H0	R/W	
	8	IOSCSLPC	1	H0	R/W	
	7–4	–	0x0	–	R	
	3	EXOSCEN	0	H0	R/W	
	2	OSC3EN	0	H0	R/W	
	1	OSC1EN	0	H0	R/W	
	0	IOSCEN	1	H0	R/W	

Bits 15–12 Reserved

- Bit 11** **EXOSCSLPC**
- Bit 10** **OSC3SLPC**
- Bit 9** **OSC1SLPC**
- Bit 8** **IOSCSLPC**

These bits control the clock source operations in SLEEP mode.

1 (R/W): Stop clock source in SLEEP mode

0 (R/W): Continue operation state before SLEEP

Each bit corresponds to the clock source as follows:

CLGOSC.EXOSCSLPC bit: EXOSC clock input

CLGOSC.OSC3SLPC bit: OSC3 oscillator circuit

CLGOSC.OSC1SLPC bit: OSC1 oscillator circuit

CLGOSC.IOSCSLPC bit: IOSC oscillator circuit

Bits 7–4 Reserved

- Bit 3** **EXOSCEN**
- Bit 2** **OSC3EN**
- Bit 1** **OSC1EN**
- Bit 0** **IOSCEN**

These bits control the clock source operation.

1(R/W): Start oscillating or clock input

0(R/W): Stop oscillating or clock input

Each bit corresponds to the clock source as follows:

CLGOSC.EXOSCEN bit: EXOSC clock input

CLGOSC.OSC3EN bit: OSC3 oscillator circuit

CLGOSC.OSC1EN bit: OSC1 oscillator circuit

CLGOSC.IOSCEN bit: IOSC oscillator circuit

2 Power Supply, Reset, and Clocks

CLG OSC1 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGOSC1	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1–0	OSC1WT[1:0]	0x1	H0	R/WP	

Bits 15–2 Reserved

Bits 1–0 OSC1WT[1:0]

These bits set the oscillation stabilization waiting time for the OSC1 oscillator circuit.

Table 2.6.5 OSC1 Oscillation Stabilization Waiting Time Setting

CLGOSC1.OSC1WT[1:0] bits	Oscillation stabilization waiting time
0x3	65,536 clocks
0x2	16,384 clocks
0x1	4,096 clocks
0x0	Reserved

CLG OSC3 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGOSC3	15–13	–	0x00	–	R	–
	12–10	OSC3FQ[2:0]	0x3	H0	R/WP	
	9–8	–	0x0	–	R	
	7–3	–	0x00	–	R	
	2–0	OSC3WT[2:0]	0x2	H0	R/WP	

Bits 15–13 Reserved

Bits 12–10 OSC3FQ[2:0]

These bits set the OSC3 oscillation frequency.

Table 2.6.6 OSC3 Oscillation Frequency Setting

CLGOSC3.OSC3FQ[2:0] bits	Oscillation frequency
0x7	Reserved
0x6	Reserved
0x5	250 kHz
0x4	384 kHz
0x3	4 MHz
0x2	2 MHz
0x1	1 MHz
0x0	500 kHz

Bits 9–3 Reserved

Bits 2–0 OSC3WT[2:0]

These bits set the oscillation stabilization waiting time for the OSC3 oscillator circuit.

Table 2.6.7 OSC3 Oscillation Stabilization Waiting Time Setting

CLGOSC3.OSC3WT[2:0] bits	Oscillation stabilization waiting time
0x7	65,536 clocks
0x6	16,384 clocks
0x5	4,096 clocks
0x4	1,024 clocks
0x3	256 clocks
0x2	64 clocks
0x1	16 clocks
0x0	4 clocks

CLG Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGINTF	15–8	–	0x00	–	R	Cleared by writing 1.
	7–3	–	0x0	H0	R	
	2	OSC3STAIF	0	H0	R/W	
	1	OSC1STAIF	0	H0	R/W	
	0	IOSCSTAIF	0	H0	R/W	

Bits 15–3 Reserved**Bit 2 OSC3STAIF****Bit 1 OSC1STAIF****Bit 0 IOSCSTAIF**

These bits indicate the CLG interrupt cause occurrence statuses.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

Each bit corresponds to the interrupt as follows:

CLGINTF.OSC3STAIF bit: OSC3 oscillation stabilization waiting completion interrupt

CLGINTF.OSC1STAIF bit: OSC1 oscillation stabilization waiting completion interrupt

CLGINTF.IOSCSTAIF bit: IOSC oscillation stabilization waiting completion interrupt

Note: The CLGINTF.IOSCSTAIF bit is 0 after system reset is canceled, but IOSCCCLK has already been stabilized.

CLG Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGINTE	15–8	–	0x00	–	R	
	7–3	–	0x0	–	R	
	2	OSC3STAIE	0	H0	R/W	
	1	OSC1STAIE	0	H0	R/W	
	0	IOSCSTAIE	0	H0	R/W	

Bits 15–3 Reserved**Bit 2 OSC3STAIE****Bit 1 OSC1STAIE****Bit 0 IOSCSTAIE**

These bits enable the CLG interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

Each bit corresponds to the interrupt as follows:

CLGINTE.OSC3STAIE bit: OSC3 oscillation stabilization waiting completion interrupt

CLGINTE.OSC1STAIE bit: OSC1 oscillation stabilization waiting completion interrupt

CLGINTE.IOSCSTAIE bit: IOSC oscillation stabilization waiting completion interrupt

CLG FOUT Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGFOUT	15–8	–	0x00	–	R	
	7	–	0	–	R	
	6–4	FOUTDIV[2:0]	0x0	H0	R/W	
	3–2	FOUTSRC[1:0]	0x0	H0	R/W	
	1	–	0	–	R	
	0	FOUTEN	0	H0	R/W	

Bits 15–7 Reserved**Bits 6–4 FOUTDIV[2:0]**

These bits set the FOUT clock division ratios.

2 Power Supply, Reset, and Clocks

Bits 3–2 FOUTSRC[1:0]

These bits select the FOUT clock sources.

Table 2.6.8 FOUT Clock Source and Division Ratio Settings

CLGFOUT. FOUTSRC[1:0] bits	CLGFOUT.FOUTSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCCLK	OSC1CLK	OSC3CLK	EXOSCCLK
0x7	1/128	1/32,768	1/128	Reserved
0x6	1/64	1/4,096	1/64	Reserved
0x5	1/32	1/1,024	1/32	Reserved
0x4	1/16	1/256	1/16	Reserved
0x3	1/8	1/8	1/8	Reserved
0x2	1/4	1/4	1/4	Reserved
0x1	1/2	1/2	1/2	Reserved
0x0	1/1	1/1	1/1	1/1

Note: When the CLGFOUT.FOUTSRC[1:0] bits are set to 0x3, the FOUT output will be stopped in SLEEP/HALT mode as SYSCLK is stopped.

Bit 1 Reserved

Bit 0 FOUTEN

This bit controls the FOUT clock external output.

1 (R/W): Enable external output

0 (R/W): Disable external output

Note: Since the FOUT signal generated is out of sync with writings to the CLGFOUT.FOUTEN bit, a glitch may occur when the FOUT output is enabled or disabled.

CLG Oscillation Frequency Trimming Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGTRIM1	15–14	–	0x0	–	R	–
	13–8	OSC1AJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.
	7–6	–	0x0	–	R	–
	5–0	IOSCAJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.

Bits 15–4 Reserved

Bits 13–8 OSC1AJ[5:0]

These bits set the frequency trimming value for the OSC1 internal oscillator circuit.

Bits 7–6 Reserved

Bits 5–0 IOSCAJ[5:0]

These bits set the frequency trimming value for the IOSC internal oscillator circuit.

Table 2.6.9 Oscillation Frequency Trimming Settings of OSC1/IOSC Internal Oscillator Circuits

CLGTRIM1.OSC1AJ[5:0] bits	OSC1 oscillation frequency
CLGTRIM1.IOSCAJ[5:0] bits	IOSC oscillation frequency
0x3f	High
:	:
0x00	Low

Note: The initial value of the CLGTRIM1.OSC1AJ[5:0]/IOSCAJ[5:0] bits was adjusted so that the OSC1/IOSC oscillator circuit characteristics described in the “Electrical Characteristics” chapter can be guaranteed. Be aware that the frequency characteristics may not be satisfied when this setting is altered. When altering this setting, always make sure that the OSC1/IOSC oscillator circuit is inactive.

CLG Oscillation Frequency Trimming Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGTRIM2	15–13	–	0x0	–	R	–
	12–8	OSC3AJ384[4:0]	*	H0	R/WP	* Determined by factory adjustment.
	7–5	–	0x0	–	R	–
	4–0	OSC3AJ4M[4:0]	*	H0	R/WP	* Determined by factory adjustment.

Bits 15–13 Reserved**Bits 12–8 OSC3AJ384[4:0]**

These bits set the frequency trimming value for the OSC3 internal oscillator circuit (oscillation frequency = 384 kHz)

Table 2.6.10 Oscillation Frequency Trimming Setting of OSC3 Internal Oscillator Circuit ($f_{\text{OSC3}} = 384 \text{ kHz}$)

CLGTRIM2.OSC3AJ384[4:0] bits	OSC3 internal oscillator frequency ($f_{\text{OSC3}} = 384 \text{ kHz}$)
0x1f	High
:	:
0x00	Low

Bits 7–5 Reserved**Bits 4–0 OSC3AJ4M[4:0]**

These bits set the frequency trimming value for the OSC3 internal oscillator circuit (oscillation frequency = 4 MHz/2 MHz/1 MHz/500 kHz/250 kHz)

Table 2.6.11 Oscillation Frequency Trimming Setting of OSC3 Internal Oscillator Circuit
($f_{\text{OSC3}} = 4 \text{ MHz}/2 \text{ MHz}/1 \text{ MHz}/500 \text{ kHz}/250 \text{ kHz}$)

CLGTRIM2.OSC3AJ4M[4:0] bits	OSC3 internal oscillator frequency ($f_{\text{OSC3}} = 4 \text{ MHz}/2 \text{ MHz}/1 \text{ MHz}/500 \text{ kHz}/250 \text{ kHz}$)
0x1f	High
:	:
0x00	Low

Note: The initial value of the CLGTRIM2.OSC3AJ384[4:0]/OSC3AJ4M[4:0] bits was adjusted so that the OSC3 oscillator circuit characteristics described in the “Electrical Characteristics” chapter can be guaranteed. Be aware that the frequency characteristic may not be satisfied when this setting is altered. When altering this setting, always make sure that the OSC3 oscillator circuit is inactive.

3 CPU and Debugger

3.1 Overview

This IC incorporates the Seiko Epson original 16-bit CPU core (S1C17) with a debugger. The CPU core's main features are listed below.

- Seiko Epson original 16-bit RISC processor
 - 24-bit general-purpose registers: 8
 - 24-bit special registers: 2
 - 8-bit special register: 1
 - Up to 16M bytes of memory space (24-bit address)
 - Harvard architecture using separated instruction bus and data bus
- Compact and fast instruction set optimized for development in C language
 - Code length: 16-bit fixed length
 - Number of instructions: 111 basic instructions (184 including variations)
 - Execution cycle: Main instructions are executed in one cycle.
 - Extended immediate instructions: Immediate data can be extended up to 24 bits.
- Supports reset, NMI, address misaligned, debug, and external interrupts.
 - Reads a vector from the vector table and branches to the interrupt handler routine directly.
 - Can generate software interrupts with a vector number specified (all vector numbers specifiable).
- HALT mode (halt instruction) and SLEEP mode (slp instruction) are provided as the standby function.
- Incorporates a debugger with three-wire communication interface to assist in software development.

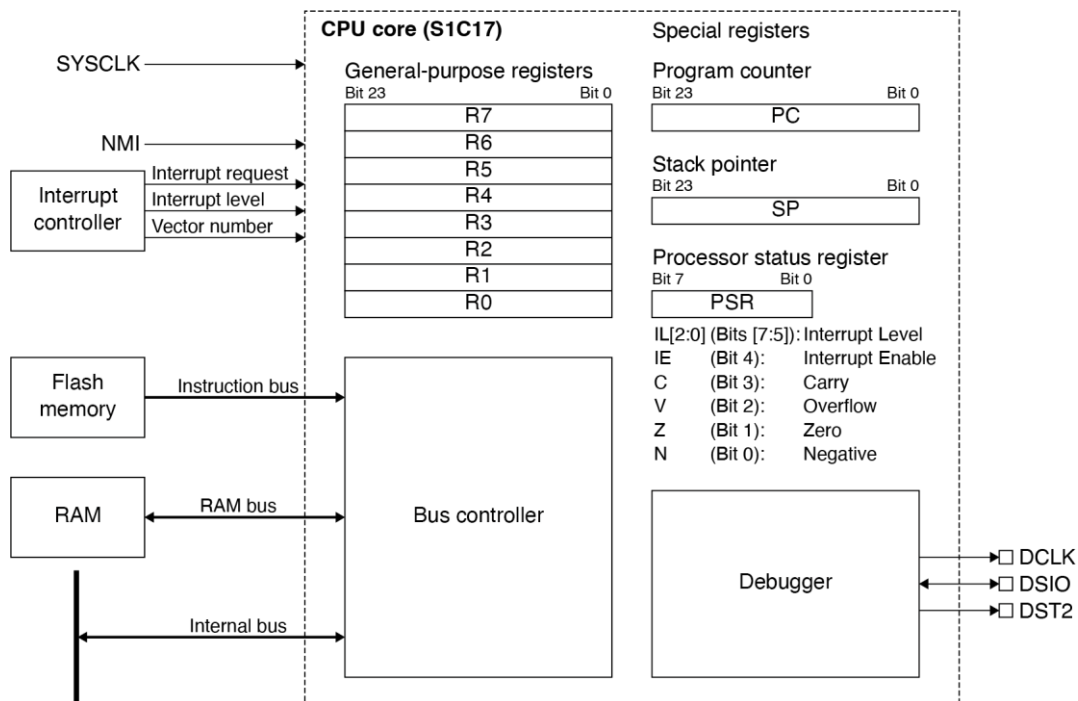


Figure 3.1.1 S1C17 Configuration

3 CPU and Debugger

3.2 CPU Core

3.2.1 CPU Registers

The CPU includes eight general-purpose registers and three special registers (Table 3.2.1.1).

Table 3.2.1.1 Initialization of CPU Registers

CPU register name			Initial	Reset
General-purpose registers		R0 to R7	0x000000	H0
Special registers	Program counter	PC	The reset vector is automatically loaded.	H0
	Stack pointer	SP	0x000000	H0
	Processor status register	PSR	0x00	H0

For details on the CPU registers, refer to the “S1C17 Family S1C17 Core Manual.” For more information on the reset vector, refer to the “Interrupt Controller” chapter.

3.2.2 Instruction Set

The CPU instruction codes are all fixed to 16 bits in length which, combined with pipelined processing, allows the most important instructions to be executed in one cycle. For details on the instructions, refer to the “S1C17 Family S1C17 Core Manual.”

3.2.3 Reading PSR

The PSR contents can be read through the MSCPSR register. Note, however, that data cannot be written to PSR through the MSCPSR register.

3.2.4 I/O Area Reserved for the S1C17 Core

The address range from 0xffffc00 to 0xfffffff is the I/O area reserved for the S1C17 core. Do not access this area except when it is required.

3.3 Debugger

3.3.1 Debugging Functions

The debugger provides the following functions:

- **Instruction break:** A debug interrupt is generated immediately before the set instruction address is executed. An instruction break can be set at up to four addresses.
- **Single step:** A debug interrupt is generated after each instruction has been executed.
- **Forcible break:** A debug interrupt is generated using an external input signal.
- **Software break:** A debug interrupt is generated when the brk instruction is executed.

When a debug interrupt occurs, the CPU enters DEBUG mode. The peripheral circuit operations in DEBUG mode depend on the setting of the DBRUN bit provided in the clock control register of each peripheral circuit. For more information on the DBRUN bit, refer to “Clock Supply in DEBUG Mode” in each peripheral circuit chapter. DEBUG mode continues until a cancel command is sent from the personal computer or the CPU executes the ret instruction. Neither hardware interrupts nor NMI are accepted during DEBUG mode.

3.3.2 Resource Requirements and Debugging Tools

Debugging work area

Debugging requires a 64-byte debugging work area. For more information on the work area location, refer to the “Memory and Bus” chapter. The start address of this debugging work area can be read from the DBRAM register.

Debugging tools

To perform debugging, connect ICDmini (S5U1C17001H) to the input/output pin for the debugger embedded in this IC and control it from the personal computer. This requires the tools shown below.

- S1C17 Family In-Circuit Debugger ICDmini (S5U1C17001H)
- S1C17 Family C Compiler Package (e.g., S5U1C17001C)

3.3.3 List of Debugger Input/Output Pins

Table 3.3.3.1 lists the debug pins.

Table 3.3.3.1 List of Debug Pins

Pin name	I/O	Initial state	Function
DCLK	O	O	On-chip debugger clock output pin Outputs a clock to the ICDmini (S5U1C17001H).
DSIO	I/O	I	On-chip debugger data input/output pin Used to input/output debugging data and input the break signal.
DST2	O	O	On-chip debugger status output pin Outputs the processor status during debugging.

The debugger input/output pins are shared with general-purpose I/O ports and are initially set as the debug pins. If the debugging function is not used, these pins can be switched to general-purpose I/O port pins. For details, refer to the “I/O Ports” chapter.

- Notes:
- Do not drive the DCLK pin with a high level from outside (e.g. pulling up with a resistor). Also, do not connect (short-circuit) between the DCLK pin and another GPIO port. In the both cases, the IC may not start up normally due to unstable pin input/output status at power on.
 - Do not drive the DSIO pin with a low level from outside, as it generates a debug interrupt that puts the CPU into DEBUG mode.

3.3.4 External Connection

Figure 3.3.4.1 shows a connection example between this IC and ICDmini when performing debugging.

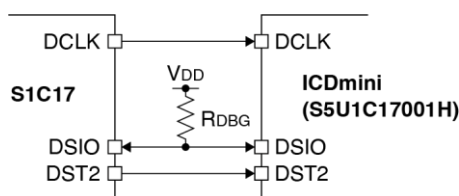


Figure 3.3.4.1 External Connection

For the recommended pull-up resistor value, refer to “Recommended Operating Conditions, DSIO pull-up resistor RDBG” in the “Electrical Characteristics” chapter. RDBG is not required when using the DSIO pin as a general-purpose I/O port pin.

3.3.5 Flash Security Function

This IC provides a security function to protect the internal Flash memory from unauthorized reading and tampering by using the debugger through ICDmini. Figure 3.3.5.1 shows a Flash security function setting flow.

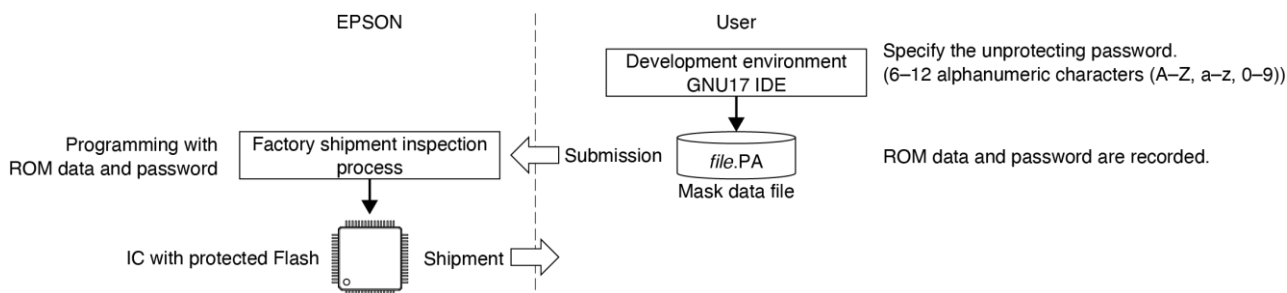


Figure 3.3.5.1 Shipment of IC with ROM Data Programmed and Flash Security Function Setting Flow

3 CPU and Debugger

The following shows the status of the IC with protected Flash:

- The Flash memory data is undefined if it is read from the debugger.
- An error occurs if an attempt is made to program the Flash memory through ICDmini.

However, the Flash security function can be disabled by entering the unprotecting password predefined to GNU17 IDE (the security function will take effect again after a reset). For setting the password, refer to the “(S1C17 Family C Compiler Package) S5U1C17001C Manual.”

Note: Disable the Flash security function before debugging an IC with protected Flash via ICDmini. The debugging functions may not run normally if the Flash security function is enabled.

3.4 Control Register

MISC PSR Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCPSR	15–8	–	0x00	–	R	–
	7–5	PSRIL[2:0]	0x0	H0	R	
	4	PSRIE	0	H0	R	
	3	PSRC	0	H0	R	
	2	PSRV	0	H0	R	
	1	PSRZ	0	H0	R	
	0	PSRN	0	H0	R	

Bits 15–8 Reserved

Bits 7–5 PSRIL[2:0]

The value (0 to 7) of the PSR IL[2:0] (interrupt level) bits can be read out with these bits.

Bit 4 PSRIE

The value (0 or 1) of the PSR IE (interrupt enable) bit can be read out with this bit.

Bit 3 PSRC

The value (0 or 1) of the PSR C (carry) flag can be read out with this bit.

Bit 2 PSRV

The value (0 or 1) of the PSR V (overflow) flag can be read out with this bit.

Bit 1 PSRZ

The value (0 or 1) of the PSR Z (zero) flag can be read out with this bit.

Bit 0 PSRN

The value (0 or 1) of the PSR N (negative) flag can be read out with this bit.

Debug RAM Base Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DBRAM	31–24	–	0x00	–	R	–
	23–0	DBRAM[23:0]	*1	H0	R	

*1 Debugging work area start address

Bits 31–24 Reserved

Bits 23–0 DBRAM[23:0]

The start address of the debugging work area (64 bytes) can be read out with these bits.

4 Memory and Bus

4.1 Overview

This IC supports up to 16M bytes of accessible memory space for both instructions and data. The features are listed below.

- Embedded Flash memory that supports on-board programming
- All memory and control registers are accessible in 16-bit width and one cycle (except for EEPROM).
- Write-protect function to protect system control registers

Figure 4.1.1 shows the memory map.

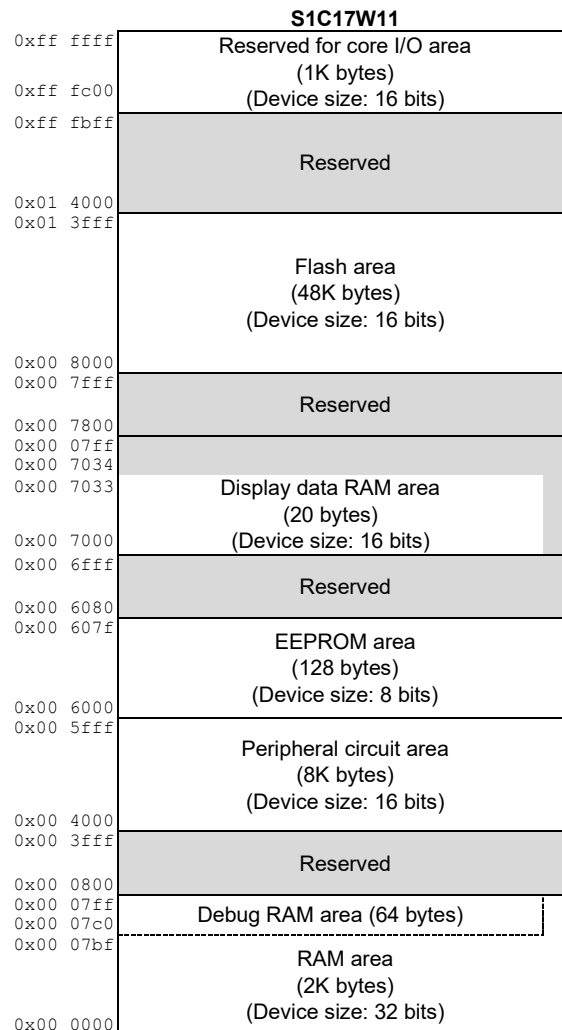


Figure 4.1.1 Memory Map

4.2 Bus Access Cycle

The CPU uses the system clock for bus access operations. First, “Bus access cycle,” “Device size,” and “Access size” are defined as follows:

- Bus access cycle: One system clock period = 1 cycle
- Device size: Bit width of the memory and peripheral circuits that can be accessed in one cycle
- Access size: Access size designated by the CPU instructions (e.g., `ld %rd, [%rb]` → 16-bit data transfer)

4 Memory and Bus

Table 4.2.1 lists numbers of bus access cycles by different device size and access size. The peripheral circuits can be accessed with an 8-bit, 16-bit, or 32-bit instruction.

Table 4.2.1 Number of Bus Access Cycles

Device size	Access size	Number of bus access cycles
8 bits	8 bits	1
	16 bits	2
	32 bits	4
16 bits	8 bits	1
	16 bits	1
	32 bits	2
32 bits	8 bits	1
	16 bits	1
	32 bits	1

Note: When data is transferred to a memory in 32-bit access, the eight high-order bits are written to the memory as 0x00 since the bit width of the S1C17 core general-purpose registers is 24 bits. Conversely when sending from a memory to a register, the eight high-order bits are ignored. The CPU performs 32-bit access for stack operations in an interrupt handling. In this case, the CPU read/write 32-bit data that consists of the PSR value as the eight high-order bits and the return address as the 24 low-order bits. For more information, refer to the “S1C17 Family S1C17 Core Manual.”

The CPU adopts Harvard architecture that allows simultaneous processing of an instruction fetch and a data access. However, they are not performed simultaneously under one of the conditions listed below. This prolongs the instruction fetch cycle for the number of data area bus cycles.

- When the CPU executes an instruction stored in the Flash area and accesses data in the Flash area
- When the CPU executes an instruction stored in the Flash area and accesses data in the display data RAM/EEPROM area
- When the CPU executes an instruction stored in the internal RAM/display data RAM area and accesses data in the internal RAM/display data RAM/EEPROM area

4.3 Flash Memory

The Flash memory is used to store application programs and data. Address 0x8000 in the Flash area is defined as the vector table base address by default, therefore a vector table must be located beginning from this address. For more information on the vector table, refer to “Vector Table” in the “Interrupt Controller” chapter.

4.3.1 Flash Memory Pin

Table 4.3.1.1 shows the Flash memory pin.

Table 4.3.1.1 Flash Memory Pin

Pin name	I/O	Initial status	Function
V _{PP}	P	–	Flash programming power supply

For the V_{PP} voltage, refer to “Recommended Operating Conditions, Flash programming voltage V_{PP}” in the “Electrical Characteristics” chapter.

Note: Do not apply external voltage to the V_{PP} pin except during Flash programming.

4.3.2 Flash Bus Access Cycle Setting

There is a limit of frequency to access the Flash memory with no wait cycle, therefore, the number of bus access cycles for reading must be changed according to the system clock frequency. The number of bus access cycles for reading can be configured using the FLASHCWAIT.RDWAIT[1:0] bits. Select a setting for higher frequency than the system clock.

4.3.3 Flash Programming

The Flash memory supports on-board programming, so it can be programmed with the ROM data by using the debugger through an ICDmini. Figure 4.3.3.1 shows connection diagrams for on-board programming.

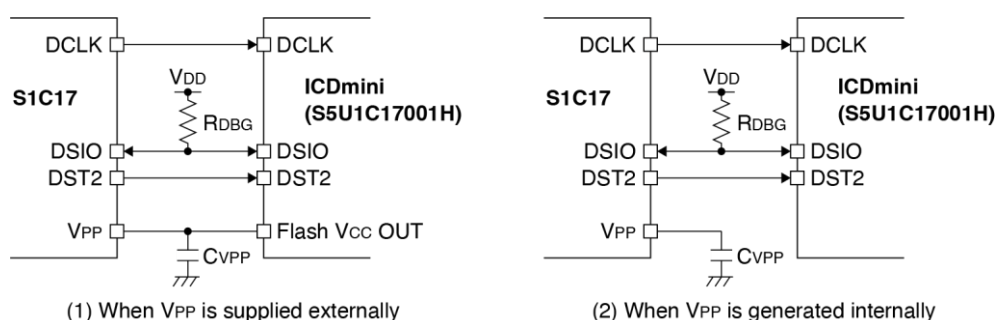


Figure 4.3.3.1 External Connection

The V_{PP} pin must be left open except when programming the Flash memory. However, it is not necessary to disconnect the wire when using ICDmini to supply the V_{PP} voltage, as ICDmini controls the power supply so that it will be supplied during Flash programming only. The V_{PP} voltage can also be generated by the internal power supply for generating the Flash programming voltage. Be sure to connect C_{VPP} for stabilizing the voltage when the V_{PP} voltage is supplied externally or for generating the voltage when the internal power supply is used.

For detailed information on ROM data programming method, refer to the “(S1C17 Family C Compiler Package) S5U1C17001C Manual.” The IC can also be shipped after being programmed in the factory with the ROM data developed. Should you desire to ship the IC with ROM data programmed from the factory, please contact our customer support.

- Notes:
- When programming the Flash memory, 2.2 V or more V_{DD} voltage is required.
 - Be sure to avoid using the V_{PP} pin output for driving external circuits when the V_{PP} voltage is generated internally.

4.4 EEPROM

This MCU includes an EEPROM that can be reprogrammed in one-byte units. This EEPROM supports 8-bit reading only, therefore, no instruction code can be stored.

4.4.1 EEPROM Pin

Table 4.4.1.1 shows the EEPROM pin.

Table 4.4.1.1 EEPROM Pin

Pin name	I/O	Initial status	Function
V_{PP}	P	–	EEPROM programming power supply

When reprogramming the EEPROM, the EEPROM controller (EEPROMC) uses the V_{PP} voltage generated internally. Connect C_{VPP} to the V_{PP} pin as shown in “(2) When V_{PP} is generated internally” of Figure 4.3.3.1. The notes described in Section 4.3.3 are also applied to EEPROM reprogramming.

4.4.2 Operations of EEPROM

Reprogramming EEPROM data

Follow the procedure below to reprogram the EEPROM.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Set the following EPRCCTL0 register bits:
 - Set the EPRCCTL0.EP_WMODE bit to 1. (Start reprogramming mode)
 - Set the EPRCCTL0.EP_PWRSET bit to 1. (Turn programming power supply on)

4 Memory and Bus

3. Wait for the programming power supply to stabilize (for the wait time, refer to “EEPROM Characteristics” in the “Electrical Characteristics” chapter.).
4. Write 1 to the EPRCINTF.RXBIF bit. (Clear interrupt flag)
5. Set the EPRCINTE.RXBIF bit to 1. (Enable interrupt)
6. Set the EPRCADR.EP_ADDR[7:0] bits. (Set reprogramming address)
7. Set the EPRCWDAT.EP_WDAT[7:0] bits. (Set programming data)
The programming data should be stored in the RAM for the verification to be performed later.
8. Write 1 to the EPRCCTL1.EP_CK bit. (Output clock pulse)
9. Wait for an interrupt.
When the reprogramming has completed, the EPRCINTF.RXBIF bit is set to 1.
10. Repeat Steps 4 to 9 for the addresses to be programmed.
11. Set the following EPRCCTL0 register bits:
 - Set the EPRCCTL0.EP_WMODE bit to 0. (Stop reprogramming mode)
 - Set the EPRCCTL0.EP_PWRSET bit to 0. (Turn programming power supply off)
12. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
13. Read EEPROM data and verify it with the programming data.

Reading EEPROM data

EEPROM data can be read from the memory area (logical address) where the EEPROM has been assigned using a standard 8-bit or 16-bit memory read instruction. If a 32-bit memory read instruction is used, only 16 bits are read from the EEPROM and the high-order bits are all set to 0. Note that EEPROM data is indefinite if it is read while the EPRCCTL0.EP_WMODE bit = 1 (reprogramming mode). When an ECC interrupt has occurred during reading data, the EPRCINTF.ECCERIF bit is set to 1.

Note: If an ECC interrupt has occurred, the reprogramming count may reach its limit. In this case, copy the data to another address and the address that generates an ECC interrupt should not be used in the subsequent reprogramming.

4.4.3 Interrupts

EEPROMC has a function to generate the interrupts shown in Table 4.4.3.1.

Table 4.4.3.1 EEPROMC Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Reprogram/read completion	EPRCINTF.RXBIF	When reprogramming/reading of the EEPROM has completed	Writing 1
ECC	EPRCINTF.ECCERIF	When data has been corrected via ECC during data reading	Writing 1

The EEPROMC provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

4.5 RAM

The RAM can be used to execute the instruction codes copied from another memory as well as storing variables or other data. This allows higher speed processing and lower power consumption than Flash memory.

Note: The 64 bytes at the end of the RAM is reserved as the debug RAM area. When using the debug functions under application development, do not access this area from the application program.
This area can be used for applications of mass-produced devices that do not need debugging.

The RAM size used by the application can be configured to equal or less than the implemented size using the MSCIRAMSZ.IRAMSZ[2:0] bits. For example, this function can be used to prevent creating programs that seek to access areas outside the RAM area of the target model when developing an application for a model in which the RAM size is smaller than this IC. After the limitation is applied, accessing an address outside the RAM area results in the same operation (undefined value is read out) as when a reserved area is accessed.

4.6 Display Data RAM

The embedded display data RAM is used to store display data for the LCD driver. Areas unused for display data in the display data RAM can be used as a general-purpose RAM. For specific information on the display data RAM, refer to “Display Data RAM” in the “LCD Driver” chapter.

4.7 Peripheral Circuit Control Registers

The control registers for the peripheral circuits are located in the 8K-byte area beginning with address 0x4000. Table 4.7.1 shows the control register map. For details of each control register, refer to “List of Peripheral Circuit Registers” in the appendix or “Control Registers” in each peripheral circuit chapter.

Table 4.7.1 Peripheral Circuit Control Register Map

Peripheral circuit	Address	Register name	
MISC registers (MISC)	0x4000	MSCPROT	MISC System Protect Register
	0x4002	MSCIRAMSZ	MISC IRAM Size Register
	0x4004	MSCTTBRL	MISC Vector Table Address Low Register
	0x4006	MSCTTBRLH	MISC Vector Table Address High Register
	0x4008	MSCPSR	MISC PSR Register
Power generator (PWG2)	0x4020	PWGVD1CTL	PWG2 V _{D1} Regulator Control Register
	0x4022	PWGRFCCTL	PWG2 RFC Regulator Control Register
Clock generator (CLG)	0x4040	CLGSCLK	CLG System Clock Control Register
	0x4042	CLGOSC	CLG Oscillation Control Register
	0x4046	CLGOSC1	CLG OSC1 Control Register
	0x4048	CLGOSC3	CLG OSC3 Control Register
	0x404c	CLGINTF	CLG Interrupt Flag Register
	0x404e	CLGINTE	CLG Interrupt Enable Register
	0x4050	CLGFOUT	CLG FOUT Control Register
	0x4052	CLGTRIM1	CLG Oscillation Frequency Trimming Register 1
	0x4054	CLGTRIM2	CLG Oscillation Frequency Trimming Register 2
Interrupt controller (ITC)	0x4080	ITCLV0	ITC Interrupt Level Setup Register 0
	0x4082	ITCLV1	ITC Interrupt Level Setup Register 1
	0x4084	ITCLV2	ITC Interrupt Level Setup Register 2
	0x4086	ITCLV3	ITC Interrupt Level Setup Register 3
	0x4088	ITCLV4	ITC Interrupt Level Setup Register 4
	0x408a	ITCLV5	ITC Interrupt Level Setup Register 5
	0x408c	ITCLV6	ITC Interrupt Level Setup Register 6
	0x408e	ITCLV7	ITC Interrupt Level Setup Register 7
	0x4090	ITCLV8	ITC Interrupt Level Setup Register 8
Watchdog timer (WDT2)	0x40a0	WDTCLK	WDT2 Clock Control Register
	0x40a2	WDTCTL	WDT2 Control Register
	0x40a4	WDTCMP	WDT2 Counter Compare Match Register
Supply voltage detector (SVD5)	0x4100	SVDCLK	SVD5 Clock Control Register
	0x4102	SVDCTL	SVD5 Control Register
	0x4104	SVDINTF	SVD5 Status and Interrupt Flag Register
	0x4106	SVDINTE	SVD5 Interrupt Enable Register
16-bit timer (T16) Ch.0	0x4160	T16_0CLK	T16 Ch.0 Clock Control Register
	0x4162	T16_0MOD	T16 Ch.0 Mode Register
	0x4164	T16_0CTL	T16 Ch.0 Control Register
	0x4166	T16_0TR	T16 Ch.0 Reload Data Register
	0x4168	T16_0TC	T16 Ch.0 Counter Data Register
	0x416a	T16_0INTF	T16 Ch.0 Interrupt Flag Register
	0x416c	T16_0INTE	T16 Ch.0 Interrupt Enable Register
Flash controller (FLASHC)	0x41b0	FLASHCWAIT	FLASHC Flash Read Cycle Register
EEPROM controller (EEPROMC)	0x41c0	EPRCTL0	EEPROMC Control Register 0
	0x41c2	EPRCTL1	EEPROMC Control Register 1
	0x41c4	EPRCADR	EEPROMC Address Register
	0x41c6	EPRCWAT	EEPROMC Write Data Register
	0x41c8	EPRCINTF	EEPROMC Interrupt Flag Register
	0x41ca	EPRCINTE	EEPROMC Interrupt Enable Register
I/O ports (PPORT)	0x4200	P0DAT	P0 Port Data Register
	0x4202	P0IOEN	P0 Port Enable Register
	0x4204	P0RCTL	P0 Port Pull-up/down Control Register

4 Memory and Bus

Peripheral circuit	Address	Register name		
I/O ports (PPORT)	0x4206	P0INTF	P0 Port Interrupt Flag Register	
	0x4208	P0INTCTL	P0 Port Interrupt Control Register	
	0x420a	P0CHATEN	P0 Port Chattering Filter Enable Register	
	0x420c	P0MODESEL	P0 Port Mode Select Register	
	0x420e	P0FNCSEL	P0 Port Function Select Register	
	0x4210	P1DAT	P1 Port Data Register	
	0x4212	P1IOEN	P1 Port Enable Register	
	0x4214	P1RCTL	P1 Port Pull-up/down Control Register	
	0x4216	P1INTF	P1 Port Interrupt Flag Register	
	0x4218	P1INTCTL	P1 Port Interrupt Control Register	
	0x421a	P1CHATEN	P1 Port Chattering Filter Enable Register	
	0x421c	P1MODESEL	P1 Port Mode Select Register	
	0x421e	P1FNCSEL	P1 Port Function Select Register	
	0x4220	P2DAT	P2 Port Data Register	
	0x4222	P2IOEN	P2 Port Enable Register	
	0x4224	P2RCTL	P2 Port Pull-up/down Control Register	
	0x4226	P2INTF	P2 Port Interrupt Flag Register	
	0x4228	P2INTCTL	P2 Port Interrupt Control Register	
	0x422a	P2CHATEN	P2 Port Chattering Filter Enable Register	
	0x422c	P2MODESEL	P2 Port Mode Select Register	
	0x422e	P2FNCSEL	P2 Port Function Select Register	
	0x42d0	PDDAT	Pd Port Data Register	
	0x42d2	PDIOEN	Pd Port Enable Register	
	0x42d4	PDRCTL	Pd Port Pull-up/down Control Register	
	0x42dc	PDMODESEL	Pd Port Mode Select Register	
	0x42de	PDFNCSEL	Pd Port Function Select Register	
	0x42e0	PCLK	P Port Clock Control Register	
	0x42e2	PINTFGRP	P Port Interrupt Flag Group Register	
	Universal port multiplexer (UPMUX)	0x4300	P0UPMUX0	P00–01 Universal Port Multiplexer Setting Register
		0x4302	P0UPMUX1	P02–03 Universal Port Multiplexer Setting Register
0x4304		P0UPMUX2	P04–05 Universal Port Multiplexer Setting Register	
0x4306		P0UPMUX3	P06–07 Universal Port Multiplexer Setting Register	
0x4308		P1UPMUX0	P10–11 Universal Port Multiplexer Setting Register	
0x430a		P1UPMUX1	P12–13 Universal Port Multiplexer Setting Register	
0x430c		P1UPMUX2	P14–15 Universal Port Multiplexer Setting Register	
0x430e		P1UPMUX3	P16–17 Universal Port Multiplexer Setting Register	
UART (UART3) Ch.0	0x4380	UA0CLK	UART3 Ch.0 Clock Control Register	
	0x4382	UA0MOD	UART3 Ch.0 Mode Register	
	0x4384	UA0BR	UART3 Ch.0 Baud-Rate Register	
	0x4386	UA0CTL	UART3 Ch.0 Control Register	
	0x4388	UA0TXD	UART3 Ch.0 Transmit Data Register	
	0x438a	UA0RXD	UART3 Ch.0 Receive Data Register	
	0x438c	UA0INTF	UART3 Ch.0 Status and Interrupt Flag Register	
	0x438e	UA0INTE	UART3 Ch.0 Interrupt Enable Register	
0x4390	UA0CAWF	UART3 Ch.0 Carrier Waveform Register		
16-bit timer (T16) Ch.1	0x43a0	T16_1CLK	T16 Ch.1 Clock Control Register	
	0x43a2	T16_1MOD	T16 Ch.1 Mode Register	
	0x43a4	T16_1CTL	T16 Ch.1 Control Register	
	0x43a6	T16_1TR	T16 Ch.1 Reload Data Register	
	0x43a8	T16_1TC	T16 Ch.1 Counter Data Register	
	0x43aa	T16_1INTF	T16 Ch.1 Interrupt Flag Register	
	0x43ac	T16_1INTE	T16 Ch.1 Interrupt Enable Register	
Synchronous serial interface (SPIA) Ch.0	0x43b0	SPI0MOD	SPIA Ch.0 Mode Register	
	0x43b2	SPI0CTL	SPIA Ch.0 Control Register	
	0x43b4	SPI0TXD	SPIA Ch.0 Transmit Data Register	
	0x43b6	SPI0RXD	SPIA Ch.0 Receive Data Register	
	0x43b8	SPI0INTF	SPIA Ch.0 Interrupt Flag Register	
	0x43ba	SPI0INTE	SPIA Ch.0 Interrupt Enable Register	
I ² C (I2C) Ch.0	0x43c0	I2C0CLK	I2C Ch.0 Clock Control Register	
	0x43c2	I2C0MOD	I2C Ch.0 Mode Register	
	0x43c4	I2C0BR	I2C Ch.0 Baud-Rate Register	
	0x43c8	I2C0OADR	I2C Ch.0 Own Address Register	
	0x43ca	I2C0CTL	I2C Ch.0 Control Register	
	0x43cc	I2C0TXD	I2C Ch.0 Transmit Data Register	

Peripheral circuit	Address	Register name	
I ² C (I2C) Ch.0	0x43ce	I2C0RXD	I2C Ch.0 Receive Data Register
	0x43d0	I2C0INTF	I2C Ch.0 Status and Interrupt Flag Register
	0x43d2	I2C0INTE	I2C Ch.0 Interrupt Enable Register
16-bit PWM timer (T16B) Ch.0	0x5000	T16B0CLK	T16B Ch.0 Clock Control Register
	0x5002	T16B0CTL	T16B Ch.0 Counter Control Register
	0x5004	T16B0MC	T16B Ch.0 Max Counter Data Register
	0x5006	T16B0TC	T16B Ch.0 Timer Counter Data Register
	0x5008	T16B0CS	T16B Ch.0 Counter Status Register
	0x500a	T16B0INTF	T16B Ch.0 Interrupt Flag Register
	0x500c	T16B0INTE	T16B Ch.0 Interrupt Enable Register
	0x5010	T16B0CCCTL0	T16B Ch.0 Compare/Capture 0 Control Register
	0x5012	T16B0CCR0	T16B Ch.0 Compare/Capture 0 Data Register
	0x5018	T16B0CCCTL1	T16B Ch.0 Compare/Capture 1 Control Register
	0x501a	T16B0CCR1	T16B Ch.0 Compare/Capture 1 Data Register
16-bit PWM timer (T16B) Ch.1	0x5040	T16B1CLK	T16B Ch.1 Clock Control Register
	0x5042	T16B1CTL	T16B Ch.1 Counter Control Register
	0x5044	T16B1MC	T16B Ch.1 Max Counter Data Register
	0x5046	T16B1TC	T16B Ch.1 Timer Counter Data Register
	0x5048	T16B1CS	T16B Ch.1 Counter Status Register
	0x504a	T16B1INTF	T16B Ch.1 Interrupt Flag Register
	0x504c	T16B1INTE	T16B Ch.1 Interrupt Enable Register
	0x5050	T16B1CCCTL0	T16B Ch.1 Compare/Capture 0 Control Register
	0x5052	T16B1CCR0	T16B Ch.1 Compare/Capture 0 Data Register
	0x5058	T16B1CCCTL1	T16B Ch.1 Compare/Capture 1 Control Register
	0x505a	T16B1CCR1	T16B Ch.1 Compare/Capture 1 Data Register
16-bit timer (T16) Ch.2	0x5260	T16_2CLK	T16 Ch.2 Clock Control Register
	0x5262	T16_2MOD	T16 Ch.2 Mode Register
	0x5264	T16_2CTL	T16 Ch.2 Control Register
	0x5266	T16_2TR	T16 Ch.2 Reload Data Register
	0x5268	T16_2TC	T16 Ch.2 Counter Data Register
	0x526a	T16_2INTF	T16 Ch.2 Interrupt Flag Register
	0x526c	T16_2INTE	T16 Ch.2 Interrupt Enable Register
Sound generator (SNDA2)	0x5300	SNDCLK	SNDA2 Clock Control Register
	0x5302	SNDSEL	SNDA2 Select Register
	0x5304	SNDCTL	SNDA2 Control Register
	0x5306	SNDDAT	SNDA2 Data Register
	0x5308	SNDINTF	SNDA2 Interrupt Flag Register
	0x530a	SNDINTE	SNDA2 Interrupt Enable Register
Charge pump regulator (CHREG)	0x5340	CHREGCONF	CHREG Configuration Register
	0x5342	CHREGCTL	CHREG Control Register
LCD driver (LCD4B)	0x5400	LCD4CLK	LCD4B Clock Control Register
	0x5402	LCD4CTL	LCD4B Control Register
	0x5404	LCD4TIM1	LCD4B Timing Control Register 1
	0x5406	LCD4TIM2	LCD4B Timing Control Register 2
	0x5408	LCD4PWR	LCD4B Power Control Register
	0x540a	LCD4DSP	LCD4B Display Control Register
	0x540c	LCD4COMC0	LCD4B COM Pin Control Register 0
	0x5410	LCD4INTF	LCD4B Interrupt Flag Register
	0x5412	LCD4INTE	LCD4B Interrupt Enable Register
R/F converter (RFC2) Ch.0	0x5440	RFC0CLK	RFC2 Ch.0 Clock Control Register
	0x5442	RFC0CTL	RFC2 Ch.0 Control Register
	0x5444	RFC0TRG	RFC2 Ch.0 Oscillation Trigger Register
	0x5446	RFC0MCL	RFC2 Ch.0 Measurement Counter Low Register
	0x5448	RFC0MCH	RFC2 Ch.0 Measurement Counter High Register
	0x544a	RFC0TCL	RFC2 Ch.0 Time Base Counter Low Register
	0x544c	RFC0TCH	RFC2 Ch.0 Time Base Counter High Register
	0x544e	RFC0INTF	RFC2 Ch.0 Interrupt Flag Register
	0x5450	RFC0INTE	RFC2 Ch.0 Interrupt Enable Register

4 Memory and Bus

4.7.1 System-Protect Function

The system-protect function protects control registers and bits from writings. They cannot be rewritten unless write protection is removed by writing 0x0096 to the MSCPROT.PROT[15:0] bits. This function is provided to prevent deadlock that may occur when a system-related register is altered by a runaway CPU. See “Control Registers” in each peripheral circuit to identify the registers and bits with write protection.

Note: Once write protection is removed using the MSCPROT.PROT[15:0] bits, write enabled status is maintained until write protection is applied again. After the registers/bits required have been altered, apply write protection.

4.8 Control Registers

MISC System Protect Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCPROT	15–0	PROT[15:0]	0x0000	H0	R/W	–

Bits 15–0 PROT[15:0]

These bits protect the control registers related to the system against writings.

0x0096 (R/W): Disable system protection

Other than 0x0096 (R/W): Enable system protection

While the system protection is enabled, any data will not be written to the affected control bits (bits with “WP” or “R/WP” appearing in the R/W column).

MISC IRAM Size Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCIRAMSZ	15–9	–	0x00	–	R	–
	8	(reserved)	0	H0	R/WP	Always set to 0.
	7–3	–	0x04	–	R	–
	2–0	IRAMSZ[2:0]	0x2	H0	R/WP	–

Bits 15–3 Reserved

Bits 2–0 IRAMSZ[2:0]

These bits set the internal RAM size that can be used.

Table 4.8.1 Internal RAM Size Selections

MSCIRAMSZ.IRAMSZ[2:0] bits	Internal RAM size
0x7–0x3	Reserved
0x2	2KB
0x1	1KB
0x0	512B

FLASHC Flash Read Cycle Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
FLASHCWAIT	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	–
	1–0	RDWAIT[1:0]	0x1	H0	R/WP	–

Bits 15–2 Reserved

Bits 1–0 RDWAIT[1:0]

These bits set the number of bus access cycles for reading from the Flash memory.

Table 4.8.2 Setting Number of Bus Access Cycles for Flash Read

$V_{DD} = 1.2$ to 1.6 V

FLASHCWAIT.RDWAIT[1:0] bits	Number of bus Access cycles	System clock frequency
0x3	4	1.1 MHz (max.)
0x2	3	1.1 MHz (max.)
0x1	2	1.1 MHz (max.)
0x0	1	800 kHz (max.)

$V_{DD} = 1.6$ to 3.6 V, $V_{D1} = 1.4$ V

FLASHWAIT.RDWAIT[1:0] bits	Number of bus Access cycles	System clock frequency
0x3	4	4.2 MHz (max.)
0x2	3	4.2 MHz (max.)
0x1	2	4.2 MHz (max.)
0x0	1	2.1 MHz (max.)

Note: Be sure to set the FLASHWAIT.RDWAIT[1:0] bits before the system clock is configured.

EEPROMC Control Register 0

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCCTL0	15–9	–	0x00	–	R	–
	8	EP_XPOR	1	H0	R/WP	
	7–2	–	0x00	–	R	
	1	EP_PWRSET	0	H0	R/WP	
	0	EP_WMODE	0	H0	R/WP	

Bits 15–9 Reserved

Bit 8 EP_XPOR

This bit controls the reset signal of the EEPROM.

1 (R/W): Negate the reset signal.

0 (R/W): Assert the reset signal.

When a reprogram/read completion interrupt has not occurred (EPRCINTF.RXBIF bit has not been set to 1) after starting an EEPROM reprogramming operation, write 0 to this bit to reset the EEPROM.

While this bit is 0, the EEPROM control functions including reading of the EEPROM are all disabled. To resume the EEPROM operations again, write 1 to this bit to cancel the reset state after waiting for a longer time than the effective EEPROM reset pulse width t_{XPOR} (refer to “EEPROM Characteristics” in the “Electrical Characteristics” chapter).

Bits 7–2 Reserved

Bit 1 EP_PWRSET

This bit controls the programming power supply.

1 (R/W): Programming power supply ON

0 (R/W): Programming power supply OFF

When this bit is set to 1, the EEPROM programming power supply circuit goes on and it generates the EEPROM programming voltage by boosting the V_{PP} voltage. This bit is effective when the EPRCCTL0.EP_WMODE bit = 1.

Bit 0 EP_WMODE

This bit starts/stops reprogramming mode.

1 (R/W): Start reprogramming mode

0 (R/W): Stop reprogramming mode

Setting this bit to 1 puts the EEPROM into reprogramming mode to enable data reprogramming.

Note that read data are indefinite when the EEPROM is read while this bit is 1.

EEPROMC Control Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCCTL1	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	EP_CK	0	H0	WP	

Bits 15–1 Reserved

Bit 0 EP_CK

This bit controls the clock pulse output to reprogram the EEPROM.

1 (W): Output one clock pulse

0 (W): Ineffective

4 Memory and Bus

Writing 1 to this bit outputs a clock to the EEPROM to reprogram the EEPROM address specified by the EPRCADR.EP_ADDR[7:0] bits with the data specified by the EPRCWDAT.EP_WDAT[7:0] bits. This bit automatically reverts to 0 after writing 1.

This bit is effective when the EPRCTL0.EP_WMODE bit = 1.

EEPROMC Address Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCADR	15–8	–	0x00	–	R	–
	7–0	EP_ADDR[7:0]	0x00	H0	R/WP	

Bits 15–8 Reserved

Bits 7–0 EP_ADDR[7:0]

These bits specify the EEPROM physical address (0 to 255) to be reprogrammed.

EEPROM (logical) address = 0x6000 + EPRCADR.EP_ADDR[7:0] bits

EEPROMC Write Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCWDAT	15–8	–	0x00	–	R	–
	7–0	EP_WDAT[7:0]	0x00	H0	R/WP	

Bits 15–8 Reserved

Bits 7–0 EP_WDAT[7:0]

These bits specify the 8-bit data to program the EEPROM.

EEPROMC Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCINTF	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	ECCERIF	0	H0	R/WP	Cleared by writing 1.
	0	RXBIF	0	H0	R/WP	

Bits 15–2 Reserved

Bit 1 ECCERIF

Bit 0 RXBIF

These bits indicate the EEPROMC interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

EPRCINTF.ECCERIF bit: ECC interrupt

EPRCINTF.RXBIF bit: Reprogram/read completion interrupt

EEPROMC Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
EPRCINTE	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	ECCERIE	0	H0	R/WP	
	0	RXBIE	0	H0	R/WP	

Bits 15–2 Reserved

Bit 1 ECCERIE

Bit 0 RXBIE

These bits enable EEPROMC interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

EPRCINTE.ECCERIE bit: ECC interrupt

EPRCINTE.RXBIE bit: Reprogram/read completion interrupt

Note: To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

5 Interrupt Controller (ITC)

5.1 Overview

The features of the ITC are listed below.

- Honors interrupt requests from the peripheral circuits and outputs the interrupt request, interrupt level and vector number signals to the CPU.
- The interrupt level of each interrupt source is selectable from among eight levels.
- Priorities of the simultaneously generated interrupts are established from the interrupt level.
- Handles the simultaneously generated interrupts with the same interrupt level as smaller vector number has higher priority.

Figure 5.1.1 shows the configuration of the ITC.

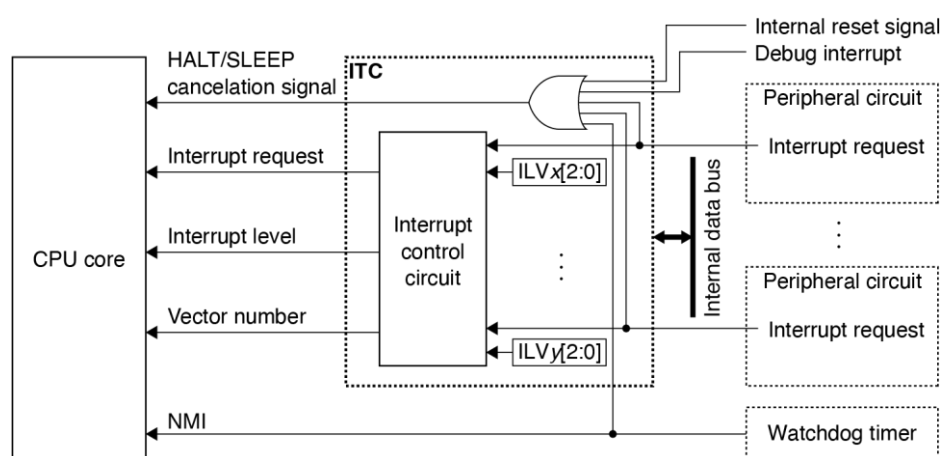


Figure 5.1.1 ITC Configuration

5.2 Vector Table

The vector table contains the vectors to the interrupt handler routines (handler routine start address) that will be read by the CPU to execute the handler when an interrupt occurs.

Table 5.2.1 shows the vector table.

Table 5.2.1 Vector Table

TTBR initial value = 0x8000

Vector number/ Software interrupt number	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
0 (0x00)	TTBR + 0x00	Reset	<ul style="list-style-type: none"> • Low input to the #RESET pin • Power-on reset • Key reset • Watchdog timer overflow *2 • Supply voltage detector reset 	1
1 (0x01)	TTBR + 0x04	Address misaligned interrupt	Memory access instruction	2
–	(0xffc00)	Debugging interrupt	brk instruction, etc.	3
2 (0x02)	TTBR + 0x08	NMI	Watchdog timer overflow *2	4
3 (0x03)	TTBR + 0x0c	Reserved for C compiler	–	–

5 Interrupt Controller (ITC)

Vector number/ Software interrupt number	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
4 (0x04)	TTBR + 0x10	Supply voltage detector interrupt	Low power supply voltage detection	High *1 ↑
5 (0x05)	TTBR + 0x14	Port interrupt	Port input	
6 (0x06)	TTBR + 0x18	reserved	—	
7 (0x07)	TTBR + 0x1c	Clock generator interrupt	<ul style="list-style-type: none"> • IOSC oscillation stabilization waiting completion • OSC1 oscillation stabilization waiting completion • OSC3 oscillation stabilization waiting completion 	
8 (0x08)	TTBR + 0x20	reserved	—	
9 (0x09)	TTBR + 0x24	16-bit timer Ch.0 interrupt	Underflow	
10 (0x0a)	TTBR + 0x28	UART Ch.0 interrupt	<ul style="list-style-type: none"> • End of transmission • Framing error • Parity error • Overrun error • Receive buffer two bytes full • Receive buffer one byte full • Transmit buffer empty 	
11 (0x0b)	TTBR + 0x2c	16-bit timer Ch.1 interrupt	Underflow	
12 (0x0c)	TTBR + 0x30	Synchronous serial interface Ch.0 interrupt	<ul style="list-style-type: none"> • End of transmission • Receive buffer full • Transmit buffer empty • Overrun error 	
13 (0x0d)	TTBR + 0x34	I ² C interrupt	<ul style="list-style-type: none"> • End of data transfer • General call address reception • NACK reception • STOP condition • START condition • Error detection • Receive buffer full • Transmit buffer empty 	
14 (0x0e)	TTBR + 0x38	16-bit PWM timer Ch.0 interrupt	<ul style="list-style-type: none"> • Capture overwrite • Compare/capture • Counter MAX • Counter zero 	
15 (0x0f)	TTBR + 0x3c	16-bit PWM timer Ch.1 interrupt	<ul style="list-style-type: none"> • Capture overwrite • Compare/capture • Counter MAX • Counter zero 	
16 (0x10)	TTBR + 0x40	reserved	—	
17 (0x11)	TTBR + 0x44	16-bit timer Ch.2 interrupt	Underflow	
18 (0x12)	TTBR + 0x48	Sound generator interrupt	<ul style="list-style-type: none"> • Sound buffer empty • Sound output completion 	
19 (0x13)	TTBR + 0x4c	LCD driver interrupt	Frame	
20 (0x14)	TTBR + 0x50	R/F converter Ch.0 interrupt	<ul style="list-style-type: none"> • Reference oscillation completion • Sensor A oscillation completion • Sensor B oscillation completion • Measurement counter overflow error • Time base counter overflow error 	
21 (0x15)	TTBR + 0x54	EEPROM controller interrupt	<ul style="list-style-type: none"> • Reprogram/read completion • ECC 	
22 (0x16)	TTBR + 0x58	reserved	—	
:	:	:	:	
31 (0x1f)	TTBR + 0x7c	reserved	—	↓ Low *1

*1 When the same interrupt level is set

*2 Either reset or NMI can be selected as the watchdog timer interrupt with software.

5.2.1 Vector Table Base Address (TTBR)

The MSCTTBRL and MSCTTBRH registers are provided to set the base (start) address of the vector table in which interrupt vectors are programmed. “TTBR” described in Table 5.2.1 means the value set to these registers. After an initial reset, the MSCTTBRL and MSCTTBRH registers are set to address 0x8000. Therefore, even when the vector table location is changed, it is necessary that at least the reset vector be written to the above address. Bits 7 to 0 in the MSCTTBRL register are fixed at 0, so the vector table always begins from a 256-byte boundary address.

5.3 Initialization

The following shows an example of the initial setting procedure related to interrupts:

1. Execute the di instruction to set the CPU into interrupt disabled state.
2. If the vector table start address is different from the default address, set it to the MSCTTBRL and MSCTTBRH registers after removing system protection by writing 0x0096 to the MSCPROT.PROT[15:0] bits. Then, write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits to set system protection.
3. Set the interrupt enable bit of the peripheral circuit to 0 (interrupt disabled).
4. Set the interrupt level for the peripheral circuit using the ITCLVx.ILVx[2:0] bits in the ITC.
5. Configure the peripheral circuit and start its operation.
6. Clear the interrupt factor flag of the peripheral circuit.
7. Set the interrupt enable bit of the peripheral circuit to 1 (interrupt enabled).
8. Execute the ei instruction to set the CPU into interrupt enabled state.

5.4 Maskable Interrupt Control and Operations

5.4.1 Peripheral Circuit Interrupt Control

The peripheral circuit that generates interrupts includes an interrupt enable bit and an interrupt flag for each interrupt cause.

Interrupt flag: The flag is set to 1 when the interrupt cause occurs. The clear condition depends on the peripheral circuit.

Interrupt enable bit: By setting this bit to 1 (interrupt enabled), an interrupt request will be sent to the ITC when the interrupt flag is set to 1. When this bit is set to 0 (interrupt disabled), no interrupt request will be sent to the ITC even if the interrupt flag is set to 1. An interrupt request is also sent to the ITC if the status is changed to interrupt enabled when the interrupt flag is 1.

For specific information on causes of interrupts, interrupt flags, and interrupt enable bits, refer to the respective peripheral circuit descriptions.

Note: To prevent occurrence of unnecessary interrupts, the corresponding interrupt flag should be cleared before setting the interrupt enable bit to 1 (interrupt enabled) and before terminating the interrupt handler routine.

5.4.2 ITC Interrupt Request Processing

On receiving an interrupt signal from a peripheral circuit, the ITC sends an interrupt request, the interrupt level, and the vector number to the CPU. Vector numbers are determined by the ITC internal hardware for each interrupt cause, as shown in Table 5.2.1. The interrupt level is a value to configure the priority, and it can be set to between 0 (low) and 7 (high) using the ITCLVx.ILVx[2:0] bits provided for each interrupt source. The default ITC settings are level 0 for all maskable interrupts. Interrupt requests are not accepted by the CPU if the level is 0.

The ITC outputs the interrupt request with the highest priority to the CPU in accordance with the following conditions if interrupt requests are input to the ITC simultaneously from two or more peripheral circuits.

- The interrupt with the highest interrupt level takes precedence.
- 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 CPU.

If an interrupt cause with higher priority occurs while the ITC is outputting an interrupt request signal to the CPU (before being accepted by the CPU), 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 circuit is cleared via software.

5 Interrupt Controller (ITC)

Note: Before changing the interrupt level, make sure that no interrupt of which the level is changed can be generated (the interrupt enable bit of the peripheral circuit is set to 0 or the peripheral circuit is deactivated).

5.4.3 Conditions to Accept Interrupt Requests by the CPU

The CPU accepts an interrupt request sent from the ITC when all of the following conditions are met:

- The IE (Interrupt Enable) bit of the PSR has been set to 1.
- The interrupt request that has occurred has a higher interrupt level than the value set in the IL[2:0] (Interrupt Level) bits of the PSR.
- No other interrupt request having higher priority, such as NMI, has occurred.

5.5 NMI

The watchdog timer embedded in this IC can generate a non-maskable interrupt (NMI). This interrupt takes precedence over other interrupts and is unconditionally accepted by the CPU.

For detailed information on generating NMI, refer to the “Watchdog Timer” chapter.

5.6 Software Interrupts

The CPU provides the “int *imm5*” and “intl *imm5, imm3*” instructions allowing the software to generate any interrupts. The operand *imm5* specifies a vector number (0–31) in the vector table. In addition to this, the intl instruction has the operand *imm3* to specify the interrupt level (0–7) to be set to the IL[2:0] bits in the PSR. The software interrupt cannot be disabled (non-maskable interrupt). The processor performs the same interrupt processing operation as that of the hardware interrupt.

5.7 Interrupt Processing by the CPU

The CPU samples interrupt requests for each cycle. On accepting an interrupt request, the CPU 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 cleared to 0 (disabling subsequent maskable interrupts).
3. The PSR IL[2:0] 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 accepted, Step 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 the IL[2:0] bits are changed by Step 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 occurred.

The program resumes processing following the instruction being executed at the time the interrupt occurred.

Note: When HALT or SLEEP mode is canceled, the CPU jumps to the interrupt handler routine after executing one instruction. To execute the interrupt handler routine immediately after HALT or SLEEP mode is canceled, place the nop instruction at just behind the halt/slp instruction.

5.8 Control Registers

MISC Vector Table Address Low Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCTTBRL	15–8	TTBR[15:8]	0x80	H0	R/WP	–
	7–0	TTBR[7:0]	0x00	H0	R	

Bits 15–0 TTBR[15:0]

These bits set the vector table base address (16 low-order bits).

MISC Vector Table Address High Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
MSCTTBRH	15–8	–	0x00	–	R	–
	7–0	TTBR[23:16]	0x00	H0	R/WP	

Bits 15–8 Reserved

Bits 7–0 TTBR[23:16]

These bits set the vector table base address (eight high-order bits).

ITC Interrupt Level Setup Register x

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ITCLV _x	15–11	–	0x00	–	R	–
	10–8	ILV _{y₁} [2:0]	0x0	H0	R/W	
	7–3	–	0x00	–	R	
	2–0	ILV _{y₀} [2:0]	0x0	H0	R/W	

Bits 15–11 Reserved

Bits 7–3 Reserved

Bits 10–8 ILV_{y₁}[2:0] ($y_1 = 2x + 1$)

Bits 2–0 ILV_{y₀}[2:0] ($y_0 = 2x$)

These bits set the interrupt level of each interrupt.

Table 5.8.1 Interrupt Level and Priority Settings

ITCLV _x .ILV _y [2:0] bits	Interrupt level	Priority
0x7	7	High
0x6	6	↑
...	...	
0x1	1	↓
0x0	0	Low

The following shows the ITCLV_x register configuration in this IC.

5 Interrupt Controller (ITC)

Table 5.8.2 List of ITCLVx Registers

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ITCLV0 (ITC Interrupt Level Setup Register 0)	15–11	–	0x00	–	R	–
	10–8	ILV1[2:0]	0x0	H0	R/W	Port interrupt (ILVPPORT)
	7–3	–	0x00	–	R	–
	2–0	ILV0[2:0]	0x0	H0	R/W	Supply voltage detector interrupt (ILVSVD5)
ITCLV1 (ITC Interrupt Level Setup Register 1)	15–11	–	0x00	–	R	–
	10–8	ILV3[2:0]	0x0	H0	R/W	Clock generator interrupt (ILVCLG)
	7–0	–	0x00	–	R	–
ITCLV2 (ITC Interrupt Level Setup Register 2)	15–11	–	0x00	–	R	–
	10–8	ILV5[2:0]	0x0	H0	R/W	16-bit timer Ch.0 interrupt (ILVT16_0)
	7–0	–	0x00	–	R	–
ITCLV3 (ITC Interrupt Level Setup Register 3)	15–11	–	0x00	–	R	–
	10–8	ILV7[2:0]	0x0	H0	R/W	16-bit timer Ch.1 interrupt (ILVT16_1)
	7–3	–	0x00	–	R	–
	2–0	ILV6[2:0]	0x0	H0	R/W	UART Ch.0 interrupt (ILVUART3_0)
ITCLV4 (ITC Interrupt Level Setup Register 4)	15–11	–	0x00	–	R	–
	10–8	ILV9[2:0]	0x0	H0	R/W	I ² C interrupt (ILVI2C_0)
	7–3	–	0x00	–	R	–
	2–0	ILV8[2:0]	0x0	H0	R/W	Synchronous serial interface Ch.0 interrupt (ILVSPIA_0)
ITCLV5 (ITC Interrupt Level Setup Register 5)	15–11	–	0x00	–	R	–
	10–8	ILV11[2:0]	0x0	H0	R/W	16-bit PWM timer Ch.1 interrupt (ILVT16B_1)
	7–3	–	0x00	–	R	–
	2–0	ILV10[2:0]	0x0	H0	R/W	16-bit PWM timer Ch.0 interrupt (ILVT16B_0)
ITCLV6 (ITC Interrupt Level Setup Register 6)	15–11	–	0x00	–	R	–
	10–8	ILV13[2:0]	0x0	H0	R/W	16-bit timer Ch.2 interrupt (ILVT16_2)
	7–0	–	0x00	–	R	–
ITCLV7 (ITC Interrupt Level Setup Register 7)	15–11	–	0x00	–	R	–
	10–8	ILV15[2:0]	0x0	H0	R/W	LCD driver interrupt (ILVLCD4B)
	7–3	–	0x00	–	R	–
	2–0	ILV14[2:0]	0x0	H0	R/W	Sound generator interrupt (ILVSND2)
ITCLV8 (ITC Interrupt Level Setup Register 8)	15–11	–	0x00	–	R	–
	10–8	ILV17[2:0]	0x0	H0	R/W	EEPROM controller interrupt (ILVEPRC)
	7–3	–	0x00	–	R	–
	2–0	ILV16[2:0]	0x0	H0	R/W	R/F converter Ch.0 interrupt (ILVRFC2_0)

6 I/O Ports (PPORT)

6.1 Overview

PPORT controls the I/O ports. The main features are outlined below.

- Allows port-by-port function configurations.
 - Each port can be configured with or without a pull-up or pull-down resistor.
 - Each port can be configured with or without a chattering filter.
 - Allows selection of the function (general-purpose I/O port (GPIO) function, up to four peripheral I/O functions) to be assigned to each port.
- Equipped with Nch open-drain LED drive output pins capable of directly driving LEDs.
- Ports, except for those shared with debug pins, are initially placed into Hi-Z state. (No current passes through the pin during this Hi-Z state.)

Note: 'x', which is used in the port names Pxy, register names, and bit names, refers to a port group (x = 0, 1, 2, d) and 'y' refers to a port number (y = 0, 1, 2, ..., 7).

Figure 6.1.1 shows the configuration of PPORT.

Table 6.1.1 Port Configuration of S1C17W11

Item	S1C17W11	
Port groups included	P0	P0[7:0] (8 ports) *1,*2
	P1	P1[7:0] (8 ports) *1,*2
	P2	P2[5:0] (6 ports) *1,*2
	Pd	Pd[2:0] (Pd2: output only) (3 ports) *1
Total number of ports	Input/output port: 24 Output port: 1	
LED drive output pins	P0[3:2]	
Ports for debug function	Pd[2:0]	
Key-entry reset function	Supported (P1[3:0])	

*1 Ports with general-purpose I/O function (GPIO)

*2 Ports with interrupt function

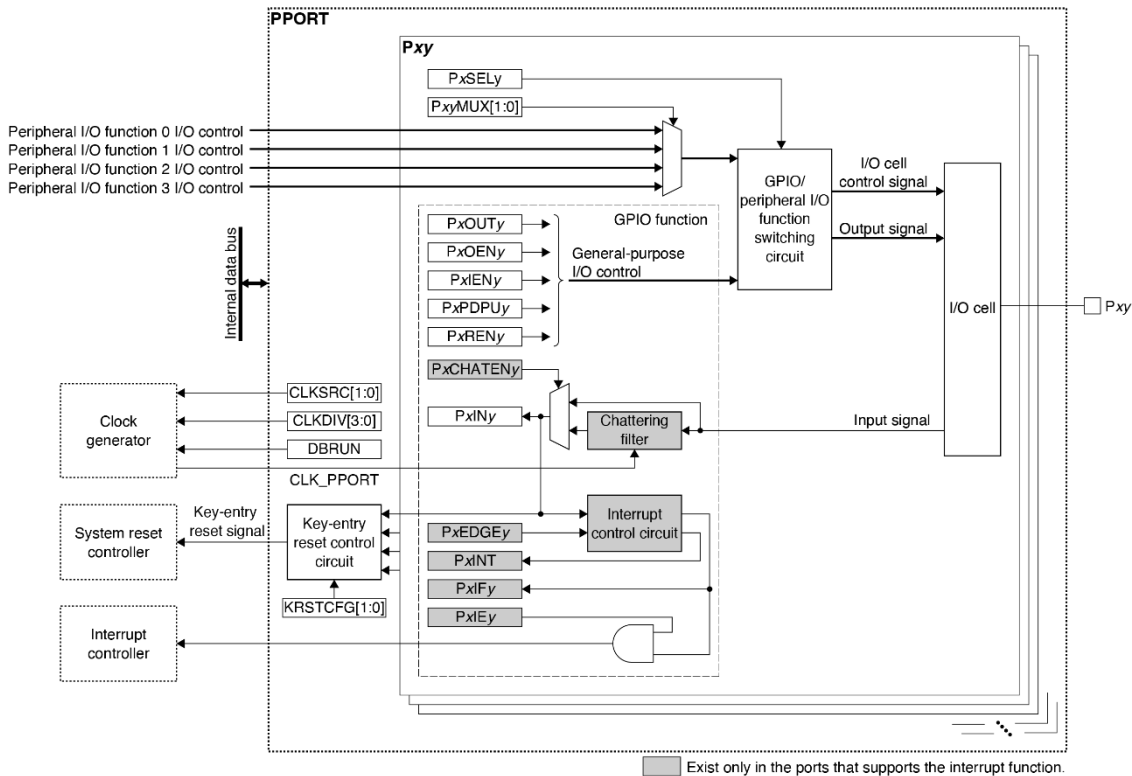


Figure 6.1.1 PPORT Configuration

6.2 I/O Cell Structure and Functions

Figure 6.2.1 shows the I/O cell Configuration.

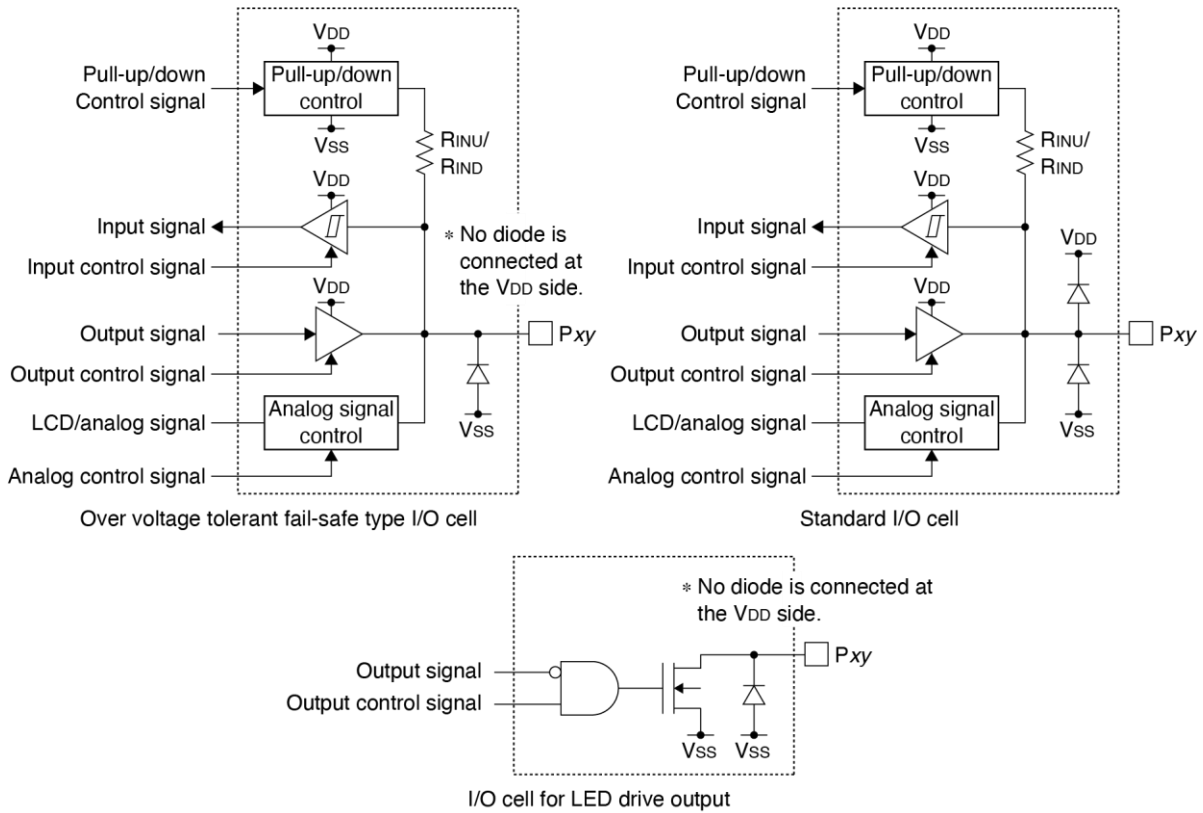


Figure 6.2.1 I/O Cell Configuration

Refer to “Pin Descriptions” in the “Overview” chapter for the cell type, either the over voltage tolerant fail-safe type I/O cell or the standard I/O cell, included in each port.

6.2.1 Schmitt Input

The input functions are all configured with the Schmitt interface level. When a port is set to input disable status ($PxIOEN.PxIENy$ bit = 0), unnecessary current is not consumed if the Pxy pin is placed into floating status.

6.2.2 Over Voltage Tolerant Fail-Safe Type I/O Cell

The over voltage tolerant fail-safe type I/O cell allows interfacing without passing unnecessary current even if a voltage exceeding V_{DD} is applied to the port. Also unnecessary current is not consumed when the port is externally biased without supplying V_{DD} . However, be sure to avoid applying a voltage exceeding the recommended maximum operating power supply voltage to the port.

6.2.3 Pull-Up/Pull-Down

The GPIO port has a pull-up/pull-down function. Either pull-up or pull-down may be selected for each port individually. This function may also be disabled for the port that does not require pulling up/down.

When the port level is switched from low to high through the pull-up resistor included in the I/O cell or from high to low through the pull-down resistor, a delay will occur in the waveform rising/falling edge depending on the time constant by the pull-up/pull-down resistance and the pin load capacitance. The rising/falling time is commonly determined by the following equation:

$$t_{PR} = -R_{INU} \times (C_{IN} + C_{BOARD}) \times \ln(1 - V_{T+}/V_{DD})$$

$$t_{PF} = -R_{IND} \times (C_{IN} + C_{BOARD}) \times \ln(1 - V_{T-}/V_{DD})$$
(Eq. 6.1)

Where

t_{PR} :	Rising time (port level = low \rightarrow high) [second]
t_{PF} :	Falling time (port level = high \rightarrow low) [second]
V_{T+} :	High level Schmitt input threshold voltage [V]
V_{T-} :	Low level Schmitt input threshold voltage [V]
R_{INU}/R_{IND} :	Pull-up/pull-down resistance [Ω]
C_{IN} :	Pin capacitance [F]
C_{BOARD} :	Parasitic capacitance on the board [F]

6.2.4 CMOS Output and High Impedance State

The I/O cells except for analog output can output signals in the V_{DD} and V_{SS} levels. Also the GPIO ports may be put into high-impedance (Hi-Z) state.

6.2.5 LED Drive Output Pins

The I/O cells of the LED drive output pins have an Nch open-drain structure, enabling direct LED driving. While these pins support input/output functionality, they do not support pull-up or pull-down control.

6.3 Clock Settings

6.3.1 PPORT Operating Clock

When using the chattering filter for entering external signals to PPORT, the PPORT operating clock CLK_PPORT must be supplied to PPORT from the clock generator.

The CLK_PPORT supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
2. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
3. Set the following PCLK register bits:
 - PCLK.CLKSRC[1:0] bits (Clock source selection)
 - PCLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Settings in Step 3 determine the input sampling time of the chattering filter.

6.3.2 Clock Supply in SLEEP Mode

When using the chattering filter function during SLEEP mode, the PPORT operating clock CLK_PPORT must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_PPORT clock source. If the CLGOSC.xxxxSLPC bit for the CLK_PPORT clock source is 1, the CLK_PPORT clock source is deactivated during SLEEP mode and it disables the chattering filter function regardless of the PxCHATEN.PxCHATENy bit setting (chattering filter enabled/disabled).

6.3.3 Clock Supply in DEBUG Mode

The CLK_PPORT supply during DEBUG mode should be controlled using the PCLK.DBRUN bit.

The CLK_PPORT supply to PPORT is suspended when the CPU enters DEBUG mode if the PCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_PPORT supply resumes. The PPORT chattering filter stops operating when the CLK_PPORT supply is suspended. If the chattering filter is enabled in PPORT, the input port function is also deactivated. However, the control registers can be altered. If the PCLK.DBRUN bit = 1, the CLK_PPORT supply is not suspended and the chattering filter will keep operating in DEBUG mode.

6.4 Operations

6.4.1 Initialization

After a reset, the ports except for the debugging function are configured as shown below.

- Port input: Disabled
- Port output: Disabled
- Pull-up: Off
- Pull-down: Off
- Port pins: High impedance state
- Port function: Configured to GPIO

This status continues until the ports are configured via software. The debugging function ports are configured for debug signal input/output.

Initial settings when using a port for a peripheral I/O function

When using the P_{xy} port for a peripheral I/O function, perform the following software initial settings:

1. Set the following PxIOEN register bits:
 - Set the PxIOEN.PxIEN_y bit to 0. (Disable input)
 - Set the PxIOEN.PxOEN_y bit to 0. (Disable output)
2. Set the PxMODSEL.PxSEL_y bit to 0. (Disable peripheral I/O function)
3. Initialize the peripheral circuit that uses the pin.
4. Set the PxFNCSEL.PxMUX[1:0] bits. (Select peripheral I/O function)
5. Set the PxMODSEL.PxSEL_y bit to 1. (Enable peripheral I/O function)

For the list of the peripheral I/O functions that can be assigned to each port of this IC, refer to “Control Register and Port Function Configuration of this IC.” For the specific information on the peripheral I/O functions, refer to the respective peripheral circuit chapter.

Initial settings when using a port as a general-purpose output port (only for the ports with GPIO function)

When using the P_{xy} port pin as a general-purpose output pin, perform the following software initial settings:

1. Set the PxIOEN.PxOEN_y bit to 1. (Enable output)
2. Set the PxMODSEL.PxSEL_y bit to 0. (Enable GPIO function)

Initial settings when using a port as a general-purpose input port (only for the ports with GPIO function)

When using the P_{xy} port pin as a general-purpose input pin, perform the following software initial settings:

1. Write 0 to the PxINTCTL.PxIE_y bit. * (Disable interrupt)
2. When using the chattering filter, configure the PPORT operating clock (see “PPORT Operating Clock”) and set the PxCHATEN.PxCHATEN_y bit to 1. *

When the chattering filter is not used, set the PxCHATEN.PxCHATEN_y bit to 0 (supply of the PPORT operating clock is not required).

3. Configure the following PxRCTL register bits when pulling up/down the port using the internal pull-up or down resistor:
 - PxRCTL.PxPDU_y bit (Select pull-up or pull-down resistor)
 - Set the PxRCTL.PxREN_y bit to 1. (Enable pull-up/down)

Set the PxRCTL.PxREN_y bit to 0 if the internal pull-up/down resistors are not used.

4. Set the PxMODSEL.PxSEL_y bit to 0. (Enable GPIO function)

5. Configure the following bits when using the port input interrupt: *
 - Write 1 to the PxINTF.PxIFy bit. (Clear interrupt flag)
 - PxINTCTL.PxEDGEy bit (Select interrupt edge (input rising edge/falling edge))
 - Set the PxINTCTL.PxIEy bit to 1. (Enable interrupt)
 6. Set the following PxIOEN register bits:
 - Set the PxIOEN.PxOENy bit to 0. (Disable output)
 - Set the PxIOEN.PxIENy bit to 1. (Enable input)
- * Steps 1 and 5 are required for the ports with an interrupt function. Step 2 is required for the ports with a chattering filter function.

Table 6.4.1.1 lists the port status according to the combination of data input/output control and pull-up/down control.

Table 6.4.1.1 GPIO Port Control List

PxIOEN. PxIENy bit	PxIOEN. PxOENy bit	PxRCTL. PxRENy bit	PxRCTL. PxPDUy bit	Input	Output	Pull-up/pull-down condition
0	0	0	x	Disabled		Off (Hi-Z) *1
0	0	1	0	Disabled		Pulled down
0	0	1	1	Disabled		Pulled up
1	0	0	x	Enabled	Disabled	Off (Hi-Z) *2
1	0	1	0	Enabled	Disabled	Pulled down
1	0	1	1	Enabled	Disabled	Pulled up
0	1	0	x	Disabled	Enabled	Off
0	1	1	0	Disabled	Enabled	Off
0	1	1	1	Disabled	Enabled	Off
1	1	1	0	Enabled	Enabled	Off
1	1	1	1	Enabled	Enabled	Off

*1: Initial status. Current does not flow if the pin is placed into floating status.

*2: Use of the pull-up or pull-down function is recommended, as undesired current will flow if the port input is set to floating status.

Note: If the PxMODESEL.PxSELy bit for the port without a GPIO function is set to 0, the port goes into initial status (refer to "Initial Settings"). The GPIO control bits are configured to a read-only bit always read out as 0.

6.4.2 Port Input/Output Control

Peripheral I/O function control

The port for which a peripheral I/O function is selected is controlled by the peripheral circuit. For more information, refer to the respective peripheral circuit chapter.

Setting output data to a GPIO port

Write data (1 = high output, 0 = low output) to be output from the Pxy pin to the PxDAT.PxOUTy bit.

Reading input data from a GPIO port

The data (1 = high input, 0 = low input) input from the Pxy pin can be read out from the PxDAT.PxINy bit.

Chattering filter function

Some ports have a chattering filter function and it can be controlled in each port. This function is enabled by setting the PxCHATEN.PxCHATENy bit to 1. The input sampling time to remove chattering is determined by the CLK_PPORF frequency configured using the PCLK register in common to all ports. The chattering filter removes pulses with a shorter width than the input sampling time.

$$\text{Input sampling time} = \frac{2 \text{ to } 3}{\text{CLK_PPORF frequency [Hz]}} \text{ [second]} \quad (\text{Eq. 6.2})$$

Make sure the Pxy port interrupt is disabled before altering the PCLK register and PxCHATEN.PxCHATENy bit settings. A Pxy port interrupt may erroneously occur if these settings are altered in an interrupt enabled status. Furthermore, enable the interrupt after a lapse of four or more CLK_PPORF cycles from enabling the chattering filter function.

6 I/O Ports (PPORT)

If the clock generator is configured so that it will supply CLK_PPORT to PPORT in SLEEP mode, the chattering filter of the port will function even in SLEEP mode. If CLK_PPORT is configured to stop in SLEEP mode, PPORT inactivates the chattering filter during SLEEP mode to input pin status transitions directly to itself.

Key-entry reset function

This function issues a reset request when low-level pulses are input to all the specified ports simultaneously. Make the following settings when using this function:

1. Configure the ports to be used for key-entry reset as general-purpose input ports (refer to “Initial settings when using a port as a general-purpose input port (only for the ports with GPIO function)”).
2. Configure the input pin combination for key-entry reset using the PCLK.KRSTCFG[1:0] bits.

Note: When enabling the key-entry reset function, be sure to configure the port pins to be used for it as general-purpose input pins before setting the PCLK.KRSTCFG[1:0] bits.

PPORT issues a reset request immediately after all the input pins specified by the PCLK.KRSTCFG[1:0] are set to a low level if the chattering filter function is disabled (initial status). To issue a reset request only when low-level signals longer than the time configured are input, enable the chattering filter function for all the ports used for key-entry reset.

The pins configured for key-entry reset can also be used as general-purpose input pins.

6.5 Interrupts

When the GPIO function is selected for the port with an interrupt function, the port input interrupt function can be used.

Table 6.5.1 Port Input Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Port input interrupt	PxINTF.PxIFy	Rising or falling edge of the input signal	Writing 1
	PINTFGRP.PxINT	Setting an interrupt flag in the port group	Clearing PxINTF.PxIFy

Interrupt edge selection

Port input interrupts will occur at the falling edge of the input signal when setting the PxINTCTL.PxEDGEy bit to 1, or the rising edge when setting to 0.

Interrupt enable

PPORT provides interrupt enable bits (PxINTCTL.PxIEy bit) corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

Interrupt check in port group unit

When interrupts are enabled in two or more port groups, check the PINTFGRP.PxINT bit in the interrupt handler first. It helps minimize the handler codes for finding the port that has generated an interrupt. If this bit is set to 1, an interrupt has occurred in the port group. Next, check the PxINTF.PxIFy bit set to 1 in the port group to determine the port that has generated an interrupt. Clearing the PxINTF.PxIFy bit also clears the PINTFGRP.PxINT bit. If the port is set to interrupt disabled status by the PxINTCTL.PxIEy bit, the PINTFGRP.PxINT bit will not be set even if the PxINTF.PxIFy bit is set to 1.

6.6 Control Registers

This section describes the same control registers of all port groups as a single register. For the register and bit configurations in each port group and their initial values, refer to “Control Register and Port Function Configuration of this IC.”

Px Port Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxDAT	15–8	PxOUT[7:0]	0x00	H0	R/W	–
	7–0	PxIN[7:0]	0x00	H0	R	

*1: This register is effective when the GPIO function is selected.

*2: The bit configuration differs depending on the port group.

*3: The initial value may be changed by the port.

Bits 15–8 PxOUT[7:0]

These bits are used to set data to be output from the GPIO port pins.

1 (R/W): Output high level from the port pin

0 (R/W): Output low level from the port pin

When output is enabled (PxIOEN.PxOEN_y bit = 1), the port pin outputs the data set here. Although data can be written when output is disabled (PxIOEN.PxOEN_y bit = 0), it does not affect the pin status.

These bits do not affect the outputs when the port is used as a peripheral I/O function.

Bits 7–0 PxIN[7:0]

The GPIO port pin status can be read out from these bits.

1 (R): Port pin = High level

0 (R): Port pin = Low level

The port pin status can be read out when input is enabled (PxIOEN.PxIEN_y bit = 1). When input is disabled (PxIOEN.PxIEN_y bit = 0), these bits are always read as 0.

When the port is used for a peripheral I/O function, the input value cannot be read out from these bits.

Px Port Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxIOEN	15–8	PxIEN[7:0]	0x00	H0	R/W	–
	7–0	PxOEN[7:0]	0x00	H0	R/W	

*1: This register is effective when the GPIO function is selected.

*2: The bit configuration differs depending on the port group.

Bits 15–8 PxIEN[7:0]

These bits enable/disable the GPIO port input.

1 (R/W): Enable (The port pin status is input.)

0 (R/W): Disable (Input data is fixed at 0.)

When both data output and data input are enabled, the pin output status controlled by this IC can be read.

These bits do not affect the input control when the port is used as a peripheral I/O function.

Bits 7–0 PxOEN[7:0]

These bits enable/disable the GPIO port output.

1 (R/W): Enable (Data is output from the port pin.)

0 (R/W): Disable (The port is placed into Hi-Z.)

These bits do not affect the output control when the port is used as a peripheral I/O function.

Px Port Pull-up/down Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxRCTL	15–8	PxPDPJ[7:0]	0x00	H0	R/W	–
	7–0	PxREN[7:0]	0x00	H0	R/W	

*1: This register is effective when the GPIO function is selected.

*2: The bit configuration differs depending on the port group.

Bits 15–8 PxPDPJ[7:0]

These bits select either the pull-up resistor or the pull-down resistor when using a resistor built into the port.

1 (R/W): Pull-up resistor

0 (R/W): Pull-down resistor

The selected pull-up/down resistor is enabled when the PxRCTL.PxREN_y bit = 1.

Bits 7–0 PxREN[7:0]

These bits enable/disable the port pull-up/down control.

1 (R/W): Enable (The built-in pull-up/down resistor is used.)

0 (R/W): Disable (No pull-up/down control is performed.)

Enabling this function pulls up or down the port when output is disabled (PxIOEN.PxOEN_y bit = 0).

When output is enabled (PxIOEN.PxOEN_y bit = 1), the PxRCTL.PxREN_y bit setting is ineffective regardless of how the PxIOEN.PxIEN_y bit is set and the port is not pulled up/down.

These bits do not affect the pull-up/down control when the port is used as a peripheral I/O function.

6 I/O Ports (PPORT)

Px Port Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxINTF	15–8	–	0x00	–	R	–
	7–0	PxIF[7:0]	0x00	H0	R/W	Cleared by writing 1.

*1: This register is effective when the GPIO function is selected.

*2: The bit configuration differs depending on the port group.

Bits 15–8 Reserved

Bits 7–0 PxIF[7:0]

These bits indicate the port input interrupt cause occurrence status.

- 1 (R): Cause of interrupt occurred
- 0 (R): No cause of interrupt occurred
- 1 (W): Clear flag
- 0 (W): Ineffective

Px Port Interrupt Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxINTCTL	15–8	PxEDGE[7:0]	0x00	H0	R/W	–
	7–0	PxIE[7:0]	0x00	H0	R/W	–

*1: This register is effective when the GPIO function is selected.

*2: The bit configuration differs depending on the port group.

Bits 15–8 PxEDGE[7:0]

These bits select the input signal edge to generate a port input interrupt.

- 1 (R/W): An interrupt will occur at a falling edge.
- 0 (R/W): An interrupt will occur at a rising edge.

Bits 7–0 PxIE[7:0]

These bits enable port input interrupts.

- 1 (R/W): Enable interrupts
- 0 (R/W): Disable interrupts

Note: To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

Px Port Chattering Filter Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxCHATEN	15–8	–	0x00	–	R	–
	7–0	PxCHATEN[7:0]	0x00	H0	R/W	–

*1: The bit configuration differs depending on the port group.

Bits 15–8 Reserved

Bits 7–0 PxCHATEN[7:0]

These bits enable/disable the chattering filter function.

- 1 (R/W): Enable (The chattering filter is used.)
- 0 (R/W): Disable (The chattering filter is bypassed.)

Px Port Mode Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxMODSEL	15–8	–	0x00	–	R	–
	7–0	PxSEL[7:0]	0x00	H0	R/W	–

*1: The bit configuration differs depending on the port group.

*2: The initial value may be changed by the port.

Bits 15–8 Reserved

Bits 7–0 PxSEL[7:0]

These bits select whether each port is used for the GPIO function or a peripheral I/O function.

- 1 (R/W): Use peripheral I/O function
- 0 (R/W): Use GPIO function

Px Port Function Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxFNCSEL	15–14	Px7MUX[1:0]	0x0	H0	R/W	–
	13–12	Px6MUX[1:0]	0x0	H0	R/W	
	11–10	Px5MUX[1:0]	0x0	H0	R/W	
	9–8	Px4MUX[1:0]	0x0	H0	R/W	
	7–6	Px3MUX[1:0]	0x0	H0	R/W	
	5–4	Px2MUX[1:0]	0x0	H0	R/W	
	3–2	Px1MUX[1:0]	0x0	H0	R/W	
	1–0	Px0MUX[1:0]	0x0	H0	R/W	

*1: The bit configuration differs depending on the port group.

*2: The initial value may be changed by the port.

Bits 15–14 Px7MUX[1:0]

⋮

Bits 1–0 Px0MUX[1:0]

These bits select the peripheral I/O function to be assigned to each port pin.

Table 6.6.1 Selecting Peripheral I/O Function

PxFNCSEL.PxyMUX[1:0] bits	Peripheral I/O function
0x3	Function 3
0x2	Function 2
0x1	Function 1
0x0	Function 0

This selection takes effect when the PxMODSEL.PxSELY bit = 1.

P Port Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/WP	
	7–4	CLKDIV[3:0]	0x0	H0	R/WP	
	3–2	KRSTCFG[1:0]	0x0	H0	R/WP	
	1–0	CLKSRC[1:0]	0x0	H0	R/WP	

Bits 15–9 Reserved**Bit 8 DBRUN**

This bit sets whether the PPORT operating clock is supplied in DEBUG mode or not.

1 (R/WP): Clock supplied in DEBUG mode

0 (R/WP): No clock supplied in DEBUG mode

Bits 7–4 CLKDIV[3:0]

These bits select the division ratio of the PPORT operating clock (chattering filter clock).

Bits 3–2 KRSTCFG[1:0]

These bits configure the key-entry reset function.

Table 6.6.2 Key-Entry Reset Function Settings

PCLK.KRSTCFG[1:0] bits	key-entry reset
0x3	Reset when P1[3:0] inputs = all low
0x2	Reset when P1[2:0] inputs = all low
0x1	Reset when P1[1:0] inputs = all low
0x0	Disable

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of PPORT (chattering filter).

The PPORT operating clock should be configured by selecting the clock source using the PCLK.CLKSRC[1:0] bits and the clock division ratio using the PCLK.CLKDIV[3:0] bits as shown in Table 6.6.3. These settings determine the input sampling time of the chattering filter.

6 I/O Ports (PPORT)

Table 6.6.3 Clock Source and Division Ratio Settings

PCLK.CLKDIV[3:0] bits	PCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0xf	1/32,768			1/1
0xe	1/16,384			
0xd	1/8,192			
0xc	1/4,096			
0xb	1/2,048			
0xa	1/1,024			
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			

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

P Port Interrupt Flag Group Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PINTFGRP	15–13	–	0x0	–	R	–
	12	PcINT	0	H0	R	
	11	PbINT	0	H0	R	
	10	PaINT	0	H0	R	
	9	P9INT	0	H0	R	
	8	P8INT	0	H0	R	
	7	P7INT	0	H0	R	
	6	P6INT	0	H0	R	
	5	P5INT	0	H0	R	
	4	P4INT	0	H0	R	
	3	P3INT	0	H0	R	
	2	P2INT	0	H0	R	
	1	P1INT	0	H0	R	
	0	P0INT	0	H0	R	

*1: Only the bits corresponding to the port groups that support interrupts are provided.

Bits 15–13 Reserved

Bits 12–0 PxINT

These bits indicate that Px port group includes a port that has generated an interrupt.

1 (R): A port generated an interrupt

0 (R): No port generated an interrupt

The PINTFGRP.PxINT bit is cleared when the interrupt flag for the port that has generated an interrupt is cleared.

6.7 Control Register and Port Function Configuration of this IC

This section shows the PPORT control register/bit configuration in this IC and the list of peripheral I/O functions selectable for each port.

Note: The control bits for the ports that are not available in the model are reserved bits. Do not alter them from the initial value.

6.7.1 P0 Port Group

The P0 port group supports the GPIO and interrupt functions.

Table 6.7.1.1 Control Registers for P0 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P0DAT (P0 Port Data Register)	15–8	P0OUT[7:0]	0x00	H0	R/W	–
	7–0	P0IN[7:0]	0x00	H0	R	
P0IOEN (P0 Port Enable Register)	15–8	P0IEN[7:0]	0x00	H0	R/W	–
	7–0	P0OEN[7:0]	0x00	H0	R/W	
P0RCTL (P0 Port Pull-up/down Control Register)	15–12	P0PDU[7:4]	0x0	H0	R/W	–
	11–10	–	0x0	–	R	
	9–8	P0PDU[1:0]	0x0	H0	R/W	
	7–4	P0REN[7:4]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	P0REN[1:0]	0x0	H0	R/W	
P0INTF (P0 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–
	7–0	P0IF[7:0]	0x00	H0	R/W	
P0INTCTL (P0 Port Interrupt Control Register)	15–8	P0EDGE[7:0]	0x00	H0	R/W	–
	7–0	P0IE[7:0]	0x00	H0	R/W	
P0CHATEN (P0 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–4	P0CHATEN[7:4]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	P0CHATEN[1:0]	0x0	H0	R/W	
P0MODSEL (P0 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P0SEL[7:0]	0x00	H0	R/W	
P0FNCSEL (P0 Port Function Select Register)	15–14	P07MUX[1:0]	0x0	H0	R/W	–
	13–12	P06MUX[1:0]	0x0	H0	R/W	
	11–10	P05MUX[1:0]	0x0	H0	R/W	
	9–8	P04MUX[1:0]	0x0	H0	R/W	
	7–6	P03MUX[1:0]	0x0	H0	R/W	
	5–4	P02MUX[1:0]	0x0	H0	R/W	
	3–2	P01MUX[1:0]	0x0	H0	R/W	
	1–0	P00MUX[1:0]	0x0	H0	R/W	

Table 6.7.1.2 P0 Port Group Function Assignment

Port name	P0SELY = 0		P0SELY = 1						
	GPIO	P0yMUX = 0x0 (Function 0)		P0yMUX = 0x1 (Function 1)		P0yMUX = 0x2 (Function 2)		P0yMUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P00	P00	SNDA2	BZOUT	UPMUX	*1	–	–	–	–
P01	P01	SNDA2	#BZOUT	UPMUX	*1	–	–	–	–
P02	P02	–	–	UPMUX	*1	–	–	–	–
P03	P03	–	–	UPMUX	*1	–	–	–	–
P04	P04	RFC2 Ch.0	SENB0	UPMUX	*1	–	–	–	–
P05	P05	RFC2 Ch.0	SENA0	UPMUX	*1	–	–	–	–
P06	P06	RFC2 Ch.0	REF0	UPMUX	*1	–	–	–	–
P07	P07	RFC2 Ch.0	RFCLK00	UPMUX	*1	–	–	–	–

*1: Refer to the “Universal Port Multiplexer” chapter.

6 I/O Ports (PPORT)

6.7.2 P1 Port Group

The P1 port group supports the GPIO and interrupt functions.

Table 6.7.2.1 Control Registers for P1 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P1DAT (P1 Port Data Register)	15–8	P1OUT[7:0]	0x00	H0	R/W	–
	7–0	P1IN[7:0]	0x00	H0	R	
P1IOEN (P1 Port Enable Register)	15–8	P1IEN[7:0]	0x00	H0	R/W	–
	7–0	P1OEN[7:0]	0x00	H0	R/W	
P1RCTL (P1 Port Pull-up/down Control Register)	15–8	P1PDU[7:0]	0x00	H0	R/W	–
	7–0	P1REN[7:0]	0x00	H0	R/W	
P1INTF (P1 Port Interrupt Flag Register)	15–8	–	0x00	–	R	Cleared by writing 1.
	7–0	P1IF[7:0]	0x00	H0	R/W	
P1INTCTL (P1 Port Interrupt Control Register)	15–8	P1EDGE[7:0]	0x00	H0	R/W	–
	7–0	P1IE[7:0]	0x00	H0	R/W	
P1CHATEN (P1 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–0	P1CHATEN[7:0]	0x00	H0	R/W	
P1MODSEL (P1 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P1SEL[7:0]	0x00	H0	R/W	
P1FNCSEL (P1 Port Function Select Register)	15–14	P17MUX[1:0]	0x0	H0	R/W	–
	13–12	P16MUX[1:0]	0x0	H0	R/W	
	11–10	P15MUX[1:0]	0x0	H0	R/W	
	9–8	P14MUX[1:0]	0x0	H0	R/W	
	7–6	P13MUX[1:0]	0x0	H0	R/W	
	5–4	P12MUX[1:0]	0x0	H0	R/W	
	3–2	P11MUX[1:0]	0x0	H0	R/W	
1–0	P10MUX[1:0]	0x0	H0	R/W		

Table 6.7.2.2 P1 Port Group Function Assignment

Port name	P1SELY = 0		P1SELY = 1						
	GPIO	P1yMUX = 0x0 (Function 0)		P1yMUX = 0x1 (Function 1)		P1yMUX = 0x2 (Function 2)		P1yMUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P10	P10	LCD4B	LFRO	UPMUX	*1	–	–	LCD4B	SEG19
P11	P11	CLG	FOUT	UPMUX	*1	–	–	LCD4B	SEG18
P12	P12	T16B Ch.0	EXCL00	UPMUX	*1	–	–	LCD4B	SEG17
P13	P13	T16B Ch.0	EXCL01	UPMUX	*1	–	–	LCD4B	SEG16
P14	P14	T16B Ch.1	EXCL10	UPMUX	*1	–	–	LCD4B	SEG15
P15	P15	T16B Ch.1	EXCL11	UPMUX	*1	–	–	LCD4B	SEG14
P16	P16	CLG	EXOSC	UPMUX	*1	–	–	LCD4B	SEG13
P17	P17	–	–	UPMUX	*1	–	–	LCD4B	SEG12

*1: Refer to the “Universal Port Multiplexer” chapter.

6.7.3 P2 Port Group

The P2 port group supports the GPIO and interrupt functions.

Table 6.7.3.1 Control Registers for P2 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
P2DAT (P2 Port Data Register)	15–14	–	0x0	–	R	–
	13–8	P2OUT[5:0]	0x00	H0	R/W	
	7–6	–	0x0	–	R	
	5–0	P2IN[5:0]	0x00	H0	R	
P2IOEN (P2 Port Enable Register)	15–14	–	0x0	–	R	–
	13–8	P2IEN[5:0]	0x00	H0	R/W	
	7–6	–	0x0	–	R	
	5–0	P2OEN[5:0]	0x00	H0	R/W	
P2RCTL (P2 Port Pull-up/down Control Register)	15–14	–	0x0	–	R	–
	13–8	P2DPU[5:0]	0x00	H0	R/W	
	7–6	–	0x0	–	R	
	5–0	P2REN[5:0]	0x00	H0	R/W	
P2INTF (P2 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–
	7–6	–	0x0	–	R	
	5–0	P2IF[5:0]	0x00	H0	R/W	
P2INTCTL (P2 Port Interrupt Control Register)	15–14	–	0x0	–	R	–
	13–8	P2EDGE[5:0]	0x00	H0	R/W	
	7–6	–	0x0	–	R	
	5–0	P2IE[5:0]	0x00	H0	R/W	
P2CHATEN (P2 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–6	–	0x0	–	R	
	5–0	P2CHATEN[5:0]	0x00	H0	R/W	
P2MODSEL (P2 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–6	–	0x0	–	R	
	5–0	P2SEL[5:0]	0x00	H0	R/W	
P2FNCSEL (P2 Port Function Select Register)	15–12	–	0x0	–	R	–
	11–10	P25MUX[1:0]	0x0	H0	R/W	
	9–8	P24MUX[1:0]	0x0	H0	R/W	
	7–6	P23MUX[1:0]	0x0	H0	R/W	
	5–4	P22MUX[1:0]	0x0	H0	R/W	
	3–2	P21MUX[1:0]	0x0	H0	R/W	
	1–0	P20MUX[1:0]	0x0	H0	R/W	

Table 6.7.3.2 P2 Port Group Function Assignment

Port name	GPIO	P2SELY = 1							
		P2yMUX = 0x0 (Function 0)		P2yMUX = 0x1 (Function 1)		P2yMUX = 0x2 (Function 2)		P2yMUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P20	P20	SVD5	EXSVD	–	–	–	–	LCD4B	SEG11
P21	P21	–	–	–	–	–	–	LCD4B	SEG10
P22	P22	–	–	–	–	–	–	LCD4B	SEG9
P23	P23	–	–	–	–	–	–	LCD4B	SEG8
P24	P24	LCD4B	CP2	–	–	–	–	LCD4B	SEG3
P25	P25	LCD4B	CP1	–	–	–	–	LCD4B	SEG2

6 I/O Ports (PPORT)

6.7.4 Pd Port Group

The Pd0–Pd2 ports are configured as a debugging function port at initialization. The Pd port group supports the GPIO functions. The GPIO function of the Pd2 port supports output only, therefore, the pull-up/down function cannot be used.

Table 6.7.4.1 Control Registers for Pd Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PDDAT (Pd Port Data Register)	15–11	–	0x00	–	R	–
	10	PDOOUT2	0	H0	R/W	
	9	PDOOUT1	0	H0	R/W	
	8	PDOOUT0	0	H0	R/W	
	7–3	–	0x00	–	R	
	2	–	0	–	R	
	1	PDIN1	X	H0	R	
PDIOEN (Pd Port Enable Register)	15–11	–	0x00	–	R	–
	10	(reserved)	0	H0	R/W	
	9	PDIEN1	0	H0	R/W	
	8	PDIEN0	0	H0	R/W	
	7–3	–	0x00	–	R	
	2	PDOEN2	0	H0	R/W	
	1	PDOEN1	0	H0	R/W	
PDRCTL (Pd Port Pull-up/down Control Register)	15–11	–	0x00	–	R	–
	10	(reserved)	0	H0	R/W	
	9	PDPDPU1	0	H0	R/W	
	8	PDPDPU0	0	H0	R/W	
	7–5	–	0x00	–	R	
	2	(reserved)	0	H0	R/W	
	1	PDREN1	0	H0	R/W	
PDINTF PDINTCTL PDCHATEN	15–0	–	0x0000	–	R	–
	0	PDREN0	0	H0	R/W	
PDMODSEL (Pd Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–3	–	0	–	R	
	2	PDSEL2	1	H0	R/W	
	1	PDSEL1	1	H0	R/W	
PDFNCSEL (Pd Port Function Select Register)	7–6	–	0x0	–	R	–
	5–4	PD2MUX[1:0]	0x0	H0	R/W	
	3–2	PD1MUX[1:0]	0x0	H0	R/W	
	1–0	PD0MUX[1:0]	0x0	H0	R/W	
	0	PDSELO	1	H0	R/W	

Table 6.7.4.2 Pd Port Group Function Assignment

Port name	PDSELY = 0		PDSELY = 1						
	GPIO	PDyMUX = 0x0 (Function 0)		PDyMUX = 0x1 (Function 1)		PDyMUX = 0x2 (Function 2)		PDyMUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
Pd0	PD0	DBG	DST2	–	–	–	–	–	–
Pd1	PD1	DBG	DSIO	–	–	–	–	–	–
Pd2	PD2	DBG	DCLK	–	–	–	–	–	–

6.7.5 Common Registers between Port Groups

Table 6.7.5.1 Control Registers for Common Use with Port Groups

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PCLK (P Port Clock Control Register)	15-9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/WP	
	7-4	CLKDIV[3:0]	0x0	H0	R/WP	
	3-2	KRSTCFG[1:0]	0x0	H0	R/WP	
	1-0	CLKSRC[1:0]	0x0	H0	R/WP	
PINTGRP (P Port Interrupt Flag Group Register)	15-8	–	0x00	–	R	–
	7-3	–	0x00	–	R	
	2	P2INT	0	H0	R	
	1	P1INT	0	H0	R	
	0	P0INT	0	H0	R	

7 Universal Port Multiplexer (UPMUX)

7.1 Overview

UPMUX is a multiplexer that allows software to assign the desired peripheral I/O function to an I/O port. The main features are outlined below.

- Allows programmable assignment of the synchronous serial interface, I²C, UART, and 16-bit PWM timer peripheral I/O functions to the P0 and P1 port groups.
- The peripheral I/O function assigned via UPMUX is enabled by setting the PxFNCSEL.PxyMUX[1:0] bits to 0x1.

Note: 'x', which is used in the port names Pxy, register names, and bit names, refers to a port group (x = 0, 1) and 'y' refers to a port number (y = 0, 1, 2, ..., 7).

Figure 7.1.1 shows the configuration of UPMUX.

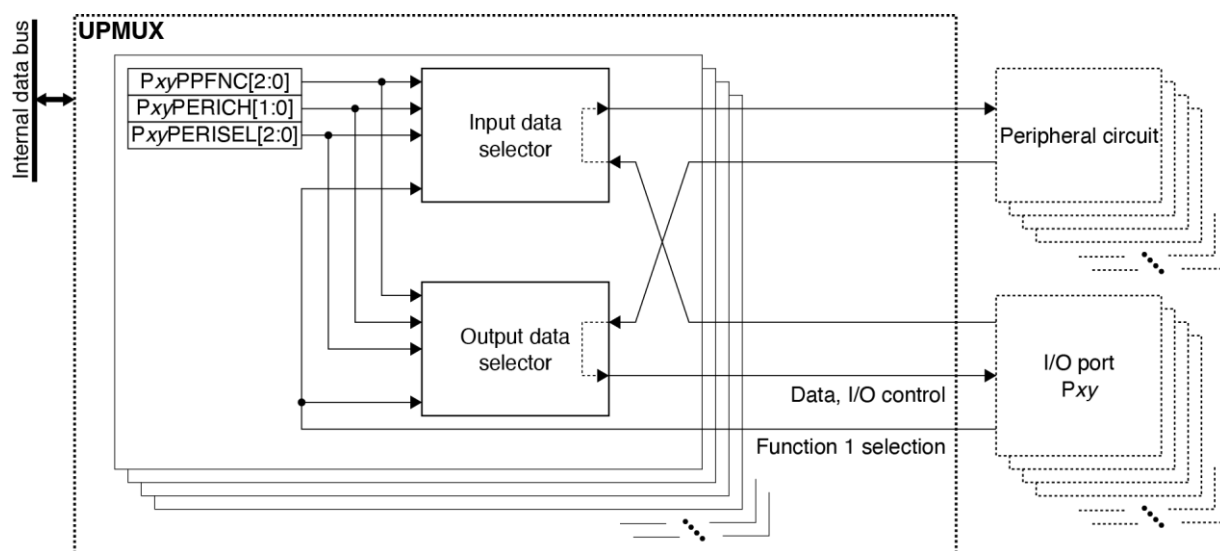


Figure 7.1.1 UPMUX Configuration

7.2 Peripheral Circuit I/O Function Assignment

An I/O function of a peripheral circuit supported may be assigned to peripheral I/O function 1 of an I/O port listed above. The following shows the procedure to assign a peripheral I/O function and enable it in the I/O port:

1. Configure the PxIOEN register of the I/O port.
 - Set the PxIOEN.PxIENy bit to 0. (Disable input)
 - Set the PxIOEN.PxOENy bit to 0. (Disable output)
2. Set the PxMODESEL.PxSELY bit of the I/O port to 0. (Disable peripheral I/O function)
3. Set the following PxUPMUXn register bits (n = 0 to 3).
 - PxUPMUXn.PxyPERISEL[2:0] bits (Select peripheral circuit)
 - PxUPMUXn.PxyPERICH[1:0] bits (Select peripheral circuit channel)
 - PxUPMUXn.PxyPPFNC[2:0] bits (Select function to assign)
4. Initialize the peripheral circuit.
5. Set the PxFNCSEL.PxyMUX[1:0] bits of the I/O port to 0x1. (Select peripheral I/O function 1)
6. Set the PxMODESEL.PxSELY bit of the I/O port to 1. (Enable peripheral I/O function)

7 Universal Port Multiplexer (UPMUX)

7.3 Control Registers

Pxy-zx Universal Port Multiplexer Setting Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxUPMUX n	15–13	PxzPPFNC[2:0]	0x0	H0	R/W	–
	12–11	PxzPERICH[1:0]	0x0	H0	R/W	
	10–8	PxzPERISEL[2:0]	0x0	H0	R/W	
	7–5	PxyPPFNC[2:0]	0x0	H0	R/W	
	4–3	PxyPERICH[1:0]	0x0	H0	R/W	
	2–0	PxyPERISEL[2:0]	0x0	H0	R/W	

*1: 'x' in the register name refers to a port group number and 'n' refers to a register number (0–3).

*2: 'x' in the bit name refers to a port group number, 'y' refers to an even port number (0, 2, 4, 6), and 'z' refers to an odd port number ($z = y + 1$).

Bits 15–13 PxzPPFNC[2:0]

Bits 7–5 PxyPPFNC[2:0]

These bits specify the peripheral I/O function to be assigned to the port. (See Table 7.3.1.)

Bits 12–11 PxzPERICH[1:0]

Bits 4–3 PxyPERICH[1:0]

These bits specify a peripheral circuit channel number. (See Table 7.3.1.)

Bits 10–8 PxzPERISEL[2:0]

Bits 2–0 PxyPERISEL[2:0]

These bits specify a peripheral circuit. (See Table 7.3.1.)

Table 7.3.1 Peripheral I/O Function Selections

PxUPMUX n . PxyPPFNC[2:0] bits (Peripheral I/O function)	PxUPMUX n .PxyPERISEL[2:0] bits (Peripheral circuit)										
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7			
	None *	I2C	SPIA	UART3	T16B	Reserved	Reserved	Reserved			
PxUPMUX n .PxyPERICH[1:0] bits (Peripheral circuit channel)	–	0x0	0x0	0x0	0x0–0x1	–	–	–			
	–	Ch.0	Ch.0	Ch.0	Ch.0–Ch.1	–	–	–			
	0x0	None *	None *	None *	None *	None *	None *	None *			
0x1	Reserved	SCL n	SDI n	USIN n	TOUT n 0/ CAP n 0	Reserved	Reserved	Reserved			
0x2		SDA n	SDO n	USOUT n	TOUT n 1/ CAP n 1						
0x3		Reserved	SPICLK n	Reserved	Reserved				Reserved		
0x4			#SPISS n								
0x5		Reserved	Reserved	Reserved	Reserved				Reserved	Reserved	Reserved
0x6											
0x7											

* "None" means no assignment. Selecting this will put the Pxy pin into Hi-Z status when peripheral I/O function 1 is selected and enabled in the I/O port.

Note: Do not assign a peripheral input function to two or more I/O ports. Although the I/O ports output the same waveforms when an output function is assigned to two or more I/O port, a skew occurs due to the internal delay.

8 Watchdog Timer (WDT2)

8.1 Overview

WDT2 restarts the system if a problem occurs, such as when the program cannot be executed normally. The features of WDT2 are listed below.

- Includes a 10-bit up counter to count NMI/reset generation cycle.
- A counter clock source and clock division ratio are selectable.
- Can generate a reset or NMI in a cycle given via software.
- Can generate a reset at the next NMI generation cycle after an NMI is generated.

Figure 8.1.1 shows the configuration of WDT2.

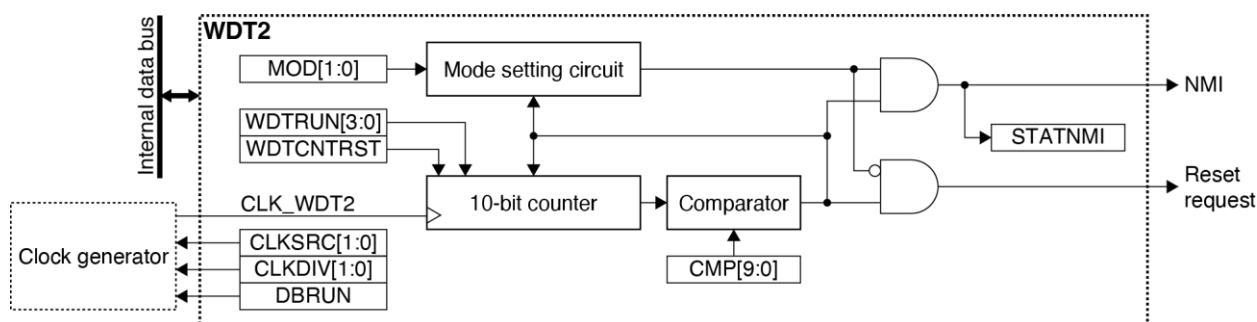


Figure 8.1.1 WDT2 Configuration

8.2 Clock Settings

8.2.1 WDT2 Operating Clock

When using WDT2, the WDT2 operating clock CLK_WDT2 must be supplied to WDT2 from the clock generator. The CLK_WDT2 supply should be controlled as in the procedure shown below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
3. Set the following WDTCLK register bits:
 - WDTCLK.CLKSRC[1:0] bits (Clock source selection)
 - WDTCLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

8.2.2 Clock Supply in DEBUG Mode

The CLK_WDT2 supply during DEBUG mode should be controlled using the WDTCLK.DBRUN bit.

The CLK_WDT2 supply to WDT2 is suspended when the CPU enters DEBUG mode if the WDTCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_WDT2 supply resumes. Although WDT2 stops operating when the CLK_WDT2 supply is suspended, the register retains the status before DEBUG mode was entered.

If the WDTCLK.DBRUN bit = 1, the CLK_WDT2 supply is not suspended and WDT2 will keep operating in DEBUG mode.

8 Watchdog Timer (WDT2)

8.3 Operations

8.3.1 WDT2 Control

Activating WDT2

WDT2 should be initialized and started up with the procedure listed below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Configure the WDT2 operating clock.
3. Set the WDTCTL.MOD[1:0] bits. (Select WDT2 operating mode)
4. Set the WDTCMP.CMP[9:0] bits. (Set NMI/reset generation cycle)
5. Write 1 to the WDTCTL.WDTCNTRST bit. (Reset WDT2 counter)
6. Write a value other than 0xa to the WDTCTL.WDTRUN[3:0] bits. (Start up WDT2)
7. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

NMI/reset generation cycle

Use the following equation to calculate the WDT2 NMI/reset generation cycle.

$$t_{\text{WDT}} = \frac{\text{CMP} + 1}{\text{CLK_WDT2}} \quad (\text{Eq. 8.1})$$

Where

- t_{WDT} : NMI/reset generation cycle [second]
 CLK_WDT2 : WDT2 operating clock frequency [Hz]
 CMP : Setting value of the WDTCMP.CMP[9:0] bits

Example) $t_{\text{WDT}} = 2.5$ seconds when $\text{CLK_WDT2} = 256$ Hz and the WDTCMP.CMP[9:0] bits = 639

Resetting WDT2 counter

To prevent an unexpected NMI/reset to be generated by WDT2, its embedded counter must be reset periodically via software while WDT2 is running.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Write 1 to the WDTCTL.WDTCNTRST bit. (Reset WDT2 counter)
3. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

A location should be provided for periodically processing this routine. Process this routine within the t_{WDT} cycle. After resetting, WDT2 starts counting with a new NMI/reset generation cycle.

Occurrence of counter compare match

If WDT2 is not reset within the t_{WDT} cycle for any reason and the counter reaches the setting value of the WDTCMP.CMP[9:0] bits, a compare match occurs to cause WDT2 to issue an NMI or reset according to the setting of the WDTCTL.MOD[1:0] bits.

If an NMI is issued, the WDTCTL.STATNMI bit is set to 1. This bit can be cleared to 0 by writing 1 to the WDTCTL.WDTCNTRST bit. Be sure to clear the WDTCTL.STATNMI bit in the NMI handler routine, If a compare match occurs, the counter is automatically reset to 0 and it continues counting.

Deactivating WDT2

WDT2 should be stopped with the procedure listed below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Write 0xa to the WDTCTL.WDTRUN[3:0] bits. (Stop WDT2)
3. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

8.3.2 Operations in HALT and SLEEP Modes

During HALT mode

WDT2 operates in HALT mode. HALT mode is therefore cleared by an NMI or reset if it continues for more than the NMI/reset generation cycle and the CPU executes the interrupt handler. To disable WDT2 in HALT mode, stop WDT2 by writing 0xa to the WDTCTL.WDTRUN[3:0] bits before executing the halt instruction. Reset WDT2 before resuming operations after HALT mode is cleared.

During SLEEP mode

WDT2 operates in SLEEP mode if the selected clock source is running. SLEEP mode is cleared by an NMI or reset if it continues for more than the NMI/reset generation cycle and the CPU executes the interrupt handler. Therefore, stop WDT2 by setting the WDTCTL.WDTRUN[3:0] bits before executing the slp instruction.

If the clock source stops in SLEEP mode, WDT2 stops. To prevent generation of an unnecessary NMI or reset after clearing SLEEP mode, reset WDT2 before executing the slp instruction. WDT2 should also be stopped as required using the WDTCTL.WDTRUN[3:0] bits.

8.4 Control Registers

WDT2 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
WDTCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/WP	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/WP	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/WP	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the WDT2 operating clock is supplied in DEBUG mode or not.

1 (R/WP): Clock supplied in DEBUG mode

0 (R/WP): No clock supplied in DEBUG mode

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

These bits select the division ratio of the WDT2 operating clock (counter clock). The clock frequency should be set to around 256 Hz.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of WDT2.

Table 8.4.1 Clock Source and Division Ratio Settings

WDTCLK. CLKDIV[1:0] bits	WDTCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x3	1/16,384	1/1,024	1/65,536	1/1
0x2	1/8,192	1/512	1/32,768	
0x1	1/4,096	1/256	1/16,384	
0x0	1/2,048	1/128	1/8,192	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

8 Watchdog Timer (WDT2)

WDT2 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
WDTCTL	15–11	–	0x00	–	R	–
	10–9	MOD[1:0]	0x0	H0	R/WP	
	8	STATNMI	0	H0	R	
	7–5	–	0x0	–	R	
	4	WDTCNTRST	0	H0	WP	Always read as 0.
	3–0	WDTRUN[3:0]	0xa	H0	R/WP	–

Bits 15–11 Reserved

Bits 10–9 MOD[1:0]

These bits set the WDT2 operating mode.

Table 8.4.2 Operating Mode Setting

WDTCTL. MOD[1:0] bits	Operating mode	Description
0x3	Reserved	–
0x2	RESET after NMI mode	If the WDTCTL.STATNMI bit is not cleared to 0 after an NMI has occurred due to a counter compare match, WDT2 issues a reset when the next compare match occurs.
0x1	NMI mode	WDT2 issues an NMI when a counter compare match occurs.
0x0	RESET mode	WDT2 issues a reset when a counter compare match occurs.

Bit 8 STATNMI

This bit indicates that a counter compare match and NMI have occurred.

1 (R): NMI (counter compare match) occurred

0 (R): NMI not occurred

When the NMI generation function of WDT2 is used, read this bit in the NMI handler routine to confirm that WDT2 was the source of the NMI.

The WDTCTL.STATNMI bit set to 1 is cleared to 0 by writing 1 to the WDTCTL.WDTCNTRST bit.

Bits 7–5 Reserved

Bit 4 WDTCNTRST

This bit resets the 10-bit counter and the WDTCTL.STATNMI bit.

1 (WP): Reset

0 (WP): Ignored

0 (R): Always 0 when being read

Bits 3–0 WDTRUN[3:0]

These bits control WDT2 to run and stop.

0xa (WP): Stop

Values other than 0xa (WP): Run

0xa (R): Idle

0x0 (R): Running

Always 0x0 is read if a value other than 0xa is written.

Since an NMI or reset may be generated immediately after running depending on the counter value, WDT2 should also be reset concurrently when running WDT2.

WDT2 Counter Compare Match Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
WDTCMP	15–10	–	0x00	–	R	–
	9–0	CMP[9:0]	0x3ff	H0	R/WP	

Bits 15–10 Reserved

Bits 9–0 CMP[9:0]

These bits set the NMI/reset generation cycle.

The value set in this register is compared with the 10-bit counter value while WDT2 is running, and an NMI or reset is generated when they are matched.

9 Supply Voltage Detector (SVD5)

9.1 Overview

SVD5 is a supply voltage detector to monitor the power supply voltage on the V_{DD} pin or the voltage applied to an external pin. The main features are listed below.

- Power supply voltage to be detected: Selectable from V_{DD} and an external power source (EXSVD) (Note: See the table below.)
- Detectable voltage level: Selectable from among 32 levels (max.) (Note: See the table below.)
- Detection results:
 - Can be read whether the power supply voltage is lower than the detection voltage level or not.
 - Can generate an interrupt or a reset when low power supply voltage is detected.
- Interrupt: 1 system (Low power supply voltage detection interrupt)
- Supports intermittent operations:
 - Three detection cycles are selectable.
 - Low power supply voltage detection count function to generate an interrupt/reset when low power supply voltage is successively detected the number of times specified.
 - Continuous operation is also possible.

Figure 9.1.1 shows the configuration of SVD5.

Table 9.1.1 SVD5 Configuration of S1C17W11

Item	S1C17W11
Power supply voltage to be detected	V_{DD} and one externally input voltage (EXSVD)
Detectable voltage level	V_{DD} : 32 levels (1.15 to 3.2 V)/external voltage: 32 levels (1.15 to 3.2 V)

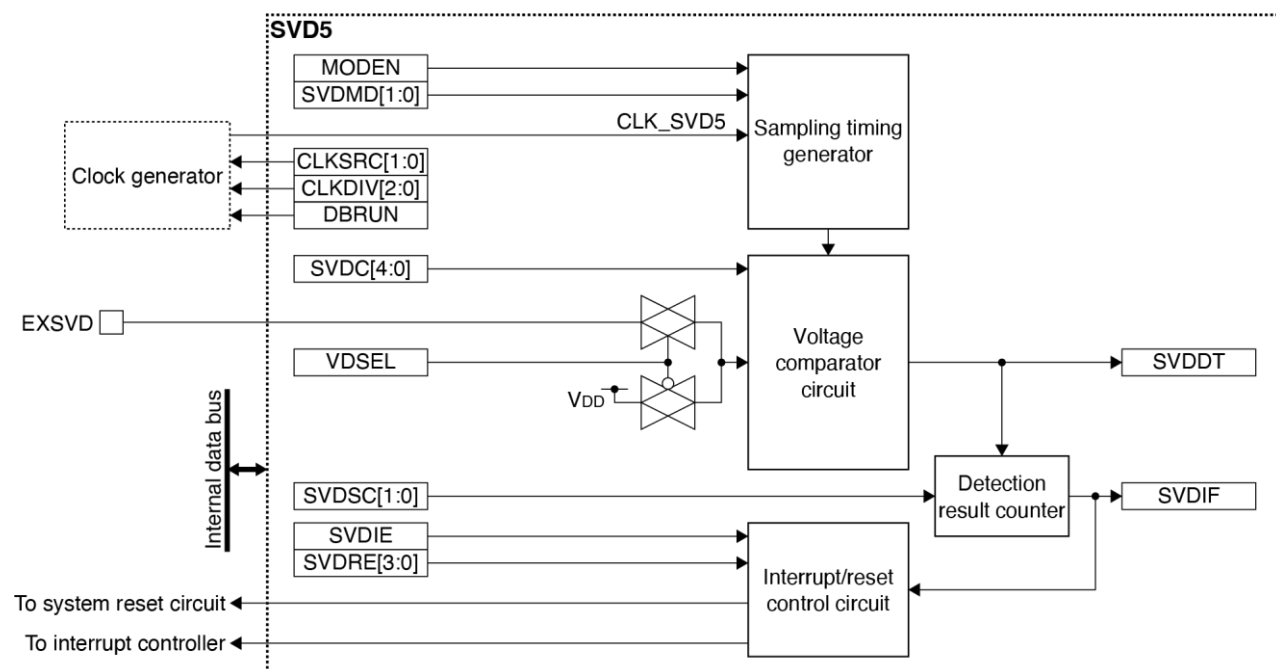


Figure 9.1.1 SVD5 Configuration

9 Supply Voltage Detector (SVD5)

9.2 Input Pins and External Connection

9.2.1 Input Pins

Table 9.2.1.1 shows the SVD5 input pins.

Table 9.2.1.1 SVD5 Input Pins

Pin name	I/O*	Initial status*	Function
EXSVD	A	A (Hi-Z)	External power supply voltage detection pin

* Indicates the status when the pin is configured for SVD5.

If the port is shared with the EXSVD pin and other functions, the EXSVD function must be assigned to the port before SVD5 can be activated. For more information, refer to the “I/O Ports” chapter.

9.2.2 External Connection

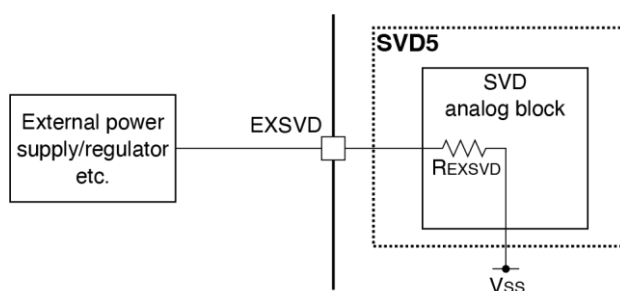


Figure 9.2.2.1 Connection between EXSVD Pin and External Power Supply

For the EXSVD pin input voltage range and the EXSVD input impedance, refer to “Supply Voltage Detector Characteristics” in the “Electrical Characteristics” chapter.

9.3 Clock Settings

9.3.1 SVD5 Operating Clock

When using SVD5, the SVD5 operating clock CLK_SVD5 must be supplied to SVD5 from the clock generator. The CLK_SVD5 supply should be controlled as in the procedure shown below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
3. Set the following SVDCLK register bits:
 - SVDCLK.CLKSRC[1:0] bits (Clock source selection)
 - SVDCLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

The CLK_SVD5 frequency should be set to around 32 kHz.

9.3.2 Clock Supply in SLEEP Mode

When using SVD5 during SLEEP mode, the SVD5 operating clock CLK_SVD5 must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_SVD5 clock source.

If the CLGOSC.xxxxSLPC bit for the CLK_SVD5 clock source is 1, the CLK_SVD5 clock source is deactivated during SLEEP mode and SVD5 stops with the register settings maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_SVD5 is supplied and the SVD5 operation resumes.

9.3.3 Clock Supply in DEBUG Mode

The CLK_SVD5 supply during DEBUG mode should be controlled using the SVDCLK.DBRUN bit. The CLK_SVD5 supply to SVD5 is suspended when the CPU enters DEBUG mode if the SVDCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_SVD5 supply resumes. Although SVD5 stops operating when the CLK_SVD5 supply is suspended, the registers retain the status before DEBUG mode was entered. If the SVDCLK.DBRUN bit = 1, the CLK_SVD5 supply is not suspended and SVD5 will keep operating in DEBUG mode.

9.4 Operations

9.4.1 SVD5 Control

Starting detection

SVD5 should be initialized and activated with the procedure listed below.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Configure the operating clock using the SVDCLK.CLKSRC[1:0] and SVDCLK.CLKDIV[2:0] bits.
3. Set the following SVDCTL register bits:
 - SVDCTL.VDSEL and SVDCTL.EXSEL bits (Select detection voltage (V_{DD} or EXSVD))
 - SVDCTL.SVDSC[1:0] bits (Set low power supply voltage detection counter)
 - SVDCTL.SVDC[4:0] bits (Set SVD detection voltage V_{SVD} /EXSVD detection voltage V_{SVD_EXT})
 - SVDCTL.SVDRE[3:0] bits (Select reset/interrupt mode)
 - SVDCTL.SVDMMD[1:0] bits (Set intermittent operation mode)
4. Set the following bits when using the interrupt:
 - Write 1 to the SVDINTF.SVDIF bit. (Clear interrupt flag)
 - Set the SVDINTE.SVDIE bit to 1. (Enable SVD5 interrupt)
5. Set the SVDCTL.MODEN bit to 1. (Enable SVD5 detection)
6. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Terminating detection

Follow the procedure shown below to stop SVD5 operation.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Write 0 to the SVDCTL.MODEN bit. (Disable SVD5 detection)
3. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)

Reading detection results

The following two detection results can be obtained by reading the SVDINTF.SVDDT bit:

- When SVDINTF.SVDDT bit = 0
Power supply voltage (V_{DD} or EXSVD) \geq SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}
- When SVDINTF.SVDDT bit = 1
Power supply voltage (V_{DD} or EXSVD) $<$ SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}

Before reading the SVDINTF.SVDDT bit, wait for at least SVD circuit enable response time after 1 is written to the SVDCTL.MODEN bit (refer to “Supply Voltage Detector Characteristics, SVD circuit enable response time t_{SVDEN} ” in the “Electrical Characteristics” chapter).

After the SVDCTL.SVDC[4:0] bits setting value is altered to change the SVD detection voltage V_{SVD} /EXSVD detection voltage V_{SVD_EXT} when the SVDCTL.MODEN bit = 1, wait for at least SVD circuit response time before reading the SVDINTF.SVDDT bit (refer to “Supply Voltage Detector Characteristics, SVD circuit response time t_{SVD} ” in the “Electrical Characteristics” chapter).

9 Supply Voltage Detector (SVD5)

9.4.2 SVD5 Operations

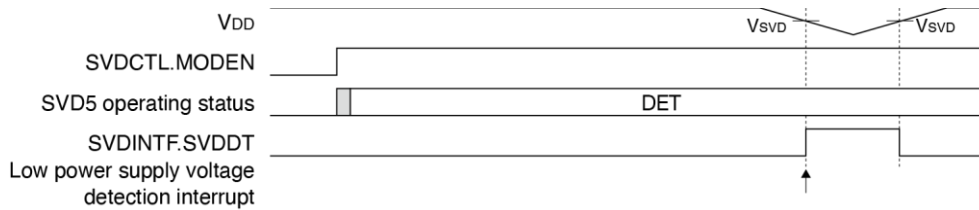
Continuous operation mode

SVD5 operates in continuous operation mode by default (SVDCTL.SVDMD[1:0] bits = 0x0). In this mode, SVD5 operates continuously while the SVDCTL.MODEN bit is set to 1 and it keeps loading the detection results to the SVDINTF.SVDDT bit. During this period, the current detection results can be obtained by reading the SVDINTF.SVDDT bit as necessary. Furthermore, an interrupt (if the SVDCTL.SVDRE[3:0] bits \neq 0xa) or a reset (if the SVDCTL.SVDRE[3:0] bits = 0xa) can be generated when the SVDINTF.SVDDT bit is set to 1 (low power supply voltage is detected). This mode can keep detecting power supply voltage drop after the voltage detection masking time has elapsed even if the IC is placed into SLEEP status or accidental clock stoppage has occurred.

Intermittent operation mode

SVD5 operates in intermittent operation mode when the SVDCTL.SVDMD[1:0] bits are set to 0x1 to 0x3. In this mode, SVD5 turns on at an interval set using the SVDCTL.SVDMD[1:0] bits to perform detection operation and then it turns off while the SVDCTL.MODEN bit is set to 1. During this period, the latest detection results can be obtained by reading the SVDINTF.SVDDT bit as necessary. Furthermore, an interrupt or a reset can be generated when SVD5 has successively detected low power supply voltage the number of times specified by the SVDCTL.SVDSC[1:0] bits.

- (1) When the SVDCTL.SVDMD[1:0] bits = 0x0 (continuous operation mode)



- (2) When the SVDCTL.SVDMD[1:0] bits \neq 0x0 (intermittent operation mode)

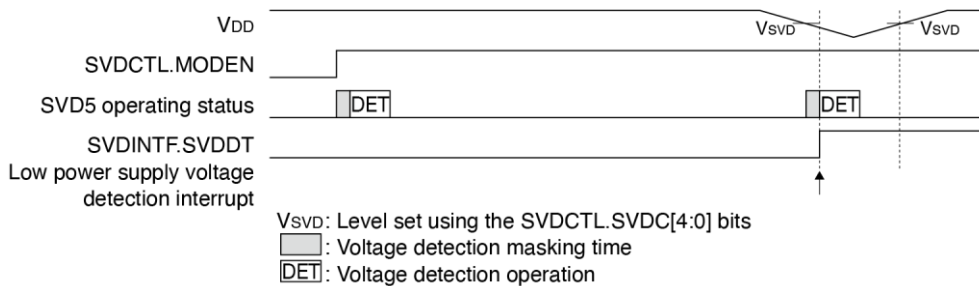


Figure 9.4.2.1 SVD5 Operations

9.5 SVD5 Interrupt and Reset

9.5.1 SVD5 Interrupt

Setting the SVDCTL.SVDRE[3:0] bits to a value other than 0xa allows use of the low power supply voltage detection interrupt function.

Table 9.5.1.1 Low Power Supply Voltage Detection Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Low power supply voltage detection	SVDINTF.SVDIF	In continuous operation mode When the SVDINTF.SVDDT bit is 1 In intermittent operation mode When low power supply voltage is successively detected the specified number of times	Writing 1

SVD5 provides the interrupt enable bit (SVDINTE.SVDIE bit) corresponding to the interrupt flag (SVDINTF.SVDIF bit). An interrupt request is sent to the interrupt controller only when the SVDINTF.SVDIF bit is set while the interrupt is enabled by the SVDINTE.SVDIE bit. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

Once the SVDINTF.SVDIF bit is set, it will not be cleared even if the power supply voltage subsequently returns to a value exceeding the SVD detection voltage V_{SVD} /EXSVD detection voltage V_{SVD_EXT} . An interrupt may occur due to a temporary power supply voltage drop, check the power supply voltage status by reading the SVDINTF.SVDDT bit in the interrupt handler routine.

9.5.2 SVD5 Reset

Setting the SVDCTL.SVDRE[3:0] bits to 0xa allows use of the SVD5 reset issuance function.

The reset issuing timing is the same as that of the SVDINTF.SVDIF bit being set when a low voltage is detected.

After a reset has been issued, SVD5 enters continuous operation mode even if it was operating in intermittent operation mode, and continues operating. Issuing an SVD5 reset initializes the port assignment. However, when EXSVD is being detected, the input of the port for the EXSVD pin is sent to SVD5 so that SVD5 will continue the EXSVD detection operation.

If the power supply voltage reverts to the normal level, the SVDINTF.SVDDT bit goes 0 and the reset state is canceled. After that, SVD5 resumes operating in the operation mode set previously via the initialization routine.

During reset state, the SVD5 control bits are set as shown in Table 9.5.2.1.

Table 9.5.2.1 SVD5 Control Bits During Reset State

Control register	Control bit	Setting
SVDCLK	DBRUN	Reset to the initial values.
	CLKDIV[2:0]	
	CLKSRC[1:0]	
SVDCTL	VDSEL	The set value is retained.
	SVDSC[1:0]	Cleared to 0. (The set value becomes invalid as SVD5 enters continuous operation mode.)
	SVDC[4:0]	The set value is retained.
	SVDRE[3:0]	The set value (0xa) is retained.
	EXSEL	The set value is retained.
	SVDMD[1:0]	Cleared to 0 to set continuous operation mode.
	MODEN	The set value (1) is retained.
SVDINTF	SVDIF	The status (1) before being reset is retained.
SVDINTE	SVDIE	Cleared to 0.

9.6 Control Registers

SVD5 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDCLK	15–9	–	0x00	–	R	–
	8	DBRUN	1	H0	R/WP	
	7	–	0	–	R	
	6–4	CLKDIV[2:0]	0x0	H0	R/WP	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/WP	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the SVD5 operating clock is supplied in DEBUG mode or not.

1 (R/WP): Clock supplied in DEBUG mode

0 (R/WP): No clock supplied in DEBUG mode

Bit 7 Reserved

Bits 6–4 CLKDIV[2:0]

These bits select the division ratio of the SVD5 operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of SVD5.

9 Supply Voltage Detector (SVD5)

Table 9.6.1 Clock Source and Division Ratio Settings

SVDCLK. CLKDIV[2:0] bits	SVDCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x7, 0x6	Reserved	1/1	Reserved	1/1
0x5	1/128		1/128	
0x4	1/64		1/64	
0x3	1/32		1/32	
0x2	1/16		1/16	
0x1	1/8		1/8	
0x0	1/4		1/4	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The clock frequency should be set to around 32 kHz.

SVD5 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDCTL	15	VDSEL	0	H1	R/WP	–
	14–13	SVSDSC[1:0]	0x0	H0	R/WP	Writing takes effect when the SVDCTL.SVDMD[1:0] bits are not 0x0.
	12–8	SVSDC[4:0]	0x1e	H1	R/WP	–
	7–4	SVSDRE[3:0]	0x0	H1	R/WP	–
	3	–	0	–	R	–
	2–1	SVSDMD[1:0]	0x0	H0	R/WP	–
	0	MODEN	0	H1	R/WP	–

Bit 15 VDSEL

This bit selects the power supply voltage to be detected by SVD5.

1 (R/WP): Voltage applied to the EXSVD pin

0 (R/WP): V_{DD}

Bits 14–13 SVSDSC[1:0]

These bits set the condition to generate an interrupt/reset (number of successive low voltage detections) in intermittent operation mode (SVDCTL.SVDMD[1:0] bits = 0x1 to 0x3).

Table 9.6.2 Interrupt/Reset Generating Condition in Intermittent Operation Mode

SVDCTL.SVSDSC[1:0] bits	Interrupt/reset generating condition
0x3	Low power supply voltage is successively detected eight times.
0x2	Low power supply voltage is successively detected four times.
0x1	Low power supply voltage is successively detected twice.
0x0	Low power supply voltage is successively detected once.

This setting is ineffective in continuous operation mode (SVDCTL.SVDMD[1:0] bits = 0x0).

Bits 12–8 SVSDC[4:0]

These bits select an SVD detection voltage V_{SVD} /EXSVD detection voltage V_{SVD_EXT} for detecting low voltage.

Table 9.6.3 Setting of SVD Detection Voltage V_{SVD} /EXSVD Detection Voltage V_{SVD_EXT}

SVDCTL.SVSDC[4:0] bits	SVD detection voltage V_{SVD} / EXSVD detection voltage V_{SVD_EXT} [V]
0x1f	High
0x1e	↑
0x1d	
⋮	
0x02	
0x01	↓
0x00	Low

For the configurable range and voltage values, refer to “Supply Voltage Detector Characteristics, SVD detection voltage V_{SVD} /EXSVD detection voltage V_{SVD_EXT} ” in the “Electrical Characteristics” chapter.

Bits 7–4 SVDRE[3:0]

These bits enable/disable the reset issuance function when a low power supply voltage is detected.

0xa (R/WP): Enable (Issue reset)
 Other than 0xa (R/WP): Disable (Generate interrupt)

For more information on the SVD5 reset issuance function, refer to “SVD5 Reset.”

Bits 3 Reserved

Bits 2–1 SVDMD[1:0]

These bits select intermittent operation mode and its detection cycle.

Table 9.6.4 Intermittent Operation Mode Detection Cycle Selection

SVDCTL.SVDMD[1:0] bits	Operation mode (detection cycle)
0x3	Intermittent operation mode (CLK_SVD5/8,192)
0x2	Intermittent operation mode (CLK_SVD5/4,096)
0x1	Intermittent operation mode (CLK_SVD5/2,048)
0x0	Continuous operation mode

For more information on intermittent and continuous operation modes, refer to “SVD5 Operations.”

Bit 0 MODEN

This bit enables/disables for the SVD5 circuit to operate.

1 (R/WP): Enable (Start detection operations)
 0 (R/WP): Disable (Stop detection operations)

After this bit has been altered, wait until the value written is read out from this bit without subsequent operations being performed.

- Notes:
- Writing 0 to the SVDCTL.MODEN bit resets the SVD5 hardware. However, the register values set and the interrupt flag are not cleared. The SVDCTL.MODEN bit is actually set to 0 after this processing has finished. If 1 is written to the SVDCTL.MODEN bit continuously without waiting for the bit being read as 0 at this time, writing 0 may be ignored and a malfunction may occur as the hardware restarts without resetting.
 - The SVD5 internal circuit is initialized if the SVDCTL.SVDSC[1:0] bits, SVDCTL.SVDRE[3:0] bits, or SVDCTL.SVDMD[1:0] bits are altered while SVD5 is in operation after 1 is written to the SVDCTL.MODEN bit.

SVD5 Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDINTF	15–9	–	0x00	–	R	–
	8	SVDDT	x	–	R	
	7–1	–	0x00	–	R	
	0	SVDIF	0	H1	R/W	Cleared by writing 1.

Bits 15–9 Reserved

Bit 8 SVDDT

The power supply voltage detection results can be read out from this bit.

1 (R): Power supply voltage (V_{DD} or EXSVD) < SVD detection voltage V_{SVD}
 or EXSVD detection voltage V_{SVD_EXT}
 0 (R): Power supply voltage (V_{DD} or EXSVD) \geq SVD detection voltage V_{SVD}
 or EXSVD detection voltage V_{SVD_EXT}

Bits 7–1 Reserved

Bit 0 SVDIF

This bit indicates the low power supply voltage detection interrupt cause occurrence status.

1 (R): Cause of interrupt occurred
 0 (R): No cause of interrupt occurred
 1 (W): Clear flag
 0 (W): Ineffective

9 Supply Voltage Detector (SVD5)

Note: The SVD5 internal circuit is initialized if the interrupt flag is cleared while SVD5 is in operation after 1 is written to the SVDCTL.MODEN bit.

SVD5 Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDINTE	15-8	-	0x00	-	R	-
	7-1	-	0x00	-	R	
	0	SVDIE	0	H0	R/W	

Bits 15-1 Reserved

Bit 0 SVDIE

This bit enables low power supply voltage detection interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

- Notes:
- If the SVDCTL.SVDRE[3:0] bits are set to 0xa, no low power supply voltage detection interrupt will occur, as a reset is issued at the same timing as an interrupt.
 - To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

10 16-bit Timers (T16)

10.1 Overview

T16 is a 16-bit timer. The features of T16 are listed below.

- 16-bit presetable down counter
- Provides a reload data register for setting the preset value.
- A clock source and clock division ratio for generating the count clock are selectable.
- Repeat mode or one-shot mode is selectable.
- Can generate counter underflow interrupts.

Figure 10.1.1 shows the configuration of a T16 channel.

Table 10.1.1 T16 Channel Configuration of S1C17W11

Item	S1C17W11
Number of channels	3 channels (Ch.0–Ch.2)
Event counter function	Not supported (No EXCL m pins are provided.)
Peripheral clock output (Outputs the counter underflow signal.)	Ch.1 → Synchronous serial interface Ch.0 master clock

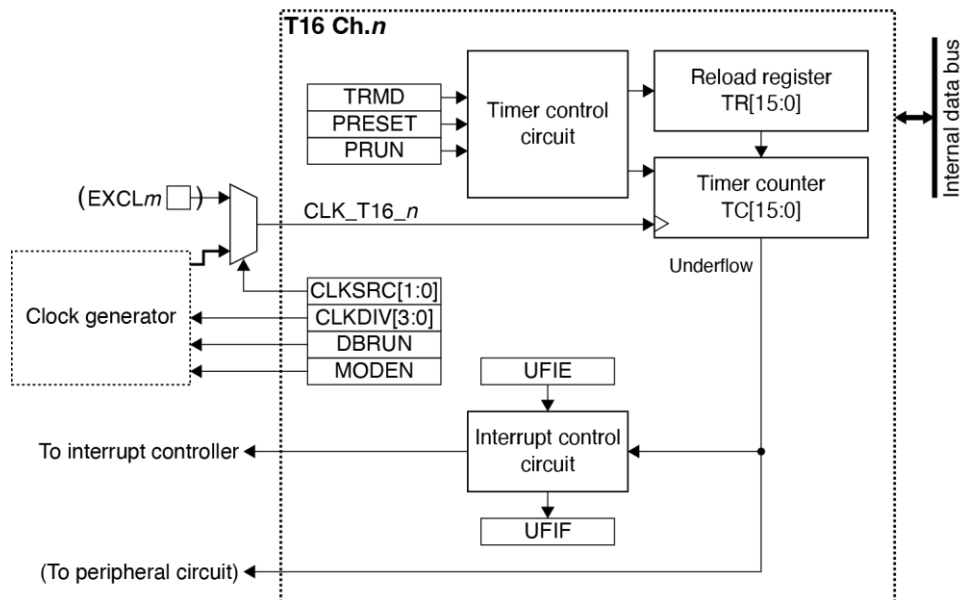


Figure 10.1.1 Configuration of a T16 Channel

10.2 Input Pin

Table 10.2.1 shows the T16 input pin.

Table 10.2.1 T16 Input Pin

Pin name	I/O*	Initial status*	Function
EXCL m	I	I (Hi-Z)	External event signal input pin

* Indicates the status when the pin is configured for T16.

If the port is shared with the EXCL m pin and other functions, the EXCL m input function must be assigned to the port before using the event counter function. For more information, refer to the “I/O Ports” chapter.

10.3 Clock Settings

10.3.1 T16 Operating Clock

When using T16 Ch.*n*, the T16 Ch.*n* operating clock CLK_T16_*n* must be supplied to T16 Ch.*n* from the clock generator. The CLK_T16_*n* supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
2. Set the following T16_*n*CLK register bits:
 - T16_*n*CLK.CLKSRC[1:0] bits (Clock source selection)
 - T16_*n*CLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)

10.3.2 Clock Supply in SLEEP Mode

When using T16 during SLEEP mode, the T16 operating clock CLK_T16_*n* must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_T16_*n* clock source.

If the CLGOSC.xxxxSLPC bit for the CLK_T16_*n* clock source is 1, the CLK_T16_*n* clock source is deactivated during SLEEP mode and T16 stops with the register settings and counter value maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_T16_*n* is supplied and the T16 operation resumes.

10.3.3 Clock Supply in DEBUG Mode

The CLK_T16_*n* supply during DEBUG mode should be controlled using the T16_*n*CLK.DBRUN bit.

The CLK_T16_*n* supply to T16 Ch.*n* is suspended when the CPU enters DEBUG mode if the T16_*n*CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_T16_*n* supply resumes. Although T16 Ch.*n* stops operating when the CLK_T16_*n* supply is suspended, the counter and registers retain the status before DEBUG mode was entered. If the T16_*n*CLK.DBRUN bit = 1, the CLK_T16_*n* supply is not suspended and T16 Ch.*n* will keep operating in DEBUG mode.

10.3.4 Event Counter Clock

The channel that supports the event counter function counts down at the rising edge of the EXCL_{*m*} pin input signal when the T16_*n*CLK.CLKSRC[1:0] bits are set to 0x3.

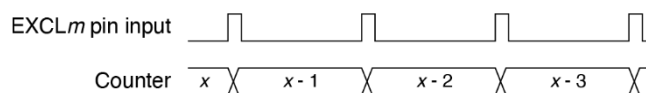


Figure 10.3.4.1 Count Down Timing

Note that the EXOSC clock is selected for the channel that does not support the event counter function.

10.4 Operations

10.4.1 Initialization

T16 Ch.*n* should be initialized and started counting with the procedure shown below.

1. Configure the T16 Ch.*n* operating clock (see “T16 Operating Clock”).
2. Set the T16_*n*CTL.MODEN bit to 1. (Enable count operation clock)
3. Set the T16_*n*MOD.TRMD bit. (Select operation mode (Repeat mode or One-shot mode))
4. Set the T16_*n*TR register. (Set reload data (counter preset data))
5. Set the following bits when using the interrupt:
 - Write 1 to the T16_*n*INTF.UFIF bit. (Clear interrupt flag)
 - Set the T16_*n*INTE.UFIE bit to 1. (Enable underflow interrupt)
6. Set the following T16_*n*CTL register bits:
 - Set the T16_*n*CTL.PRESET bit to 1. (Preset reload data to counter)
 - Set the T16_*n*CTL.PRUN bit to 1. (Start counting)

10.4.2 Counter Underflow

Normally, the T16 counter starts counting down from the reload data value preset and generates an underflow signal when an underflow occurs. This signal is used to generate an interrupt and may be output to a specific peripheral circuit as a clock (T16 Ch.*n* must be set to repeat mode to generate a clock). The underflow cycle is determined by the T16 Ch.*n* operating clock setting and reload data (counter initial value) set in the T16_*n*TR register. The following shows the equations to calculate the underflow cycle and frequency:

$$T = \frac{TR + 1}{f_{CLK_T16_n}} \quad f_T = \frac{f_{CLK_T16_n}}{TR + 1} \quad (\text{Eq. 10.1})$$

Where

- T: Underflow cycle [s]
- f_T: Underflow frequency [Hz]
- TR: T16_*n*TR register setting
- f_{CLK_T16_n}: T16 Ch.*n* operating clock frequency [Hz]

10.4.3 Operations in Repeat Mode

T16 Ch.*n* enters repeat mode by setting the T16_*n*MOD.TRMD bit to 0.

In repeat mode, the count operation starts by writing 1 to the T16_*n*CTL.PRUN bit and continues until 0 is written. A counter underflow presets the T16_*n*TR register value to the counter, so underflow occurs periodically. Select this mode to generate periodic underflow interrupts or when using the timer to output a trigger/clock to the peripheral circuit.

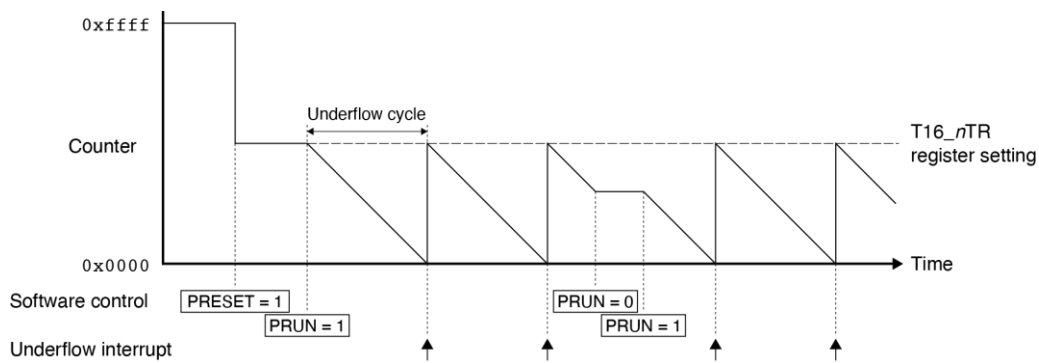


Figure 10.4.3.1 Count Operations in Repeat Mode

10.4.4 Operations in One-shot Mode

T16 Ch.*n* enters one-shot mode by setting the T16_*n*MOD.TRMD bit to 1.

In one-shot mode, the count operation starts by writing 1 to the T16_*n*CTL.PRUN bit and stops after the T16_*n*TR register value is preset to the counter when an underflow has occurred. At the same time the counter stops, the T16_*n*CTL.PRUN bit is cleared automatically. Select this mode to stop the counter after an interrupt has occurred once, such as for checking a specific lapse of time.

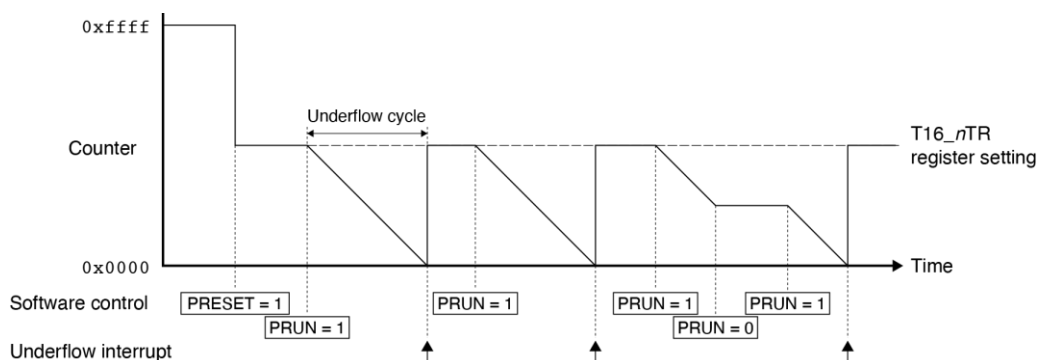


Figure 10.4.4.1 Count Operations in One-shot Mode

10 16-bit Timers (T16)

10.4.5 Counter Value Read

The counter value can be read out from the T16_nTC.TC[15:0] bits. However, since T16 operates on CLK_T16_n, one of the operations shown below is required to read correctly by the CPU.

- Read the counter value twice or more and check to see if the same value is read.
- Stop the timer and then read the counter value.

10.5 Interrupt

Each T16 channel has a function to generate the interrupt shown in Table 10.5.1.

Table 10.5.1 T16 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Underflow	T16_nINTF.UFIF	When the counter underflows	Writing 1

The T16 provides an interrupt enable bit corresponding to the interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

10.6 Control Registers

T16 Ch.n Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_nCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–4	CLKDIV[3:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 **DBRUN**

This bit sets whether the T16 Ch.n operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–4 **CLKDIV[3:0]**

These bits select the division ratio of the T16 Ch.n operating clock (counter clock).

Bits 3–2 Reserved

Bits 1–0 **CLKSRC[1:0]**

These bits select the clock source of T16 Ch.n.

Table 10.6.1 Clock Source and Division Ratio Settings

T16_nCLK. CLKDIV[3:0] bits	T16_nCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0xf	1/32,768	1/1	1/32,768	1/1
0xe	1/16,384		1/16,384	
0xd	1/8,192		1/8,192	
0xc	1/4,096		1/4,096	
0xb	1/2,048		1/2,048	
0xa	1/1,024		1/1,024	
0x9	1/512		1/512	
0x8	1/256	1/256	1/256	
0x7	1/128	1/128	1/128	
0x6	1/64	1/64	1/64	
0x5	1/32	1/32	1/32	
0x4	1/16	1/16	1/16	
0x3	1/8	1/8	1/8	
0x2	1/4	1/4	1/4	
0x1	1/2	1/2	1/2	
0x0	1/1	1/1	1/1	

(Note 1) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

(Note 2) When the T16_nCLK.CLKSRC[1:0] bits are set to 0x3, EXCL m is selected for the channel with an event counter function or EXOSC is selected for other channels.

T16 Ch.n Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_nMOD	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	TRMD	0	H0	R/W	

Bits 15–1 Reserved

Bit 0 TRMD

This bit selects the T16 operation mode.

1 (R/W): One-shot mode

0 (R/W): Repeat mode

For detailed information on the operation mode, refer to “Operations in One-shot Mode” and “Operations in Repeat Mode.”

T16 Ch.n Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_nCTL	15–9	–	0x00	–	R	–
	8	PRUN	0	H0	R/W	
	7–2	–	0x00	–	R	
	1	PRESET	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 PRUN

This bit starts/stops the timer.

1 (W): Start timer

0 (W): Stop timer

1 (R): Timer is running

0 (R): Timer is idle

By writing 1 to this bit, the timer starts count operations. However, the T16_nCTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance. While the timer is running, writing 0 to this bit stops count operations. When the counter stops due to a counter underflow in one-shot mode, this bit is automatically cleared to 0.

10 16-bit Timers (T16)

Bits 7–2 Reserved

Bit 1 PRESET

This bit presets the reload data stored in the T16_nTR register to the counter.

- 1 (W): Preset
- 0 (W): Ineffective
- 1 (R): Presetting in progress
- 0 (R): Presetting finished or normal operation

By writing 1 to this bit, the timer presets the T16_nTR register value to the counter. However, the T16_nCTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance. This bit retains 1 during presetting and is automatically cleared to 0 after presetting has finished.

Bit 0 MODEN

This bit enables the T16 Ch.n operations.

- 1 (R/W): Enable (Start supplying operating clock)
- 0 (R/W): Disable (Stop supplying operating clock)

T16 Ch.n Reload Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_nTR	15–0	TR[15:0]	0xffff	H0	R/W	–

Bits 15–0 TR[15:0]

These bits are used to set the initial value to be preset to the counter.

The value set to this register will be preset to the counter when 1 is written to the T16_nCTL.PRESET bit or when the counter underflows.

- Notes:
- The T16_nTR register cannot be altered while the timer is running (T16_nCTL.PRUN bit = 1), as an incorrect initial value may be preset to the counter.
 - When one-shot mode is set, the T16_nTR.TR[15:0] bits should be set to a value equal to or greater than 0x0001.

T16 Ch.n Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_nTC	15–0	TC[15:0]	0xffff	H0	R	–

Bits 15–0 TC[15:0]

The current counter value can be read out from these bits.

T16 Ch.n Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_nINTF	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	UFIF	0	H0	R/W	Cleared by writing 1.

Bits 15–1 Reserved

Bit 0 UFIF

This bit indicates the T16 Ch.n underflow interrupt cause occurrence status.

- 1 (R): Cause of interrupt occurred
- 0 (R): No cause of interrupt occurred
- 1 (W): Clear flag
- 0 (W): Ineffective

T16 Ch.n Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_nINTE	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	UFIE	0	H0	R/W	

Bits 15–1 Reserved

Bit 0 **UFIE**

This bit enables T16 Ch.*n* underflow interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

Note: To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

11 UART (UART3)

11.1 Overview

The UART3 is an asynchronous serial interface. The features of the UART3 are listed below.

- Includes a baud rate generator for generating the transfer clock.
- Supports 7- and 8-bit data length (LSB first).
- Odd parity, even parity, or non-parity mode is selectable.
- The start bit length is fixed at 1 bit.
- The stop bit length is selectable from 1 bit and 2 bits.
- Supports full-duplex communications.
- Includes a 2-byte receive data buffer and a 1-byte transmit data buffer.
- Includes an RZI modulator/demodulator circuit to support IrDA 1.0-compatible infrared communications.
- Can detect parity error, framing error, and overrun error.
- Can generate receive buffer full (1 byte/2 bytes), transmit buffer empty, end of transmission, parity error, framing error, and overrun error interrupts.
- Input pin can be pulled up with an internal resistor.
- The output pin is configurable as an open-drain output.
- Provides the carrier modulation output function.

Figure 11.1.1 shows the UART3 configuration.

Table 11.1.1 UART3 Channel Configuration of S1C17W11

Item	S1C17W11
Number of channels	1 channels (Ch.0)

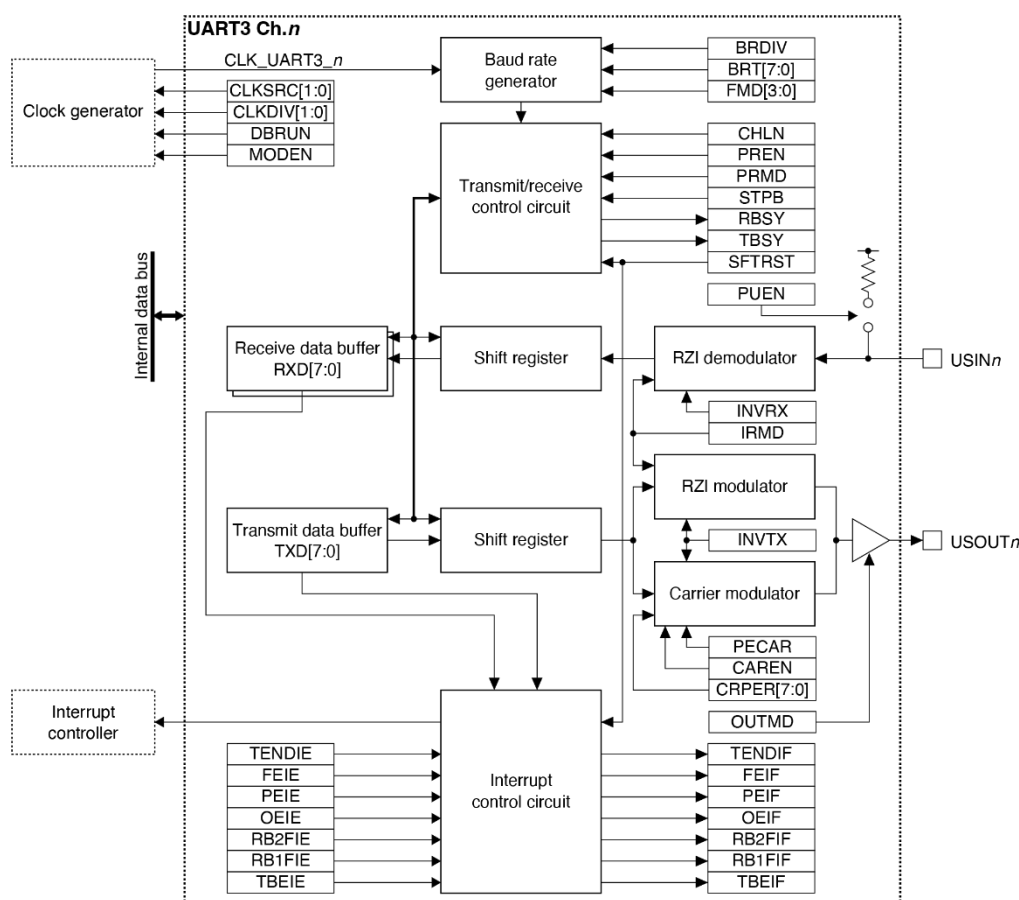


Figure 11.1.1 UART3 Configuration

11 UART (UART3)

11.2 Input/Output Pins and External Connections

11.2.1 List of Input/Output Pins

Table 11.2.1.1 lists the UART3 pins.

Table 11.2.1.1 List of UART3 Pins

Pin name	I/O*	Initial status*	Function
USIN n	I	I (Hi-Z)	UART3 Ch. n data input pin
USOUT n	O	O (High)	UART3 Ch. n data output pin

* Indicates the status when the pin is configured for the UART3.

If the port is shared with the UART3 pin and other functions, the UART3 input/output function must be assigned to the port before activating the UART3. For more information, refer to the “I/O Ports” chapter.

11.2.2 External Connections

Figure 11.2.2.1 shows a connection diagram between the UART3 in this IC and an external UART device.

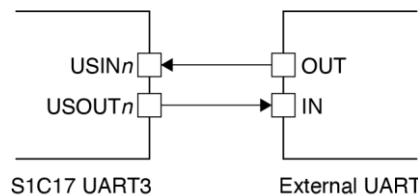


Figure 11.2.2.1 Connections between UART3 and an External UART Device

11.2.3 Input Pin Pull-Up Function

The UART3 includes a pull-up resistor for the USIN n pin. Setting the UAnMOD.PUEN bit to 1 enables the resistor to pull up the USIN n pin.

11.2.4 Output Pin Open-Drain Output Function

The USOUT n pin supports the open-drain output function. Default configuration is a push-pull output and it is switched to an open-drain output by setting the UAnMOD.OUTMD bit to 1.

11.2.5 Input/Output Signal Inverting Function

The UART3 can invert the signal polarities of the USIN n pin input and the USOUT n pin output by setting the UAnMOD.INVRX bit and the UAnMOD.INVTX bit, respectively, to 1.

Note: Unless otherwise specified, this chapter shows input/output signals with non-inverted waveforms (UAnMOD.INVRX bit = 0, UAnMOD.INVTX bit = 0).

11.3 Clock Settings

11.3.1 UART3 Operating Clock

When using the UART3 Ch. n , the UART3 Ch. n operating clock CLK_UART3_ n must be supplied to the UART3 Ch. n from the clock generator. The CLK_UART3_ n supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
2. Set the following UAnCLK register bits:
 - UAnCLK.CLKSRC[1:0] bits (Clock source selection)
 - UAnCLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)

The UART3 operating clock should be selected so that the baud rate generator will be configured easily.

11.3.2 Clock Supply in SLEEP Mode

When using the UART3 during SLEEP mode, the UART3 operating clock CLK_UART3_n must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_UART3_n clock source.

11.3.3 Clock Supply in DEBUG Mode

The CLK_UART3_n supply during DEBUG mode should be controlled using the UAnCLK.DBRUN bit. The CLK_UART3_n supply to the UART3 Ch.n is suspended when the CPU enters DEBUG mode if the UAnCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_UART3_n supply resumes. Although the UART3 Ch.n stops operating when the CLK_UART3_n supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the UAnCLK.DBRUN bit = 1, the CLK_UART3_n supply is not suspended and the UART3 Ch.n will keep operating in DEBUG mode.

11.3.4 Baud Rate Generator

The UART3 includes a baud rate generator to generate the transfer (sampling) clock. The transfer rate is determined by the UAnMOD.BRDIV, UAnBR.BRT[7:0], and UAnBR.FMD[3:0] bit settings. Use the following equations to calculate the setting values for obtaining the desired transfer rate.

$$\text{bps} = \frac{\text{CLK_UART3}}{\frac{\text{BRT} + 1}{\text{BRDIV}} + \text{FMD}} \quad \text{BRT} = \text{BRDIV} \times \left(\frac{\text{CLK_UART3}}{\text{bps}} - \text{FMD} \right) - 1 \quad (\text{Eq. 11.1})$$

Where

- bps: Transfer rate [bit/s]
- CLK_UART3: UART3 operating clock frequency [Hz]
- BRDIV: Baud rate division ratio (1/16 or 1/4) * Selected by the UAnMOD.BRDIV bit
- BRT: UAnBR.BRT[7:0] setting value (0 to 255)
- FMD: UAnBR.FMD[3:0] setting value (0 to 15)

For the transfer rate range configurable in the UART3, refer to “UART Characteristics, Transfer baud rates UBRT1 and UBRT2” in the “Electrical Characteristics” chapter.

11.4 Data Format

The UART3 allows setting of the data length, stop bit length, and parity function. The start bit length is fixed at one bit.

Data length

With the UAnMOD.CHLN bit, the data length can be set to seven bits (UAnMOD.CHLN bit = 0) or eight bits (UAnMOD.CHLN bit = 1).

Stop bit length

With the UAnMOD.STPB bit, the stop bit length can be set to one bit (UAnMOD.STPB bit = 0) or two bits (UAnMOD.STPB bit = 1).

Parity function

The parity function is configured using the UAnMOD.PREN and UAnMOD.PRMD bits.

Table 11.4.1 Parity Function Setting

UAnMOD.PREN bit	UAnMOD.PRMD bit	Parity function
1	1	Odd parity
1	0	Even parity
0	*	Non parit

11 UART (UART3)

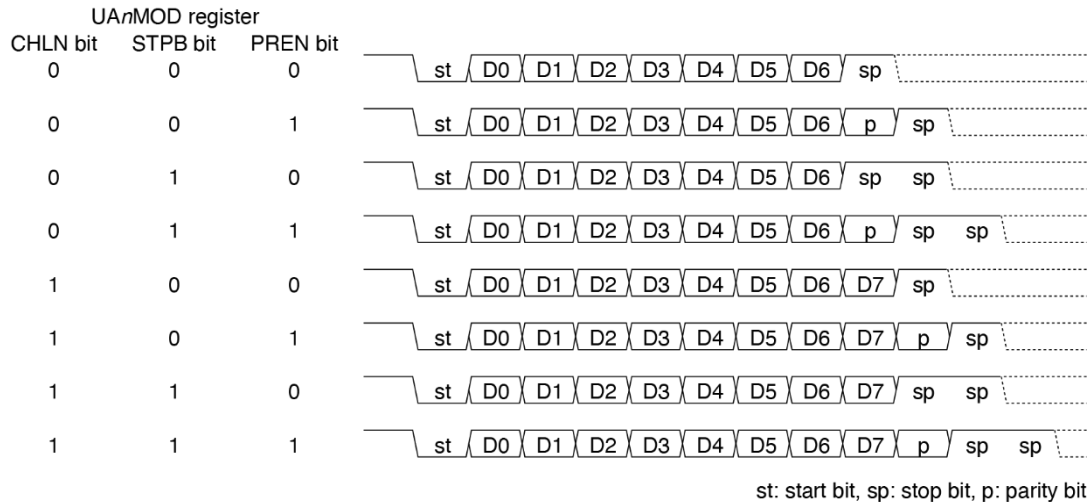


Figure 11.4.1 Data Format

11.5 Operations

11.5.1 Initialization

The UART3 Ch.*n* should be initialized with the procedure shown below.

1. Assign the UART3 Ch.*n* input/output function to the ports. (Refer to the “I/O Ports” chapter.)
2. Set the UAnCLK.CLKSRC[1:0] and UAnCLK.CLKDIV[1:0] bits. (Configure operating clock)
3. Configure the following UAnMOD register bits:
 - UAnMOD.BRDIV bit (Select baud rate division ratio (1/16 or 1/4))
 - UAnMOD.INVRX bit (Enable/disable USIN*n* input signal inversion)
 - UAnMOD.INVTX bit (Enable/disable USOUT*n* output signal inversion)
 - UAnMOD.PUEN bit (Enable/disable USIN*n* pin pull-up)
 - UAnMOD.OUTMD bit (Enable/disable USOUT*n* pin open-drain output)
 - UAnMOD.IRMD bit (Enable/disable IrDA interface)
 - UAnMOD.CHLN bit (Set data length (7 or 8 bits))
 - UAnMOD.PREN bit (Enable/disable parity function)
 - UAnMOD.PRMD bit (Select parity mode (even or odd))
 - UAnMOD.STPB bit (Set stop bit length (1 or 2 bits))
 - UAnMOD.CAREN bit (Enable/disable carrier modulation function)
 - UAnMOD.PECAR bit (Select carrier modulation period (H data period/L data period))
4. Set the UAnBR.BRT[7:0] and UAnBR.FMD[3:0] bits. (Set transfer rate)
5. Set the UAnCAWF.CRPER[7:0] bits. (Set carrier cycle)
6. Set the following UAnCTL register bits:
 - Set the UAnCTL.SFTRST bit to 1. (Execute software reset)
 - Set the UAnCTL.MODEN bit to 1. (Enable UART3 Ch.*n* operations)
7. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the UAnINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the UAnINTE register to 1. * (Enable interrupts)

* The initial value of the UAnINTF.TBEIF bit is 1, therefore, an interrupt will occur immediately after the UAnINTE.TBEIE bit is set to 1.

11.5.2 Data Transmission

A data sending procedure and the UART3 Ch.*n* operations are shown below. Figure 11.5.2.1 and Figure 11.5.2.2 show a timing chart and a flowchart, respectively.

Data sending procedure

1. Check to see if the UAnINTF.TBEIF bit is set to 1 (transmit buffer empty).
2. Write transmit data to the UAnTXD register.
3. Wait for a UART3 interrupt when using the interrupt.
4. Repeat Steps 1 to 3 (or 1 and 2) until the end of transmit data.

UART3 data sending operations

The UART3 Ch.n starts data sending operations when transmit data is written to the UAnTXD register.

The transmit data in the UAnTXD register is automatically transferred to the shift register and the UAnINTF.TBEIF bit is set to 1 (transmit buffer empty).

The USOUTn pin outputs a start bit and the UAnINTF.TBSY bit is set to 1 (transmit busy). The shift register data bits are then output successively from the LSB. Following output of MSB, the parity bit (if parity is enabled) and the stop bit are output.

Even if transmit data is being output from the USOUTn pin, the next transmit data can be written to the UAnTXD register after making sure the UAnINTF.TBEIF bit is set to 1.

If no transmit data remains in the UAnTXD register after the stop bit has been output from the USOUTn pin, the UAnINTF.TBSY bit is cleared to 0 and the UAnINTF.TENDIF bit is set to 1 (transmission completed).

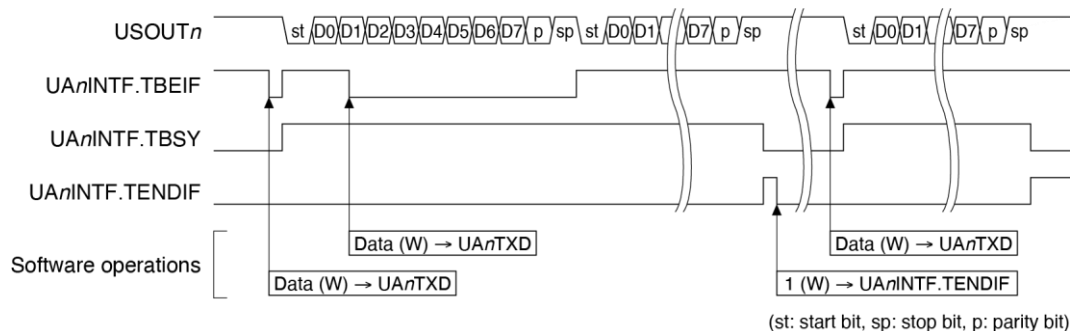


Figure 11.5.2.1 Example of Data Sending Operations

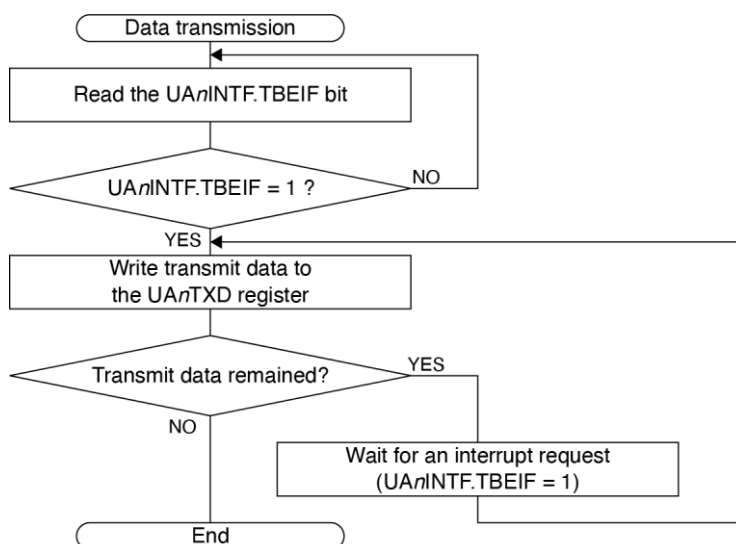


Figure 11.5.2.2 Data Transmission Flowchart

11 UART (UART3)

11.5.3 Data Reception

A data receiving procedure and the UART3 Ch.*n* operations are shown below. Figure 11.5.3.1 and Figure 11.5.3.2 show a timing chart and flowcharts, respectively.

Data receiving procedure (read by one byte)

1. Wait for a UART3 interrupt when using the interrupt.
2. Check to see if the `UAnINTF.RB1FIF` bit is set to 1 (receive buffer one byte full).
3. Read the received data from the `UAnRXD` register.
4. Repeat Steps 1 to 3 (or 2 and 3) until the end of data reception.

Data receiving procedure (read by two bytes)

1. Wait for a UART3 interrupt when using the interrupt.
2. Check to see if the `UAnINTF.RB2FIF` bit is set to 1 (receive buffer two bytes full).
3. Read the received data from the `UAnRXD` register twice.
4. Repeat Steps 1 to 3 (or 2 and 3) until the end of data reception.

UART3 data receiving operations

The UART3 Ch.*n* starts data receiving operations when a start bit is input to the `USINn` pin.

After the receive circuit has detected a low level as a start bit, it starts sampling the following data bits and loads the received data into the receive shift register. The `UAnINTF.RBSY` bit is set to 1 when the start bit is detected. The `UAnINTF.RBSY` bit is cleared to 0 and the receive shift register data is transferred to the receive data buffer at the stop bit receive timing.

The receive data buffer consists of a 2-byte FIFO and receives data until it becomes full. When the receive data buffer receives the first data, it sets the `UAnINTF.RB1FIF` bit to 1 (receive buffer one byte full). If the second data is received without reading the first data, the `UAnINTF.RB2FIF` bit is set to 1 (receive buffer two bytes full).

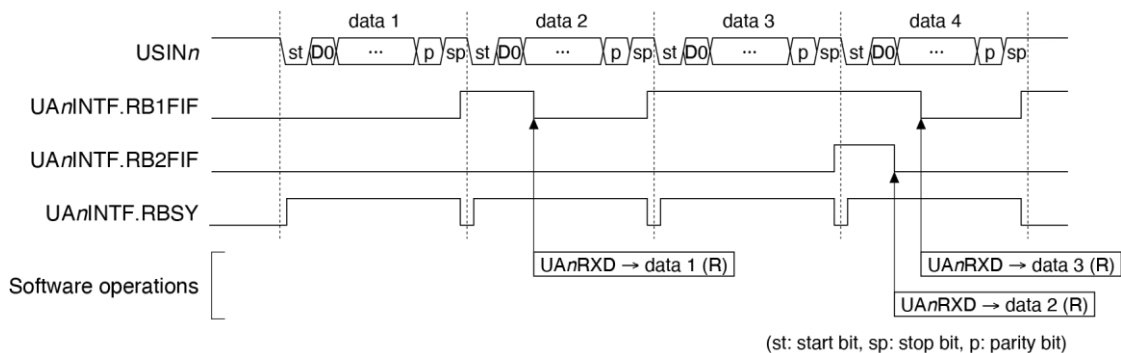


Figure 11.5.3.1 Example of Data Receiving Operations

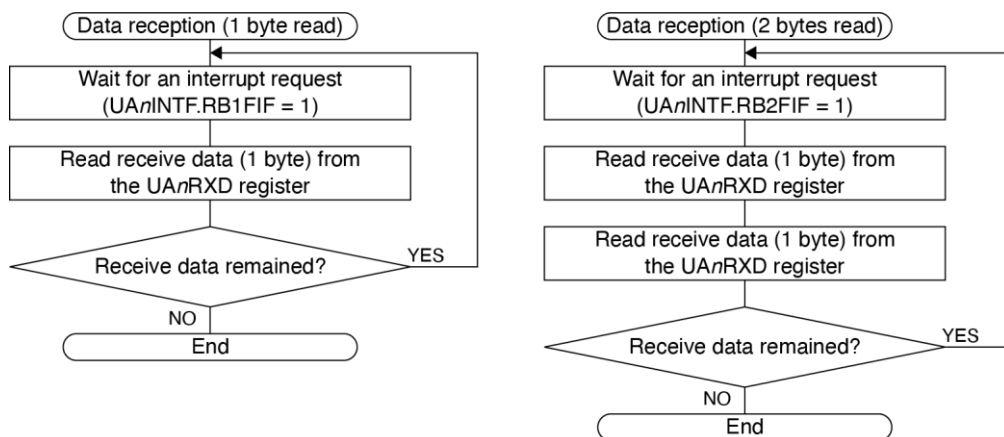


Figure 11.5.3.2 Data Reception Flowcharts

11.5.4 IrDA Interface

This UART3 includes an RZI modulator/demodulator circuit enabling implementation of IrDA 1.0-compatible infrared communication function simply by adding simple external circuits.

Set the $UA_nMOD.IRMD$ bit to 1 to use the IrDA interface.

Data transfer control is identical to that for normal interface even if the IrDA interface function is enabled.

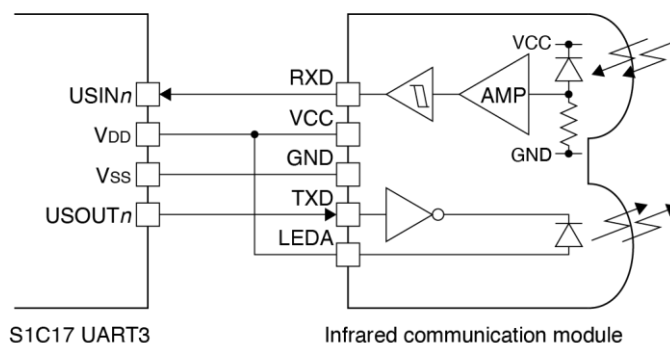


Figure 11.5.4.1 Example of Connections with an Infrared Communication Module

The transmit data output from the UART3 Ch. n transmit shift register is output from the $USOUT_n$ pin after the low pulse width is converted into $3/16$ by the RZI modulator in SIR method.

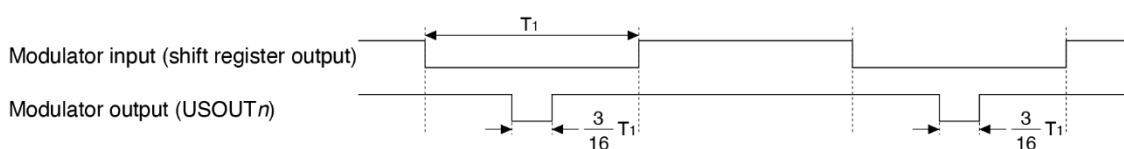


Figure 11.5.4.2 IrDA Transmission Signal Waveform

The received IrDA signal is input to the RZI demodulator and the low pulse width is converted into the normal width before input to the receive shift register.

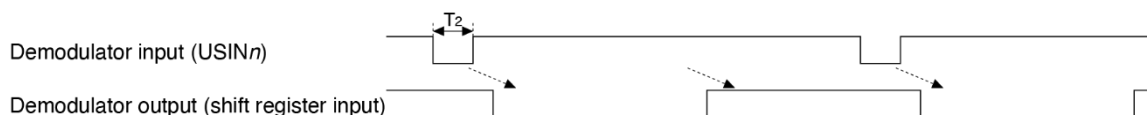


Figure 11.5.4.3 IrDA Receive Signal Waveform

- Notes:
- Set the baud rate division ratio to $1/16$ when using the IrDA interface function.
 - The low pulse width (T_2) of the IrDA signal input must be $CLK_UART3 \times 3$ cycles or longer.

11.5.5 Carrier Modulation

The UART3 has a carrier modulation function.

Writing 1 to the $UA_nMOD.CAREN$ bit enables the carrier modulation function allowing carrier modulation waveforms to be output according to the $UA_nMOD.PECAR$ bit setting. Data transmit control is identical to that for normal interface even in this case.

11 UART (UART3)

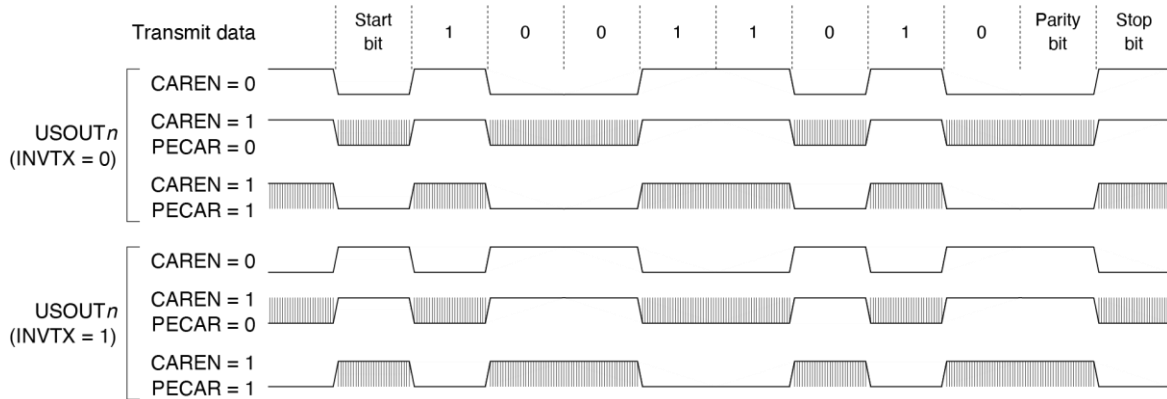


Figure 11.5.5.1 Carrier Modulation Waveform ($UAnMOD.CHNLN = 1$, $UAnMOD.STPB = 0$, $UAnMOD.PREN = 1$)

The carrier modulation output frequency is determined by the $UAnCAWF.CRPER[7:0]$ bit settings. Use the following equations to calculate the setting values for obtaining the desired frequency.

$$\text{Carrier modulation output frequency} = \frac{\text{CLK_UART3}}{(\text{CRPER} + 1) \times 2} \text{ [Hz]} \quad (\text{Eq. 11.2})$$

Where

CLK_UART3: UART3 operating clock frequency [Hz]
 CRPER: $UAnCAWF.CRPER[7:0]$ setting value (0 to 255)

11.6 Receive Errors

Three different receive errors, framing error, parity error, and overrun error, may be detected while receiving data. Since receive errors are interrupt causes, they can be processed by generating interrupts.

11.6.1 Framing Error

The UART3 determines loss of sync if a stop bit is not detected (when the stop bit is received as 0) and assumes that a framing error has occurred. The received data that encountered an error is still transferred to the receive data buffer and the $UAnINTF.FEIF$ bit (framing error interrupt flag) is set to 1 when the data becomes ready to read from the $UAnRXD$ register.

Note: Framing error/parity error interrupt flag set timings

These interrupt flags will be set after the data that encountered an error is transferred to the receive data buffer. Note, however, that the set timing depends on the buffer status at that point.

- When the receive data buffer is empty
The interrupt flag will be set when the data that encountered an error is transferred to the receive data buffer.
- When the receive data buffer has a one-byte free space
The interrupt flag will be set when the first data byte already loaded is read out after the data that encountered an error is transferred to the second byte entry of the receive data buffer.

11.6.2 Parity Error

If the parity function is enabled, a parity check is performed when data is received. The UART3 checks matching between the data received in the shift register and its parity bit, and issues a parity error if the result is a non-match. The received data that encountered an error is still transferred to the receive data buffer and the $UAnINTF.PEIF$ bit (parity error interrupt flag) is set to 1 when the data becomes ready to read from the $UAnRXD$ register (see the Note on framing error).

11.6.3 Overrun Error

If the receive data buffer is still full (two bytes of received data have not been read) when a data reception to the shift register has completed, an overrun error occurs as the data cannot be transferred to the receive data buffer. When an overrun error occurs, the $UAnINTF.OEIF$ bit (overrun error interrupt flag) is set to 1.

11.7 Interrupts

The UART3 has a function to generate the interrupts shown in Table 11.7.1.

Table 11.7.1 UART3 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
End of transmission	UAnINTF.TENDIF	When the UAnINTF.TBEIF bit = 1 after the stop bit has been sent	Writing 1 or software reset
Framing error	UAnINTF.FEIF	Refer to the "Receive Errors."	Writing 1, reading received data that encountered an error, or software reset
Parity error	UAnINTF.PEIF	Refer to the "Receive Errors."	Writing 1, reading received data that encountered an error, or software reset
Overrun error	UAnINTF.OEIF	Refer to the "Receive Errors."	Writing 1 or software reset
Receive buffer two bytes full	UAnINTF.RB2FIF	When the second received data byte is loaded to the receive data buffer in which the first byte is already received	Reading received data or software reset
Receive buffer one byte full	UAnINTF.RB1FIF	When the first received data byte is loaded to the emptied receive data buffer	Reading data to empty the receive data buffer or software reset
Transmit buffer empty	UAnINTF.TBEIF	When transmit data written to the transmit data buffer is transferred to the shift register	Writing transmit data

The UART3 provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

11.8 Control Registers

UART3 Ch.n Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UAnCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 **DBRUN**

This bit sets whether the UART3 operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–6 Reserved

Bits 5–4 **CLKDIV[1:0]**

These bits select the division ratio of the UART3 operating clock.

Bits 3–2 Reserved

Bits 1–0 **CLKSRC[1:0]**

These bits select the clock source of the UART3.

11 UART (UART3)

Table 11.8.1 Clock Source and Division Ratio Settings

UAnCLK. CLKDIV[1:0] bits	UAnCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x3	1/8	1/1	1/8	1/1
0x2	1/4		1/4	
0x1	1/2		1/2	
0x0	1/1		1/1	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The UAnCLK register settings can be altered only when the UAnCTL.MODEN bit = 0.

UART3 Ch.n Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UAnMOD	15–13	–	0x0	–	R	–
	12	PECAR	0	H0	R/W	
	11	CAREN	0	H0	R/W	
	10	BRDIV	0	H0	R/W	
	9	INVRX	0	H0	R/W	
	8	INVTX	0	H0	R/W	
	7	–	0	–	R	
	6	PUEN	0	H0	R/W	
	5	OUTMD	0	H0	R/W	
	4	IRMD	0	H0	R/W	
	3	CHLN	0	H0	R/W	
	2	PREN	0	H0	R/W	
	1	PRMD	0	H0	R/W	
0	STPB	0	H0	R/W		

Bits 15–13 Reserved

Bit 12 PECAR

This bit selects the carrier modulation period.

1 (R/W): Carrier modulation during H data period

0 (R/W): Carrier modulation during L data period

Bit 11 CAREN

This bit enables the carrier modulation function.

1 (R/W): Enable carrier modulation function

0 (R/W): Disable carrier modulation function

Bit 10 BRDIV

This bit sets the UART3 operating clock division ratio for generating the transfer (sampling) clock using the baud rate generator.

1 (R/W): 1/4

0 (R/W): 1/16

Bit 9 INVRX

This bit enables the USIN n input inverting function.

1 (R/W): Enable input inverting function

0 (R/W): Disable input inverting function

Bit 8 INVTX

This bit enables the USOUT n output inverting function.

1 (R/W): Enable output inverting function

0 (R/W): Disable output inverting function

Bit 7 Reserved

- Bit 6 PUEN**
This bit enables pull-up of the USIN_n pin.
1 (R/W): Enable pull-up
0 (R/W): Disable pull-up
- Bit 5 OUTMD**
This bit sets the USOUT_n pin output mode.
1 (R/W): Open-drain output
0 (R/W): Push-pull output
- Bit 4 IRMD**
This bit enables the IrDA interface function.
1 (R/W): Enable IrDA interface function
0 (R/W): Disable IrDA interface function
- Bit 3 CHLN**
This bit sets the data length.
1 (R/W): 8 bits
0 (R/W): 7 bits
- Bit 2 PREN**
This bit enables the parity function.
1 (R/W): Enable parity function
0 (R/W): Disable parity function
- Bit 1 PRMD**
This bit selects either odd parity or even parity when using the parity function.
1 (R/W): Odd parity
0 (R/W): Even parity
- Bit 0 STPB**
This bit sets the stop bit length.
1 (R/W): 2 bits
0 (R/W): 1 bit

- Notes:
- The UAnMOD register settings can be altered only when the UAnCTL.MODEN bit = 0.
 - Do not set both the UAnMOD.IRMD and UAnMOD.CAREN bits simultaneously.

UART3 Ch.n Baud-Rate Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UAnBR	15–12	–	0x0	–	R	–
	11–8	FMD[3:0]	0x0	H0	R/W	
	7–0	BRT[7:0]	0x00	H0	R/W	

Bits 15–12 Reserved

Bits 11–8 FMD[3:0]

Bits 7–0 BRT[7:0]

These bits set the UART3 transfer rate. For more information, refer to “Baud Rate Generator.”

- Notes:
- The UAnBR register settings can be altered only when the UAnCTL.MODEN bit = 0.
 - Do not set the UAnBR.FMD[3:0] bits to a value other than 0 to 3 when the UAnMOD.BRDIV bit = 1.

UART3 Ch.n Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UAnCTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	SFTRST	0	H0	R/W	
	0	MODEN	0	H0	R/W	

11 UART (UART3)

Bits 15–2 Reserved

Bit 1 SFTRST

This bit issues software reset to the UART3.

- 1 (W): Issue software reset
- 0 (W): Ineffective
- 1 (R): Software reset is executing.
- 0 (R): Software reset has finished. (During normal operation)

Setting this bit resets the UART3 transmit/receive control circuit and interrupt flags. This bit is automatically cleared after the reset processing has finished.

Bit 0 MODEN

This bit enables the UART3 operations.

- 1 (R/W): Enable UART3 operations (The operating clock is supplied.)
- 0 (R/W): Disable UART3 operations (The operating clock is stopped.)

Note: If the UAnCTL.MODEN bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the UAnCTL.MODEN bit to 1 again after that, be sure to write 1 to the UAnCTL.SFTRST bit as well.

UART3 Ch.n Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UAnTXD	15–8	–	0x00	–	R	–
	7–0	TXD[7:0]	0x00	H0	R/W	

Bits 15–8 Reserved

Bits 7–0 TXD[7:0]

Data can be written to the transmit data buffer through these bits. Make sure the UAnINTF.TBEIF bit is set to 1 before writing data.

UART3 Ch.n Receive Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UAnRXD	15–8	–	0x00	–	R	–
	7–0	RXD[7:0]	0x00	H0	R	

Bits 15–8 Reserved

Bits 7–0 RXD[7:0]

The receive data buffer can be read through these bits. The receive data buffer consists of a 2-byte FIFO, and older received data is read first.

UART3 Ch.n Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	
UAnINTF	15–10	–	0x00	–	R	–	
	9	RBSY	0	H0/S0	R		
	8	TBSY	0	H0/S0	R		
	7	–	0	–	R		
	6	TENDIF	0	H0/S0	R/W		Cleared by writing 1.
	5	FEIF	0	H0/S0	R/W		Cleared by writing 1 or reading the UAnRXD register.
	4	PEIF	0	H0/S0	R/W		Cleared by writing 1.
	3	OEIF	0	H0/S0	R/W		Cleared by reading the UAnRXD register.
	2	RB2FIF	0	H0/S0	R		Cleared by writing to the UAnTXD register.
	1	RB1FIF	0	H0/S0	R		
0	TBEIF	1	H0/S0	R			

Bits 15–10 Reserved

Bit 9 RBSY

This bit indicates the receiving status. (See Figure 11.5.3.1.)

- 1 (R): During receiving
- 0 (R): Idle

Bit 8 TBSY
 This bit indicates the sending status. (See Figure 11.5.2.1.)
 1 (R): During sending
 0 (R): Idle

Bit 7 Reserved

Bit 6 TENDIF

Bit 5 FEIF

Bit 4 PEIF

Bit 3 OEIF

Bit 2 RB2FIF

Bit 1 RB1FIF

Bit 0 TBEIF

These bits indicate the UART3 interrupt cause occurrence status.

1 (R): Cause of interrupt occurred
 0 (R): No cause of interrupt occurred
 1 (W): Clear flag
 0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

UA n INTF.TENDIF bit: End-of-transmission interrupt
 UA n INTF.FEIF bit: Framing error interrupt
 UA n INTF.PEIF bit: Parity error interrupt
 UA n INTF.OEIF bit: Overrun error interrupt
 UA n INTF.RB2FIF bit: Receive buffer two bytes full interrupt
 UA n INTF.RB1FIF bit: Receive buffer one byte full interrupt
 UA n INTF.TBEIF bit: Transmit buffer empty interrupt

UART3 Ch. n Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UA n INTE	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6	TENDIE	0	H0	R/W	
	5	FEIE	0	H0	R/W	
	4	PEIE	0	H0	R/W	
	3	OEIE	0	H0	R/W	
	2	RB2FIE	0	H0	R/W	
	1	RB1FIE	0	H0	R/W	
0	TBEIE	0	H0	R/W		

Bits 15–7 Reserved

Bit 6 TENDIE

Bit 5 FEIE

Bit 4 PEIE

Bit 3 OEIE

Bit 2 RB2FIE

Bit 1 RB1FIE

Bit 0 TBEIE

These bits enable UART3 interrupts.

1 (R/W): Enable interrupts
 0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

UA n INTE.TENDIE bit: End-of-transmission interrupt
 UA n INTE.FEIE bit: Framing error interrupt
 UA n INTE.PEIE bit: Parity error interrupt
 UA n INTE.OEIE bit: Overrun error interrupt
 UA n INTE.RB2FIE bit: Receive buffer two bytes full interrupt
 UA n INTE.RB1FIE bit: Receive buffer one byte full interrupt
 UA n INTE.TBEIE bit: Transmit buffer empty interrupt

11 UART (UART3)

UART3 Ch.*n* Carrier Waveform Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UAnCAWF	15–8	–	0x00	–	R	–
	7–0	CRPER[7:0]	0x00	H0	R/W	

Bits 15–8 **Reserved**

Bits 7–0 **CRPER[7:0]**

These bits set the carrier modulation output frequency. For more information, refer to “Carrier Modulation.”

12 Synchronous Serial Interface (SPIA)

12.1 Overview

SPIA is a synchronous serial interface. The features of SPIA are listed below.

- Supports both master and slave modes.
- Data length: 2 to 16 bits programmable
- Either MSB first or LSB first can be selected for the data format.
- Clock phase and polarity are configurable.
- Supports full-duplex communications.
- Includes separated transmit data buffer and receive data buffer registers.
- Can generate receive buffer full, transmit buffer empty, end of transmission, and overrun interrupts.
- Master mode allows use of a 16-bit timer to set baud rate.
- Slave mode is capable of being operated with the external input clock SPICLK_n only.
- Slave mode is capable of being operated in SLEEP mode allowing wake-up by an SPIA interrupt.
- Input pins can be pulled up/down with an internal resistor.

Figure 12.1.1 shows the SPIA configuration.

Table 12.1.1 SPIA Channel Configuration of S1C17W11

Item	S1C17W11
Number of channels	1 channels (Ch.0)
Internal clock input	Ch.0 ← 16-bit timer Ch.1

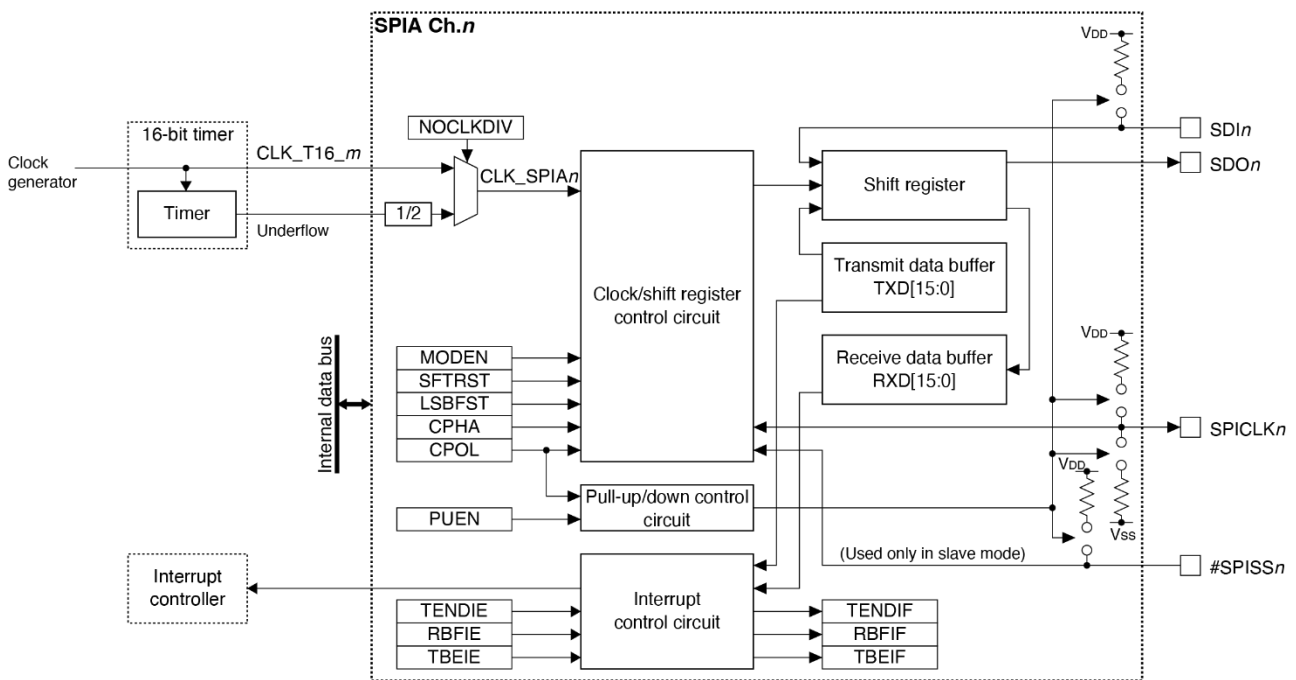


Figure 12.1.1 SPIA Configuration

12 Synchronous Serial Interface (SPIA)

12.2 Input/Output Pins and External Connections

12.2.1 List of Input/Output Pins

Table 12.2.1.1 lists the SPIA pins.

Table 12.2.1.1 List of SPIA Pins

Pin name	I/O*	Initial status*	Function
SDIn	I	I (Hi-Z)	SPIA Ch.n data input pin
SDOn	O or Hi-Z	Hi-Z	SPIA Ch.n data output pin
SPICLK _n	I or O	I (Hi-Z)	SPIA Ch.n external clock input/output pin
#SPISS _n	I	I (Hi-Z)	SPIA Ch.n slave select signal input pin

* Indicates the status when the pin is configured for SPIA.

If the port is shared with the SPIA pin and other functions, the SPIA input/output function must be assigned to the port before activating SPIA. For more information, refer to the “I/O Ports” chapter.

12.2.2 External Connections

SPIA operates in master mode or slave mode. Figure 12.2.2.1 and Figure 12.2.2.2 show connection diagrams between SPIA in each mode and external SPI devices.

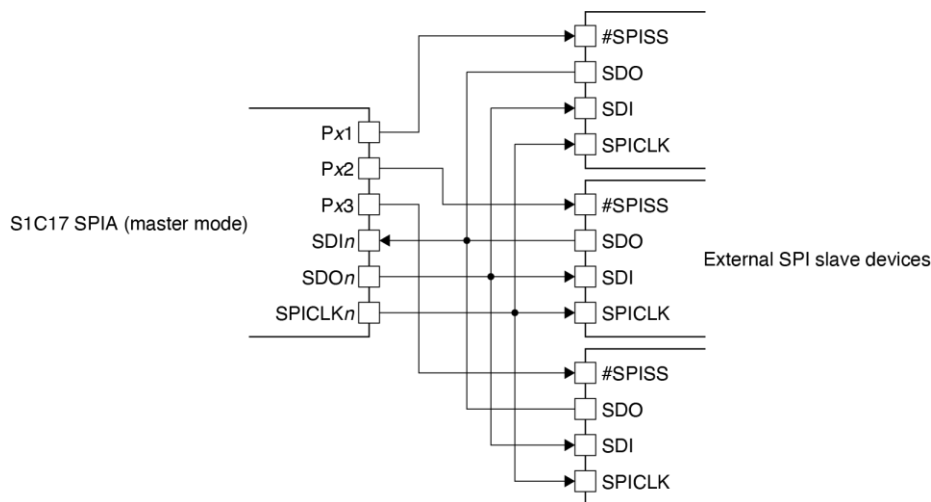


Figure 12.2.2.1 Connections between SPIA in Master Mode and External SPI Slave Devices

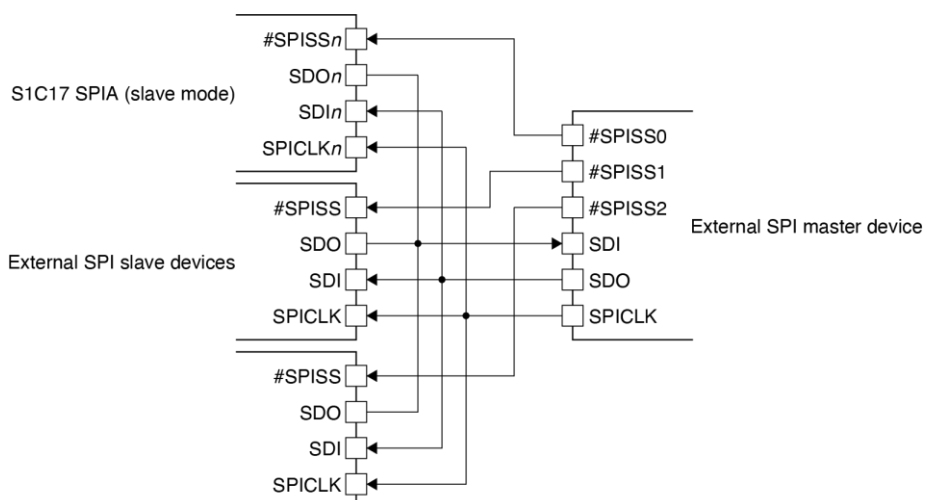


Figure 12.2.2.2 Connections between SPIA in Slave Mode and External SPI Master Device

12.2.3 Pin Functions in Master Mode and Slave Mode

The pin functions are changed according to the master or slave mode selection. The differences in pin functions between the modes are shown in Table 12.2.3.1.

Table 12.2.3.1 Pin Function Differences between Modes

Pin	Function in master mode	Function in slave mode
<i>SDIn</i>	Always placed into input state.	
<i>SDOn</i>	Always placed into output state.	This pin is placed into output state while a low level is applied to the #SPISS <i>n</i> pin or placed into Hi-Z state while a high level is applied to the #SPISS <i>n</i> pin.
<i>SPICLK_n</i>	Outputs the SPI clock to external devices. Output clock polarity and phase can be configured if necessary.	Inputs an external SPI clock. Clock polarity and phase can be designated according to the input clock.
#SPISS <i>n</i>	Not used. This input function is not required to be assigned to the port. To output the slave select signal in master mode, use a general-purpose I/O port function.	Applying a low level to the #SPISS <i>n</i> pin enables SPIA to transmit/receive data. While a high level is applied to this pin, SPIA is not selected as a slave device. Data input to the <i>SDIn</i> pin and the clock input to the <i>SPICLK_n</i> pin are ignored. When a high level is applied, the transmit/receive bit count is cleared to 0 and the already received bits are discarded.

12.2.4 Input Pin Pull-Up/Pull-Down Function

The SPIA input pins (*SDIn* in master mode or *SDIn*, *SPICLK_n*, and #SPISS*n* pins in slave mode) have a pull-up or pull-down function as shown in Table 12.2.4.1. This function is enabled by setting the *SPI_nMOD.PUEN* bit to 1.

Table 12.2.4.1 Pull-Up or Pull-Down of Input Pins

Pin	Master mode	Slave mode
<i>SDIn</i>	Pull-up	Pull-up
<i>SPICLK_n</i>	–	<i>SPI_nMOD.CPOL</i> bit = 1: Pull-up <i>SPI_nMOD.CPOL</i> bit = 0: Pull-down
#SPISS <i>n</i>	–	Pull-up

12.3 Clock Settings

12.3.1 SPIA Operating Clock

Operating clock in master mode

In master mode, the SPIA operating clock is supplied from the 16-bit timer. The following two options are provided for the clock configuration.

Use the 16-bit timer operating clock without dividing

By setting the *SPI_nMOD.NOCLKDIV* bit to 1, the operating clock *CLK_T16_m*, which is configured by selecting a clock source and a division ratio, for the 16-bit timer channel corresponding to the SPIA channel is input to SPIA as *CLK_SPIA_n*. Since this clock is also used as the SPI clock *SPICLK_n* without changing, the *CLK_SPIA_n* frequency becomes the baud rate.

To supply *CLK_SPIA_n* to SPIA, the 16-bit timer clock source must be enabled in the clock generator. It does not matter how the *T16_mCTL.MODEN* and *T16_mCTL.PRUN* bits of the corresponding 16-bit timer channel are set (1 or 0).

When setting this mode, the timer function of the corresponding 16-bit timer channel may be used for another purpose.

Use the 16-bit timer as a baud rate generator

By setting the *SPI_nMOD.NOCLKDIV* bit to 0, SPIA inputs the underflow signal generated by the corresponding 16-bit timer channel and converts it to the *SPICLK_n*. The 16-bit timer must be run with an appropriate reload data set. The *SPICLK_n* frequency (baud rate) and the 16-bit timer reload data are calculated by the equations shown below.

12 Synchronous Serial Interface (SPIA)

$$f_{\text{SPICLK}} = \frac{f_{\text{CLK_SPIA}}}{2 \times (\text{RLD} + 1)} \quad \text{RLD} = \frac{f_{\text{CLK_SPIA}}}{f_{\text{SPICLK}} \times 2} - 1 \quad (\text{Eq. 12.1})$$

Where

- f_{SPICLK} : SPICLK n frequency [Hz] (= baud rate [bps])
- $f_{\text{CLK_SPIA}}$: SPIA operating clock frequency [Hz]
- RLD: 16-bit timer reload data value

For controlling the 16-bit timer, refer to the “16-bit Timers” chapter.

Operating clock in slave mode

SPIA set in slave mode operates with the clock supplied from the external SPI master to the SPICLK n pin. The 16-bit timer channel (including the clock source selector and the divider) corresponding to the SPIA channel is not used. Furthermore, the SPI n MOD.NOCLKDIV bit setting becomes ineffective.

SPIA keeps operating using the clock supplied from the external SPI master even if all the internal clocks halt during SLEEP mode, so SPIA can receive data and can generate receive buffer full interrupts.

12.3.2 Clock Supply in DEBUG Mode

In master mode, the operating clock supply during DEBUG mode should be controlled using the T16 $_m$ CLK.DBRUN bit.

The CLK_T16 $_m$ supply to SPIA Ch. n is suspended when the CPU enters DEBUG mode if the T16 $_m$ CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_T16 $_m$ supply resumes. Although SPIA Ch. n stops operating when the CLK_T16 $_m$ supply is suspended, the output pins and registers retain the status before DEBUG mode was entered. If the T16 $_m$ CLK.DBRUN bit = 1, the CLK_T16 $_m$ supply is not suspended and SPIA Ch. n will keep operating in DEBUG mode.

SPIA in slave mode operates with the external SPI master clock input from the SPICLK n pin regardless of whether the CPU is placed into DEBUG mode or normal mode.

12.3.3 SPI Clock (SPICLK n) Phase and Polarity

The SPICLK n phase and polarity can be configured separately using the SPI n MOD.CPHA bit and the SPI n MOD.CPOL bit, respectively. Figure 12.3.3.1 shows the clock waveform and data input/output timing in each setting.

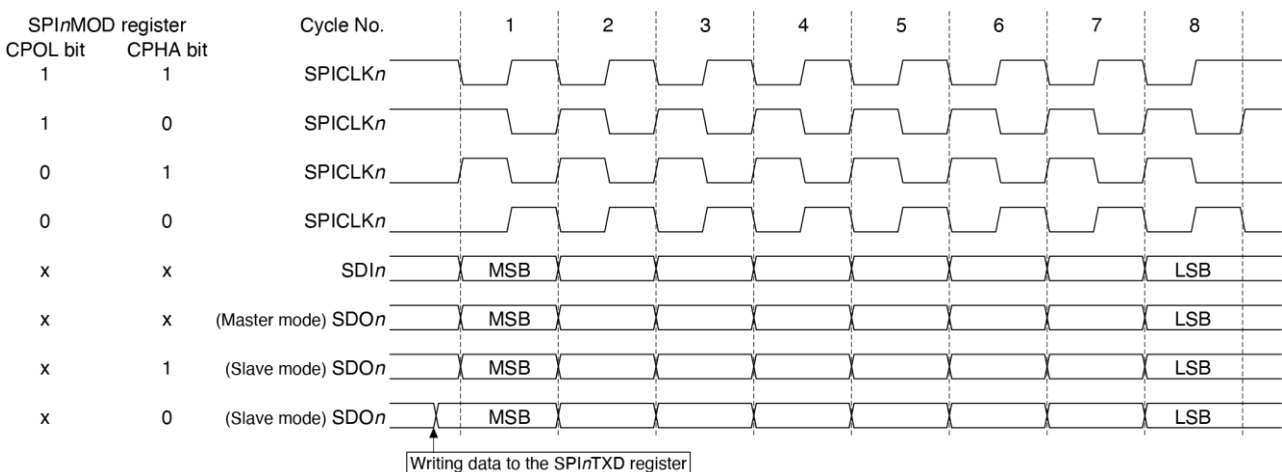


Figure 12.3.3.1 SPI Clock Phase and Polarity (SPI n MOD.LSBFST bit = 0, SPI n MOD.CHLN[3:0] bits = 0x7)

12.4 Data Format

The SPIA data length can be selected from 2 bits to 16 bits by setting the SPI n MOD.CHLN[3:0] bits. The input/output permutation is configurable to MSB first or LSB first using the SPI n MOD.LSBFST bit. Figure 12.4.1 shows a data format example when the SPI n MOD.CHLN[3:0] bits = 0x7, the SPI n MOD.CPOL bit = 0 and the SPI n MOD.CPHA bit = 0.

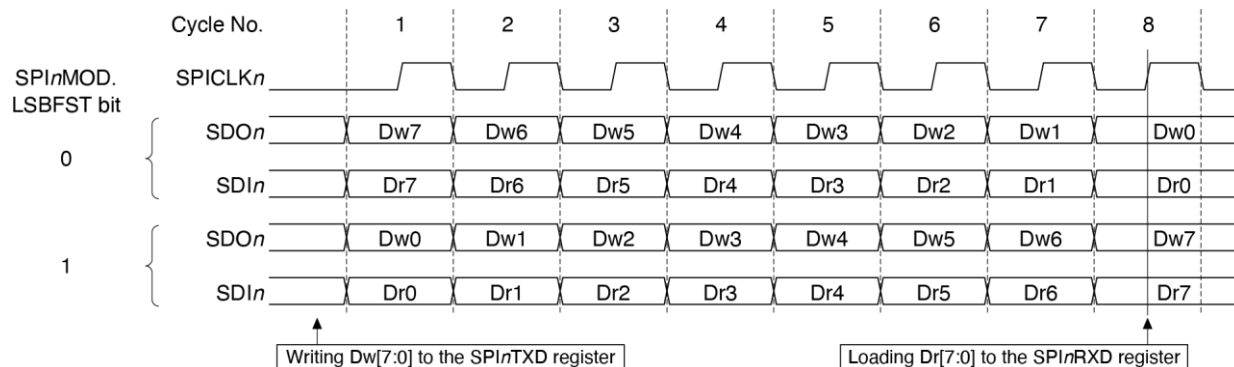


Figure 12.4.1 Data Format Selection Using the SPIInMOD.LSBFST Bit
(SPIInMOD.CHNLN[3:0] bits = 0x7, SPIInMOD.CPOL bit = 0, SPIInMOD.CPHA bit = 0)

12.5 Operations

12.5.1 Initialization

SPIA Ch.*n* should be initialized with the procedure shown below.

- <Master mode only> Generate a clock by controlling the 16-bit timer and supply it to SPIA Ch.*n*.
- Configure the following SPIInMOD register bits:
 - SPIInMOD.PUEN bit (Enable input pin pull-up/down)
 - SPIInMOD.NOCLKDIV bit (Select master mode operating clock)
 - SPIInMOD.LSBFST bit (Select MSB first/LSB first)
 - SPIInMOD.CPHA bit (Select clock phase)
 - SPIInMOD.CPOL bit (Select clock polarity)
 - SPIInMOD.MST bit (Select master/slave mode)
- Assign the SPIA Ch.*n* input/output function to the ports. (Refer to the “I/O Ports” chapter.)
- Set the following SPIInCTL register bits:
 - Set the SPIInCTL.SFTRST bit to 1. (Execute software reset)
 - Set the SPIInCTL.MODEN bit to 1. (Enable SPIA Ch.*n* operations)
- Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the SPIInINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the SPIInINTE register to 1. * (Enable interrupts)

* The initial value of the SPIInINTF.TBEIF bit is 1, therefore, an interrupt will occur immediately after the SPIInINTE.TBEIE bit is set to 1.

12.5.2 Data Transmission in Master Mode

A data sending procedure and operations in master mode are shown below. Figure 12.5.2.1 and Figure 12.5.2.2 show a timing chart and a flowchart, respectively.

Data sending procedure

- Assert the slave select signal by controlling the general-purpose output port (if necessary).
- Check to see if the SPIInINTF.TBEIF bit is set to 1 (transmit buffer empty).
- Write transmit data to the SPIInTXD register.
- Wait for an SPIA interrupt when using the interrupt.
- Repeat Steps 2 to 4 (or 2 and 3) until the end of transmit data.
- Negate the slave select signal by controlling the general-purpose output port (if necessary).

12 Synchronous Serial Interface (SPIA)

Data sending operations

SPIA Ch.*n* starts data sending operations when transmit data is written to the SPI*n*TXD register.

The transmit data in the SPI*n*TXD register is automatically transferred to the shift register and the SPI*n*INTF.TBEIF bit is set to 1. If the SPI*n*INTE.TBEIE bit = 1 (transmit buffer empty interrupt enabled), a transmit buffer empty interrupt occurs at the same time.

The SPICLK*n* pin outputs clocks of the number of the bits specified by the SPI*n*MOD.CHLN[3:0] bits and the transmit data bits are output in sequence from the SDO*n* pin in sync with these clocks.

Even if the clock is being output from the SPICLK*n* pin, the next transmit data can be written to the SPI*n*TXD register after making sure the SPI*n*INTF.TBEIF bit is set to 1.

If transmit data has not been written to the SPI*n*TXD register after the last clock is output from the SPICLK*n* pin, the clock output halts and the SPI*n*INTF.TENDIF bit is set to 1. At the same time SPIA issues an end-of-transmission interrupt request if the SPI*n*INTE.TENDIE bit = 1.

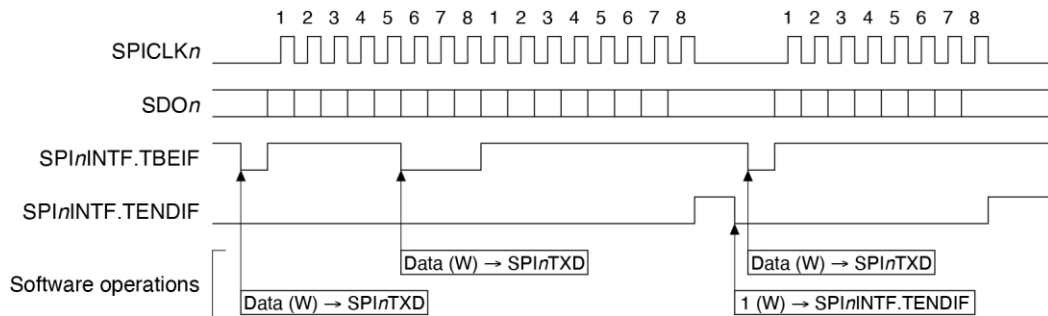


Figure 12.5.2.1 Example of Data Sending Operations in Master Mode (SPI*n*MOD.CHLN[3:0] bits = 0x7)

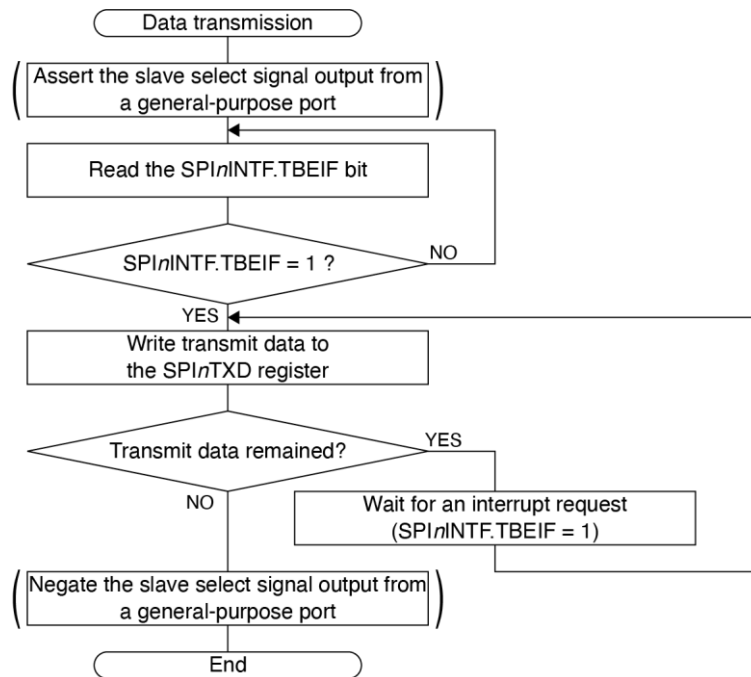


Figure 12.5.2.2 Data Transmission Flowchart in Master Mode

12.5.3 Data Reception in Master Mode

A data receiving procedure and operations in master mode are shown below. Figure 12.5.3.1 and Figure 12.5.3.2 show a timing chart and flowcharts, respectively.

Data receiving procedure

1. Assert the slave select signal by controlling the general-purpose output port (if necessary).
2. Check to see if the SPI n INTF.TBEIF bit is set to 1 (transmit buffer empty).
3. Write dummy data (or transmit data) to the SPI n TXD register.
4. Wait for a transmit buffer empty interrupt (SPI n INTF.TBEIF bit = 1).
5. Write dummy data (or transmit data) to the SPI n TXD register.
6. Wait for a receive buffer full interrupt (SPI n INTF.RBFIF bit = 1).
7. Read the received data from the SPI n RXD register.
8. Repeat Steps 5 to 7 until the end of data reception.
9. Negate the slave select signal by controlling the general-purpose output port (if necessary).

Note: To perform continuous data reception without stopping SPI CLK_n , Steps 7 and 5 operations must be completed within the SPI CLK_n cycles equivalent to “Data bit length - 1” after Step 6.

Data receiving operations

SPIA Ch. n starts data receiving operations simultaneously with data sending operations when transmit data (may be dummy data if data transmission is not required) is written to the SPI n TXD register.

The SPI CLK_n pin outputs clocks of the number of the bits specified by the SPI n MOD.CHLN[3:0] bits. The transmit data bits are output in sequence from the SDO n pin in sync with these clocks and the receive data bits input from the SDI n pin are shifted into the shift register.

When the last clock is output from the SPI CLK_n pin and receive data bits are all shifted into the shift register, the received data is transferred to the receive data buffer and the SPI n INTF.RBFIF bit is set to 1. At the same time SPIA issues a receive buffer full interrupt request if the SPI n INTE.RBFIE bit = 1. After that, the received data in the receive data buffer can be read through the SPI n RXD register.

Note: If data of the number of the bits specified by the SPI n MOD.CHLN[3:0] bits is received when the SPI n INTF.RBFIF bit is set to 1, the SPI n RXD register is overwritten with the newly received data and the previously received data is lost. In this case, the SPI n INTF.OEIF bit is set.

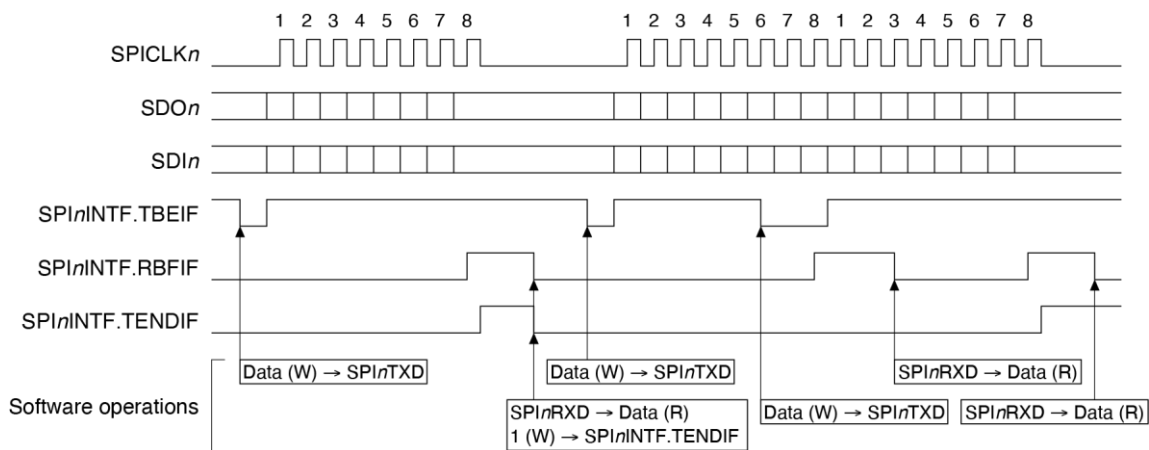


Figure 12.5.3.1 Example of Data Receiving Operations in Master Mode (SPI n MOD.CHLN[3:0] bits = 0x7)

12 Synchronous Serial Interface (SPIA)

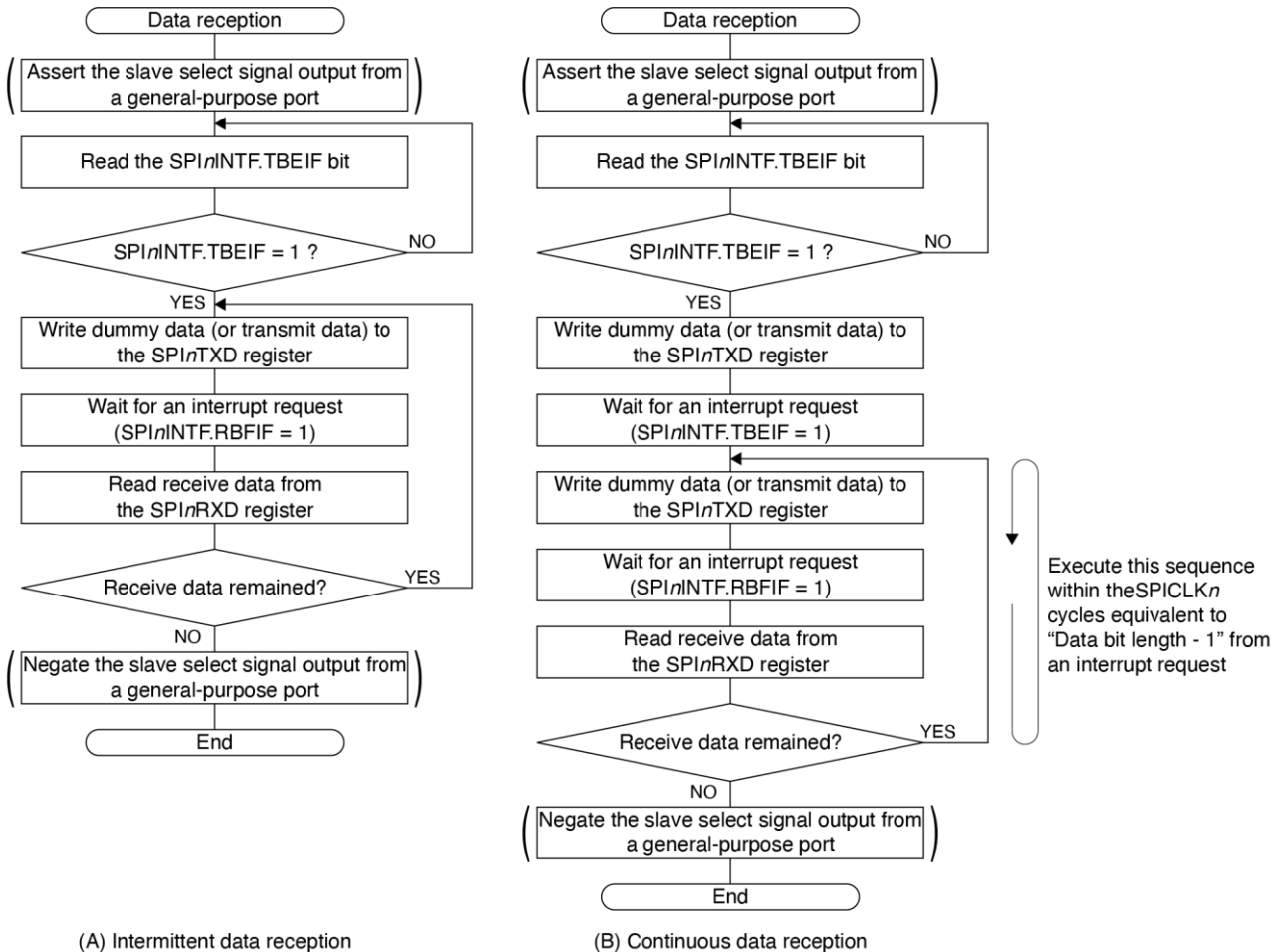


Figure 12.5.3.2 Data Reception Flowcharts in Master Mode

12.5.4 Terminating Data Transfer in Master Mode

A procedure to terminate data transfer in master mode is shown below.

1. Wait for an end-of-transmission interrupt ($SPI_nINTF.TENDIF$ bit = 1).
2. Set the $SPI_nCTL.MODEN$ bit to 0 to disable the SPIA Ch. n operations.
3. Stop the 16-bit timer to disable the clock supply to SPIA Ch. n .

12.5.5 Data Transfer in Slave Mode

A data sending/receiving procedure and operations in slave mode are shown below. Figure 12.5.5.1 and Figure 12.5.5.2 show a timing chart and flowcharts, respectively.

Data sending procedure

1. Check to see if the $SPI_nINTF.TBEIF$ bit is set to 1 (transmit buffer empty).
2. Write transmit data to the SPI_nTXD register.
3. Wait for a transmit buffer empty interrupt ($SPI_nINTF.TBEIF$ bit = 1).
4. Repeat Steps 2 and 3 until the end of transmit data.

Note: Transmit data must be written to the SPI_nTXD register after the $SPI_nINTF.TBEIF$ bit is set to 1 by the time the sending SPI_nTXD register data written is completed. If no transmit data is written during this period, the data bits input from the $SDIn$ pin are shifted and output from the $SDOn$ pin without being modified.

Data receiving procedure

1. Wait for a receive buffer full interrupt (SPI n INTF.RBFIF bit = 1).
2. Read the received data from the SPI n RXD register.
3. Repeat Steps 1 and 2 until the end of data reception.

Data transfer operations

The following shows the slave mode operations different from master mode:

- Slave mode operates with the SPI clock supplied from the external SPI master to the SPICLK n pin. The data transfer rate is determined by the SPICLK n frequency. It is not necessary to control the 16-bit timer.
- SPIA can operate as a slave device only when the slave select signal input from the external SPI master to the #SPISS n pin is set to the active (low) level.

If #SPISS n = high, the software transfer control, the SPICLK n pin input, and the SDI n pin input are all ineffective. If the #SPISS n signal goes high during data transfer, the transfer bit counter is cleared and data in the shift register is discarded.

- Slave mode starts data transfer when SPICLK n is input from the external SPI master after the #SPISS n signal is asserted. Writing transmit data is not a trigger to start data transfer. Therefore, it is not necessary to write dummy data to the transmit data buffer when performing data reception only.
- Data transmission/reception can be performed even in SLEEP mode, it makes it possible to wake the CPU up using an SPIA interrupt.

Other operations are the same as master mode.

- Notes:
- If data of the number of bits specified by the SPI n MOD.CHLN[3:0] bits is received when the SPI n INTF.RBFIF bit is set to 1, the SPI n RXD register is overwritten with the newly received data and the previously received data is lost. In this case, the SPI n INTF.OEIF bit is set.
 - When the clock for the first bit is input from the SPICLK n pin, SPIA starts sending the data currently stored in the shift register even if the SPI n INTF.TBEIF bit is set to 1.

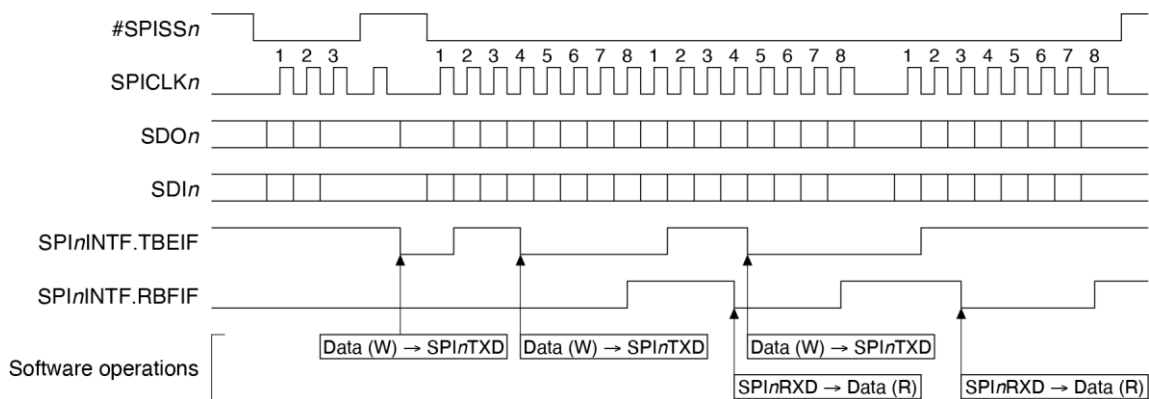


Figure 12.5.5.1 Example of Data Transfer Operations in Slave Mode (SPI n MOD.CHLN[3:0] bits = 0x7)

12 Synchronous Serial Interface (SPIA)

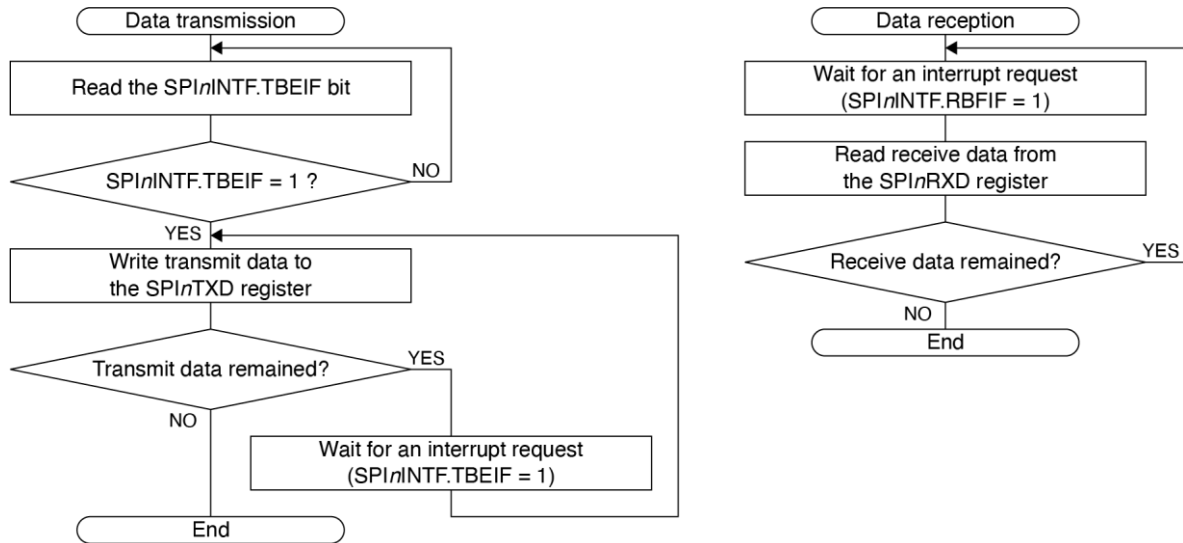


Figure 12.5.5.2 Data Transfer Flowcharts in Slave Mode

12.5.6 Terminating Data Transfer in Slave Mode

A procedure to terminate data transfer in slave mode is shown below.

1. Wait for an end-of-transmission interrupt ($SPI_nINTF.TENDIF$ bit = 1). Or determine end of transfer via the received data.
2. Set the $SPI_nCTL.MODEN$ bit to 0 to disable the SPIA Ch. n operations.

12.6 Interrupts

SPIA has a function to generate the interrupts shown in Table 12.6.1.

Table 12.6.1 SPIA Interrupt Function

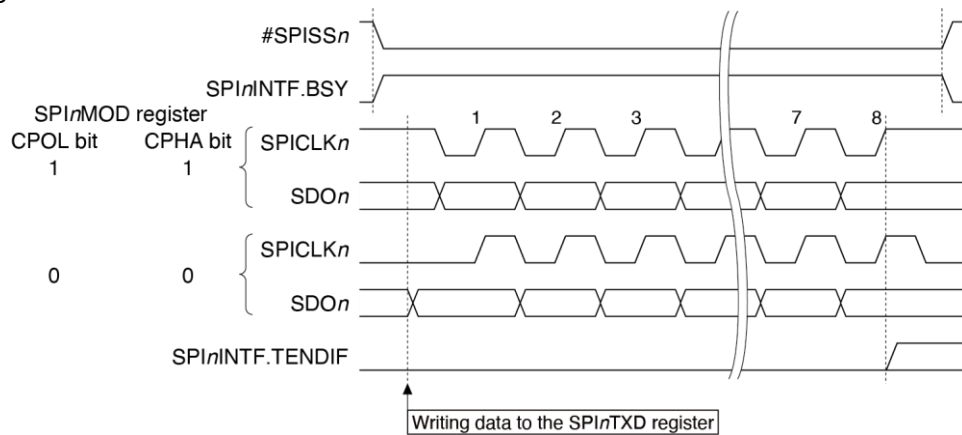
Interrupt	Interrupt flag	Set condition	Clear condition
End of transmission	$SPI_nINTF.TENDIF$	When the $SPI_nINTF.TBEIF$ bit = 1 after data of the specified bit length (defined by the $SPI_nMOD.CHLN[3:0]$ bits) has been sent	Writing 1
Receive buffer full	$SPI_nINTF.RBFIF$	When data of the specified bit length is received and the received data is transferred from the shift register to the received data buffer	Reading the SPI_nRXD register
Transmit buffer empty	$SPI_nINTF.TBEIF$	When transmit data written to the transmit data buffer is transferred to the shift register	Writing to the SPI_nTXD register
Overrun error	$SPI_nINTF.OEIF$	When the receive data buffer is full (when the received data has not been read) at the point that receiving data to the shift register has completed	Writing 1

SPIA provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

The SPI_nINTF register also contains the BSY bit that indicates the SPIA operating status.

Figure 12.6.1 shows the $SPI_nINTF.BSY$ and $SPI_nINTF.TENDIF$ bit set timings.

Master mode



Slave mode

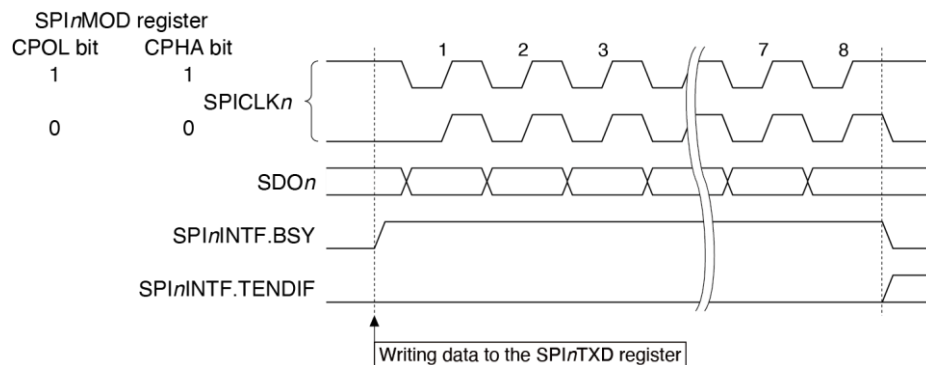


Figure 12.6.1 SPIInINTF.BSY and SPIInINTF.TENDIF Bit Set Timings (when SPIInMOD.CHLN[3:0] bits = 0x7)

12.7 Control Registers

SPIA Ch.n Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIInMOD	15–12	–	0x0	–	R	–
	11–8	CHLN[3:0]	0x7	H0	R/W	
	7–6	–	0x0	–	R	
	5	PUEN	0	H0	R/W	
	4	NOCLKDIV	0	H0	R/W	
	3	LSBFST	0	H0	R/W	
	2	CPHA	0	H0	R/W	
	1	CPOL	0	H0	R/W	
	0	MST	0	H0	R/W	

Bits 15–12 Reserved

Bits 11–8 CHLN[3:0]

These bits set the bit length of transfer data.

12 Synchronous Serial Interface (SPIA)

Table 12.7.1 Data Bit Length Settings

SPI n MOD.CHLN[3:0] bits	Data bit length
0xf	16 bits
0xe	15 bits
0xd	14 bits
0xc	13 bits
0xb	12 bits
0xa	11 bits
0x9	10 bits
0x8	9 bits
0x7	8 bits
0x6	7 bits
0x5	6 bits
0x4	5 bits
0x3	4 bits
0x2	3 bits
0x1	2 bits
0x0	Setting not allowed

Bits 7–6 **Reserved**

Bit 5 **PUEN**

This bit enables pull-up/down of the input pins.

1 (R/W): Enable pull-up/down

0 (R/W): Disable pull-up/down

For more information, refer to “Input Pin Pull-Up/Pull-Down Function.”

Bit 4 **NOCLKDIV**

This bit selects SPICLK n in master mode. This setting is ineffective in slave mode.

1 (R/W): SPICLK n frequency = CLK_SPIA n frequency (= 16-bit timer operating clock frequency)

0 (R/W): SPICLK n frequency = 16-bit timer output frequency / 2

For more information, refer to “SPIA Operating Clock.”

Bit 3 **LSBFST**

This bit configures the data format (input/output permutation).

1 (R/W): LSB first

0 (R/W): MSB first

Bit 2 **CPHA**

Bit 1 **CPOL**

These bits set the SPI clock phase and polarity. For more information, refer to “SPI Clock (SPICLK n) Phase and Polarity.”

Bit 0 **MST**

This bit sets the SPIA operating mode (master mode or slave mode).

1 (R/W): Master mode

0 (R/W): Slave mode

Note: The SPI n MOD register settings can be altered only when the SPI n CTL.MODEN bit = 0.

SPIA Ch. n Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI n CTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	SFTRST	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–2 **Reserved**

Bit 1 SFTRST

This bit issues software reset to SPIA.

1 (W): Issue software reset

0 (W): Ineffective

1 (R): Software reset is executing.

0 (R): Software reset has finished. (During normal operation)

Setting this bit resets the SPIA shift register and transfer bit counter. This bit is automatically cleared after the reset processing has finished.

Bit 0 MODEN

This bit enables the SPIA operations.

1 (R/W): Enable SPIA operations (In master mode, the operating clock is supplied.)

0 (R/W): Disable SPIA operations (In master mode, the operating clock is stopped.)

Note: If the `SPI n CTL.MODEN` bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the `SPI n CTL.MODEN` bit to 1 again after that, be sure to write 1 to the `SPI n CTL.SFTRST` bit as well.

SPIA Ch. n Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
<code>SPInTXD</code>	15–0	<code>TXD[15:0]</code>	0x0000	H0	R/W	–

Bits 15–0 TXD[15:0]

Data can be written to the transmit data buffer through these bits.

In master mode, writing to these bits starts data transfer.

Transmit data can be written when the `SPI n INTF.TBEIF` bit = 1 regardless of whether data is being output from the `SDO n` pin or not.

Note that the upper data bits that exceed the data bit length configured by the `SPI n MOD.CHLN[3:0]` bits will not be output from the `SDO n` pin.

Note: Be sure to avoid writing to the `SPI n TXD` register when the `SPI n INTF.TBEIF` bit = 0. Otherwise, transfer data cannot be guaranteed.

SPIA Ch. n Receive Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
<code>SPInRXD</code>	15–0	<code>RXD[15:0]</code>	0x0000	H0	R	–

Bits 15–0 RXD[15:0]

The receive data buffer can be read through these bits. Received data can be read when the `SPI n INTF.RBFIF` bit = 1 regardless of whether data is being input from the `SDI n` pin or not. Note that the upper bits that exceed the data bit length configured by the `SPI n MOD.CHLN[3:0]` bits become 0.

Note: The `SPI n RXD.RXD[15:0]` bits are cleared to 0x0000 when 1 is written to the `SPI n CTL.MODEN` bit or the `SPI n CTL.SFTRST` bit.

SPIA Ch. n Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
<code>SPInINTF</code>	15–8	–	0x00	–	R	–
	7	<code>BSY</code>	0	H0	R	–
	6–4	–	0x0	–	R	–
	3	<code>OEIF</code>	0	H0/S0	R/W	Cleared by writing 1.
	2	<code>TENDIF</code>	0	H0/S0	R/W	–
	1	<code>RBFIF</code>	0	H0/S0	R	Cleared by reading the <code>SPInRXD</code> register.
	0	<code>TBEIF</code>	1	H0/S0	R	Cleared by writing to the <code>SPInTXD</code> register.

Bits 15–8 Reserved

12 Synchronous Serial Interface (SPIA)

Bit 7 BSY

This bit indicates the SPIA operating status.

1 (R): Transmit/receive busy (master mode), #SPISS n = Low level (slave mode)

0 (R): Idle

Bits 6–4 Reserved

Bit 3 OEIF

Bit 2 TENDIF

Bit 1 RBFIF

Bit 0 TBEIF

These bits indicate the SPIA interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag (OEIF, TENDIF)

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

SPI n INTF.OEIF bit: Overrun error interrupt

SPI n INTF.TENDIF bit: End-of-transmission interrupt

SPI n INTF.RBFIF bit: Receive buffer full interrupt

SPI n INTF.TBEIF bit: Transmit buffer empty interrupt

SPIA Ch. n Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI n INTE	15–8	–	0x00	–	R	–
	7–4	–	0x0	–	R	
	3	OEIE	0	H0	R/W	
	2	TENDIE	0	H0	R/W	
	1	RBFIE	0	H0	R/W	
	0	TBEIE	0	H0	R/W	

Bits 15–4 Reserved

Bit 3 OEIE

Bit 2 TENDIE

Bit 1 RBFIE

Bit 0 TBEIE

These bits enable SPIA interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

SPI n INTE.OEIE bit: Overrun error interrupt

SPI n INTE.TENDIE bit: End-of-transmission interrupt

SPI n INTE.RBFIE bit: Receive buffer full interrupt

SPI n INTE.TBEIE bit: Transmit buffer empty interrupt

13 I²C (I2C)

13.1 Overview

The I²C is a subset of the I²C bus interface. The features of the I²C are listed below.

- Functions as an I²C bus master (single master) or a slave device.
- Supports standard mode (up to 100 kbit/s) and fast mode (up to 400 kbit/s).
- Supports 7-bit and 10-bit address modes.
- Supports clock stretching.
- Includes a baud rate generator for generating the clock in master mode.
- No clock source is required to run the I²C in slave mode, as it can run with the I²C bus signals only.
- Slave mode is capable of being operated in SLEEP mode allowing wake-up by an interrupt when an address match is detected.
- Master mode supports automatic bus clear sending function.
- Can generate receive buffer full, transmit buffer empty, and other interrupts.
- The input filter for the SDA and SCL inputs does not comply with the standard for removing noise spikes less than 50 ns.

Figure 13.1.1 shows the I²C configuration.

Table 13.1.1 I²C Channel Configuration of S1C17W11

Item	S1C17W11
Number of channels	1 channel (Ch.0)

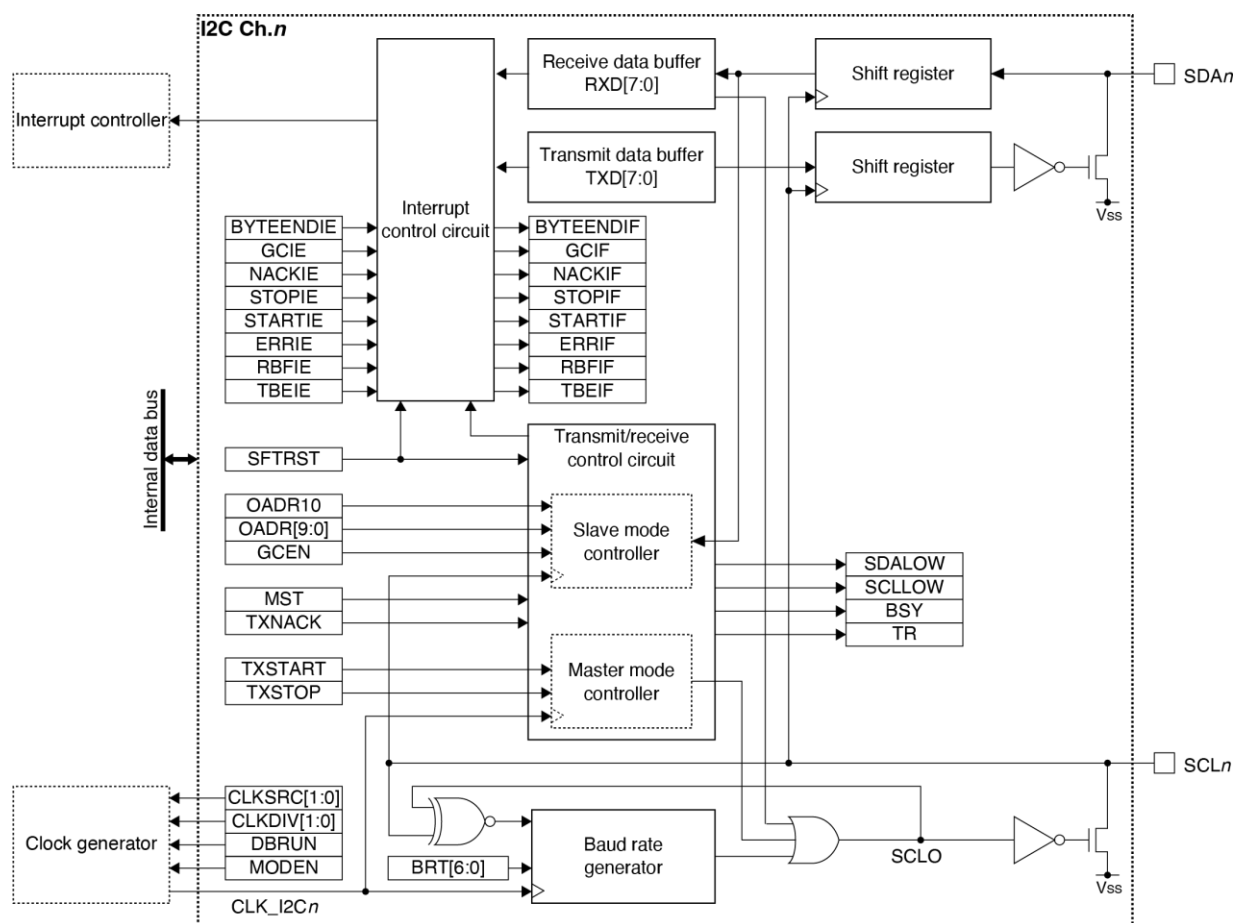


Figure 13.1.1 I²C Configuration

13.2 Input/Output Pins and External Connections

13.2.1 List of Input/Output Pins

Table 13.2.1.1 lists the I2C pins.

Table 13.2.1.1 List of I2C Pins

Pin name	I/O*	Initial status*	Function
SDAn	I/O	I	I ² C bus serial data input/output pin
SCLn	I/O	I	I ² C bus clock input/output pin

* Indicates the status when the pin is configured for the I2C.

If the port is shared with the I2C pin and other functions, the I2C input/output function must be assigned to the port before activating the I2C. For more information, refer to the “I/O Ports” chapter.

13.2.2 External Connections

Figure 13.2.2.1 shows a connection diagram between the I2C in this IC and external I²C devices.

The serial data (SDA) and serial clock (SCL) lines must be pulled up with an external resistor.

When the I2C is set into master mode, one or more slave devices that have a unique address may be connected to the I²C bus. When the I2C is set into slave mode, one or more master and slave devices that have a unique address may be connected to the I²C bus.

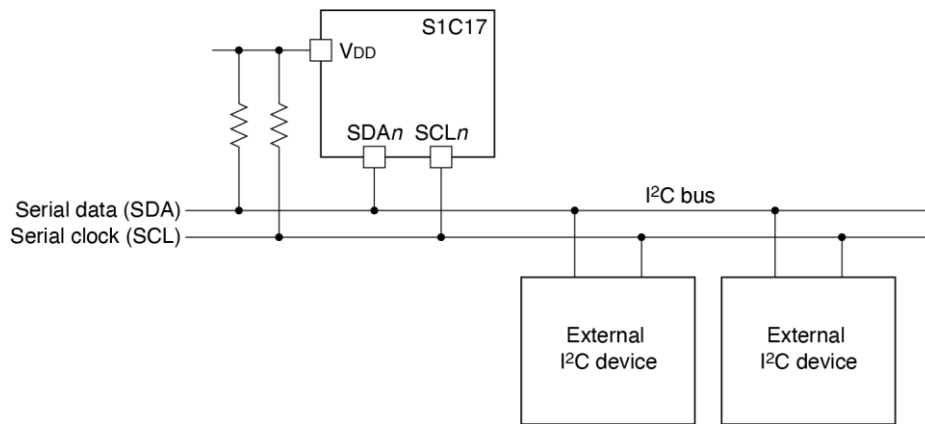


Figure 13.2.2.1 Connections between I2C and External I²C Devices

- Notes:
- The SDA and SCL lines must be pulled up to a V_{DD} of this IC or lower voltage. However, if the I2C input/output ports are configured with the over voltage tolerant fail-safe type I/O, these lines can be pulled up to a voltage exceeding the V_{DD} of this IC but within the recommended operating voltage range of this IC.
 - The internal pull-up resistors for the I/O ports cannot be used for pulling up SDA and SCL.
 - When the I2C is set into master mode, no other master device can be connected to the I²C bus.

13.3 Clock Settings

13.3.1 I2C Operating Clock

Master mode operating clock

When using the I2C Ch.*n* in master mode, the I2C Ch.*n* operating clock CLK_I2C*n* must be supplied to the I2C Ch.*n* from the clock generator. The CLK_I2C*n* supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
2. Set the following I2CnCLK register bits:
 - I2CnCLK.CLKSRC[1:0] bits (Clock source selection)
 - I2CnCLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)

When using the I2C in master mode during SLEEP mode, the I2C Ch.*n* operating clock CLK_I2C*n* must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_I2C*n* clock source.

The I2C operating clock should be selected so that the baud rate generator will be configured easily.

Slave mode operating clock

The I2C set to slave mode uses the SCL supplied from the I²C master as its operating clock. The clock setting by the I2CnCLK register is ineffective.

The I2C keeps operating using the clock supplied from the external I²C master even if all the internal clocks halt during SLEEP mode, so the I2C can receive data and can generate receive buffer full interrupts.

13.3.2 Clock Supply in DEBUG Mode

In master mode, the CLK_I2C*n* supply during DEBUG mode should be controlled using the I2CnCLK.DBRUN bit. The CLK_I2C*n* supply to the I2C Ch.*n* is suspended when the CPU enters DEBUG mode if the I2CnCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_I2C*n* supply resumes. Although the I2C Ch.*n* stops operating when the CLK_I2C*n* supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the I2CnCLK.DBRUN bit = 1, the CLK_I2C*n* supply is not suspended and the I2C Ch.*n* will keep operating in DEBUG mode.

In slave mode, the I2C Ch.*n* operates with the external I²C master clock input from the SCL*n* pin regardless of whether the CPU is placed into DEBUG mode or normal mode.

13.3.3 Baud Rate Generator

The I2C includes a baud rate generator to generate the serial clock SCL used in master mode. The I2C set to slave mode does not use the baud rate generator, as it operates with the serial clock input from the SCL*n* pin.

Setting data transfer rate (for master mode)

The transfer rate is determined by the I2CnBR.BRT[6:0] bit settings. Use the following equations to calculate the setting values for obtaining the desired transfer rate.

$$\text{bps} = \frac{f_{\text{CLK_I2C}n}}{(\text{BRT} + 3) \times 2} \quad \text{BRT} = \frac{f_{\text{CLK_I2C}n}}{\text{bps} \times 2} - 3 \quad (\text{Eq. 13.1})$$

Where

bps: Data transfer rate [bit/s]
 $f_{\text{CLK_I2C}n}$: I2C operating clock frequency [Hz]
 BRT: I2CnBR.BRT[6:0] bits setting value (1 to 127)

* The equations above do not include SCL rising/falling time and delay time by clock stretching (see Figure 13.3.3.1).

Note: The I²C bus transfer rate is limited to 100 kbit/s in standard mode or 400 kbit/s in fast mode. Do not set a transfer rate exceeding the limit.

Baud rate generator clock output and operations for supporting clock stretching

Figure 13.3.3.1 shows the clock generated by the baud rate generator and the clock waveform on the I²C bus.

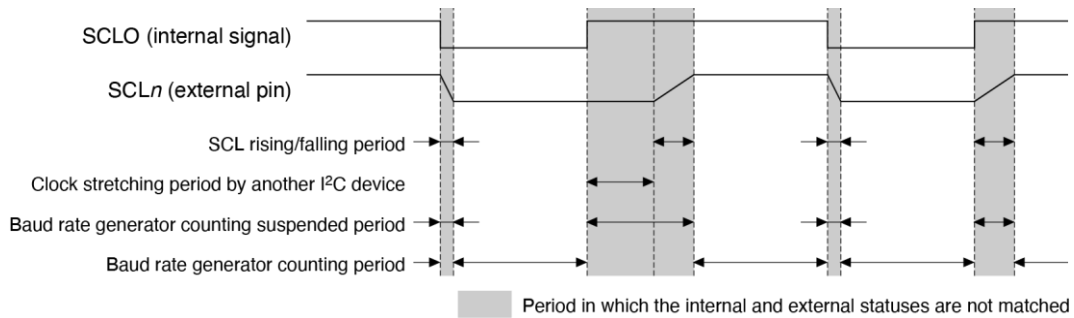


Figure 13.3.3.1 Baud Rate Generator Output Clock and SCL_n Output Waveform

The baud rate generator output clock SCLO is compared with the SCL_n pin status and the results are returned to the baud rate generator. If a mismatch has occurred between SCLO and SCL_n pin levels, the baud rate generator suspends counting. This extends the clock to control data transfer during the SCL signal rising/falling period and clock stretching period in which SCL is fixed at low by a slave device.

13.4 Operations

13.4.1 Initialization

The I2C Ch.*n* should be initialized with the procedure shown below.

When using the I2C in master mode

1. Configure the operating clock and the baud rate generator using the I2C_nCLK and I2C_nBR registers.
2. Assign the I2C Ch.*n* input/output function to the ports. (Refer to the “I/O Ports” chapter.)
3. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the I2C_nINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the I2C_nINTE register to 1. (Enable interrupts)
4. Set the following I2C_nCTL register bits:
 - Set the I2C_nCTL.MST bit to 1. (Set master mode)
 - Set the I2C_nCTL.SFTRST bit to 1. (Execute software reset)
 - Set the I2C_nCTL.MODEN bit to 1. (Enable I2C Ch.*n* operations)

When using the I2C in slave mode

1. Set the following I2C_nMOD register bits:
 - I2C_nMOD.OADR10 bit (Set 10/7-bit address mode)
 - I2C_nMOD.GCEN bit (Enable response to general call address)
2. Set its own address to the I2C_nOADR.OADR[9:0] (or OADR[6:0]) bits.
3. Assign the I2C Ch.*n* input/output function to the ports. (Refer to the “I/O Ports” chapter.)
4. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the I2C_nINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the I2C_nINTE register to 1. (Enable interrupts)
5. Set the following I2C_nCTL register bits:
 - Set the I2C_nCTL.MST bit to 0. (Set slave mode)
 - Set the I2C_nCTL.SFTRST bit to 1. (Execute software reset)
 - Set the I2C_nCTL.MODEN bit to 1. (Enable I2C Ch.*n* operations)

13.4.2 Data Transmission in Master Mode

A data sending procedure in master mode and the I2C Ch.*n* operations are shown below. Figure 13.4.2.1 and Figure 13.4.2.2 show an operation example and a flowchart, respectively.

Data sending procedure

1. Issue a START condition by setting the I2CnCTL.TXSTART bit to 1.
2. Wait for a transmit buffer empty interrupt (I2CnINTF.TBEIF bit = 1) or a START condition interrupt (I2CnINTF.STARTIF bit = 1).
Clear the I2CnINTF.STARTIF bit by writing 1 after the interrupt has occurred.
3. Write the 7-bit slave address to the I2CnTXD.TXD[7:1] bits and 0 that represents WRITE as the data transfer direction to the I2CnTXD.TXD0 bit.
4. Wait for a transmit buffer empty interrupt (I2CnINTF.TBEIF bit = 1) generated when an ACK is received or a NACK reception interrupt (I2CnINTF.NACKIF bit = 1) generated when a NACK is received.
 - i. Go to Step 5 if transmit data remains when a transmit buffer empty interrupt has occurred.
 - ii. Go to Step 7 or 1 after clearing the I2CnINTF.NACKIF bit when a NACK reception interrupt has occurred.
5. Write transmit data to the I2CnTXD register.
6. Repeat Steps 4 and 5 until the end of transmit data.
7. Issue a STOP condition by setting the I2CnCTL.TXSTOP bit to 1.
8. Wait for a STOP condition interrupt (I2CnINTF.STOPIF bit = 1).
Clear the I2CnINTF.STOPIF bit by writing 1 after the interrupt has occurred.

Data sending operations

Generating a START condition

The I2C Ch.*n* starts generating a START condition when the I2CnCTL.TXSTART bit is set to 1. When the generating operation has completed, the I2C Ch.*n* clears the I2CnCTL.TXSTART bit to 0 and sets both the I2CnINTF.STARTIF and I2CnINTF.TBEIF bits to 1.

Sending slave address and data

If the I2CnINTF.TBEIF bit = 1, a slave address or data can be written to the I2CnTXD register. The I2C Ch.*n* pulls down SCL to low and enters standby state until data is written to the I2CnTXD register. The writing operation triggers the I2C Ch.*n* to send the data to the shift register automatically and to output eight clock pulses and data bits to the I²C bus.

When the slave device returns an ACK as the response, the I2CnINTF.TBEIF bit is set to 1. After this interrupt occurs, the subsequent data may be sent or a STOP/repeated START condition may be issued to terminate transmission. If the slave device returns NACK, the I2CnINTF.NACKIF bit is set to 1 without setting the I2CnINTF.TBEIF bit.

Generating a STOP/repeated START condition

After the I2CnINTF.TBEIF bit is set to 1 (transmit buffer empty) or the I2CnINTF.NACKIF bit is set to 1 (NACK received), setting the I2CnCTL.TXSTOP bit to 1 generates a STOP condition. When the bus free time (t_{BUF} defined in the I2C Specifications) has elapsed after the STOP condition has been generated, the I2CnCTL.TXSTOP bit is cleared to 0 and the I2CnINTF.STOPIF bit is set to 1.

When setting the I2CnCTL.TXSTART bit to 1 while the I2CnINTF.TBEIF bit = 1 (transmit buffer empty) or the I2CnINTF.NACKIF bit = 1 (NACK received), the I2C Ch.*n* generates a repeated START condition. When the repeated START condition has been generated, the I2CnINTF.STARTIF and I2CnINTF.TBEIF bits are both set to 1 same as when a START condition has been generated.

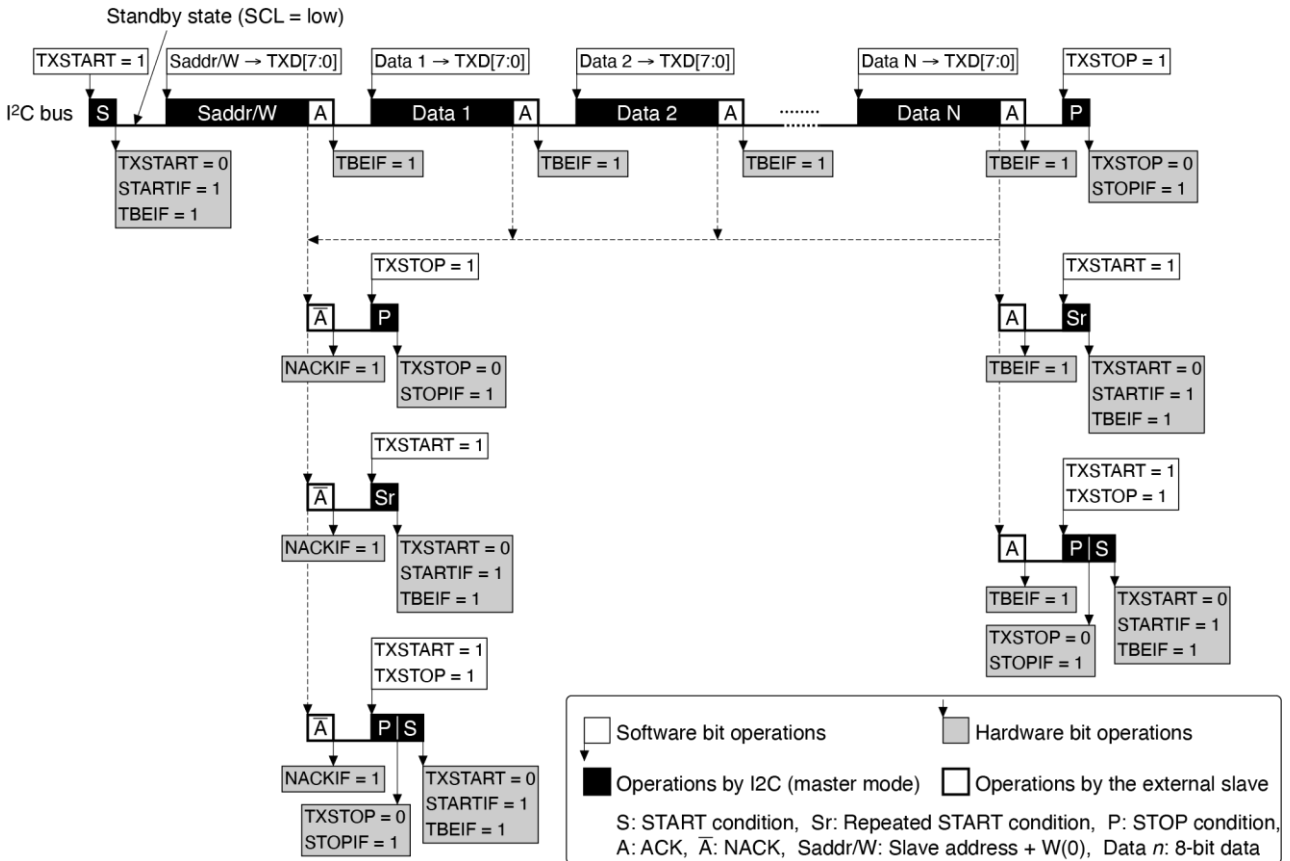


Figure 13.4.2.1 Example of Data Sending Operations in Master Mode

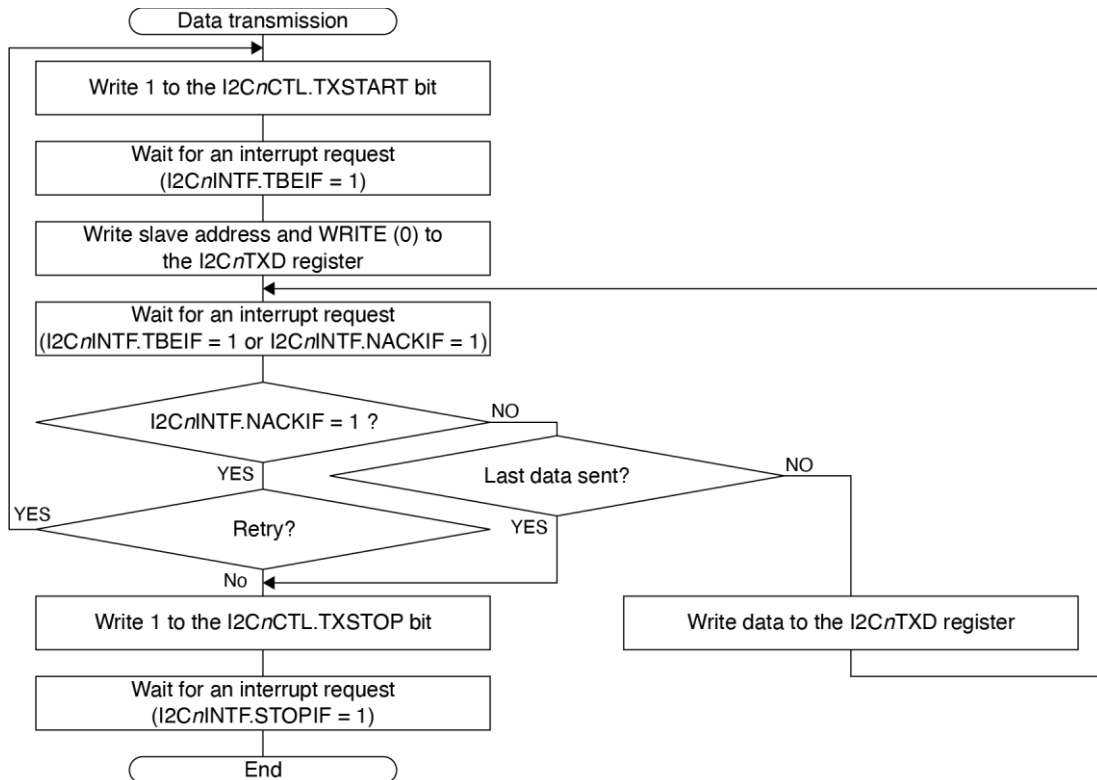


Figure 13.4.2.2 Master Mode Data Transmission Flowchart

13.4.3 Data Reception in Master Mode

A data receiving procedure in master mode and the I2C Ch.*n* operations are shown below. Figure 13.4.3.1 and Figure 13.4.3.2 show an operation example and a flowchart, respectively.

Data receiving procedure

1. When receiving one-byte data, write 1 to the I2C*n*CTL.TXNACK bit.
2. Issue a START condition by setting the I2C*n*CTL.TXSTART bit to 1.
3. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1) or a START condition interrupt (I2C*n*INTF.STARTIF bit = 1).
Clear the I2C*n*INTF.STARTIF bit by writing 1 after the interrupt has occurred.
4. Write the 7-bit slave address to the I2C*n*TXD.TXD[7:1] bits and 1 that represents READ as the data transfer direction to the I2C*n*TXD.TXD0 bit.
5. Wait for a receive buffer full interrupt (I2C*n*INTF.RBFIF bit = 1) generated when a one-byte reception has completed or a NACK reception interrupt (I2C*n*INTF.NACKIF bit = 1) generated when a NACK is received.
 - i. Go to Step 6 when a receive buffer full interrupt has occurred.
 - ii. Clear the I2C*n*INTF.NACKIF bit and issue a STOP condition by setting the I2C*n*CTL.TXSTOP bit to 1 when a NACK reception interrupt has occurred. Then go to Step 9 or Step 2 if making a retry.
6. Perform one of the operations below when the last or next-to-last data is received.
 - i. When the next-to-last data is received, write 1 to the I2C*n*CTL.TXNACK bit to send a NACK after the last data is received, and then go to Step 7.
 - ii. When the last data is received, read the received data from the I2C*n*RXD register and set the I2C*n*CTL.TXSTOP to 1 to generate a STOP condition. Then go to Step 9.
7. Read the received data from the I2C*n*RXD register.
8. Repeat Steps 5 to 7 until the end of data reception.
9. Wait for a STOP condition interrupt (I2C*n*INTF.STOPIF bit = 1).
Clear the I2C*n*INTF.STOPIF bit by writing 1 after the interrupt has occurred.

Data receiving operations

Generating a START condition

It is the same as the data transmission in master mode.

Sending slave address

It is the same as the data transmission in master mode. Note, however, that the I2C*n*TXD.TXD0 bit must be set to 1 that represents READ as the data transfer direction to issue a request to the slave to send data.

Receiving data

After the slave address has been sent, the slave device sends an ACK and the first data. The I2C Ch.*n* sets the I2C*n*INTF.RBFIF bit to 1 after the data reception has completed. Furthermore, the I2C Ch.*n* returns an ACK. To return a NACK, such as for a response after the last data has been received, write 1 to the I2C*n*CTL.TXNACK bit before the I2C*n*INTF.RBFIF bit is set to 1.

The received data can be read out from the I2C*n*RXD register after a receive buffer full interrupt has occurred. The I2C Ch.*n* pulls down SCL to low and enters standby state until data is read out from the I2C*n*RXD register. This reading triggers the I2C Ch.*n* to start subsequent data reception.

Generating a STOP or repeated START condition

It is the same as the data transmission in master mode.

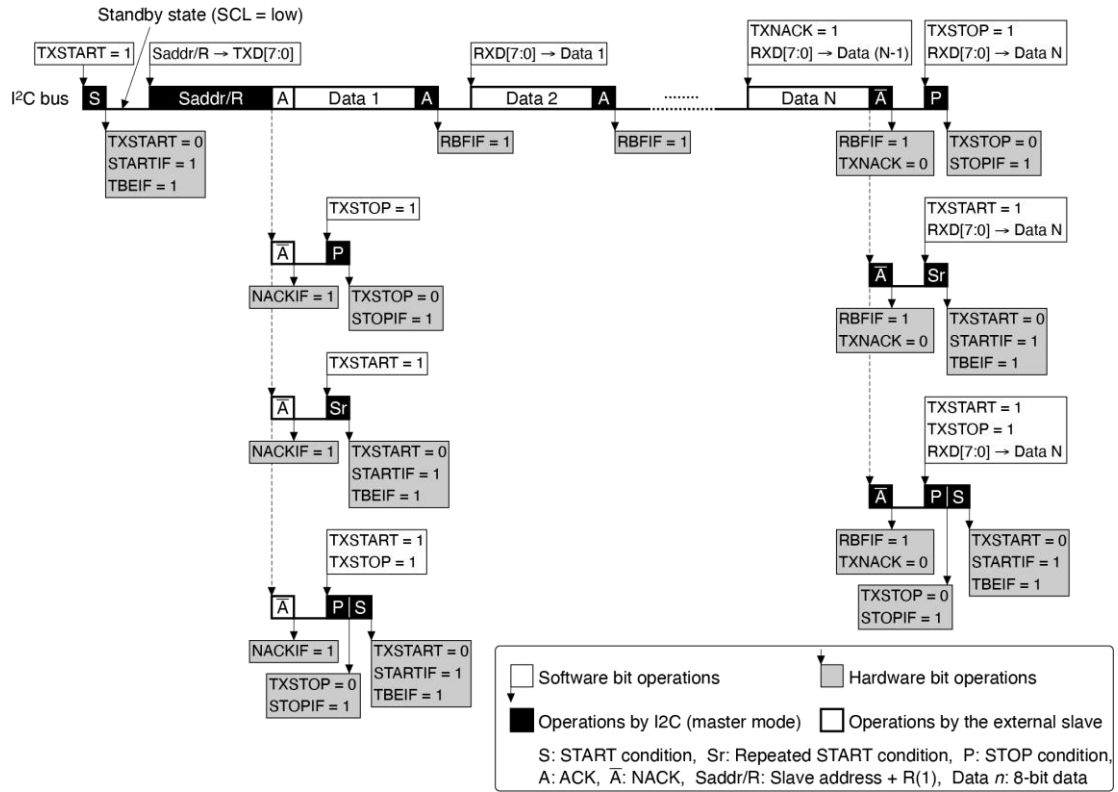


Figure 13.4.3.1 Example of Data Receiving Operations in Master Mode

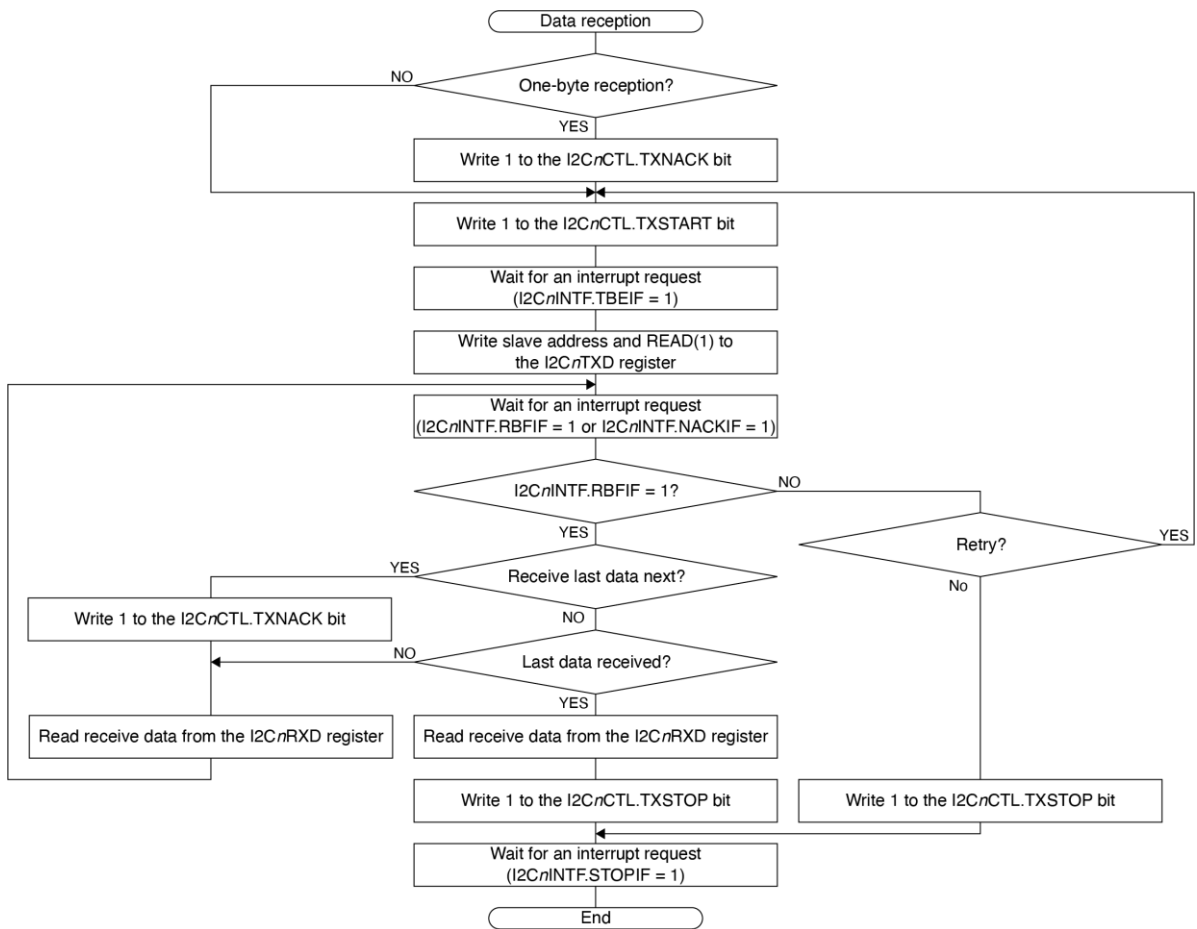


Figure 13.4.3.2 Master Mode Data Reception Flowchart

13.4.4 10-bit Addressing in Master Mode

A 10-bit address consists of the first address that contains two high-order bits and the second address that contains eight low-order bits.



Figure 13.4.4.1 10-bit Address Configuration

The following shows a procedure to start data transfer in 10-bit address mode when the I2C Ch.*n* is placed into master mode (see the 7-bit mode descriptions above for control procedures when a NACK is received or sending/receiving data). Figure 13.4.4.2 shows an operation example.

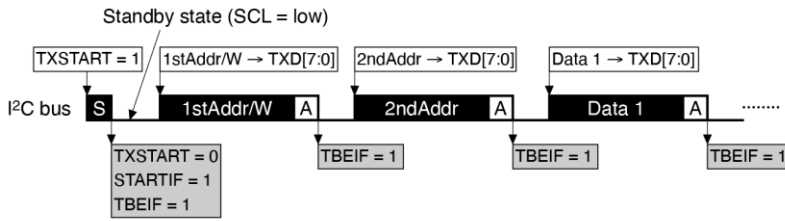
Starting data transmission in 10-bit address mode

1. Issue a START condition by setting the I2CnCTL.TXSTART bit to 1.
2. Wait for a transmit buffer empty interrupt (I2CnINTF.TBEIF bit = 1) or a START condition interrupt (I2CnINTF.STARTIF bit = 1).
Clear the I2CnINTF.STARTIF bit by writing 1 after the interrupt has occurred.
3. Write the first address to the I2CnTXD.TXD[7:1] bits and 0 that represents WRITE as the data transfer direction to the I2CnTXD.TXD0 bit.
4. Wait for a transmit buffer empty interrupt (I2CnINTF.TBEIF bit = 1).
5. Write the second address to the I2CnTXD.TXD[7:0] bits.
6. Wait for a transmit buffer empty interrupt (I2CnINTF.TBEIF bit = 1).
7. Perform data transmission.

Starting data reception in 10-bit address mode

- 1 to 6. These steps are the same as the data transmission starting procedure described above.
7. Issue a repeated START condition by setting the I2CnCTL.TXSTART bit to 1.
8. Wait for a transmit buffer empty interrupt (I2CnINTF.TBEIF bit = 1) or a START condition interrupt (I2CnINTF.STARTIF bit = 1).
Clear the I2CnINTF.STARTIF bit by writing 1 after the interrupt has occurred.
9. Write the first address to the I2CnTXD.TXD[7:1] bits and 1 that represents READ as the data transfer direction to the I2CnTXD.TXD0 bit.
10. Perform data reception.

At start of data transmission



At start of data reception

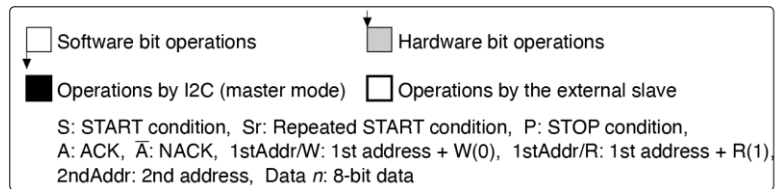
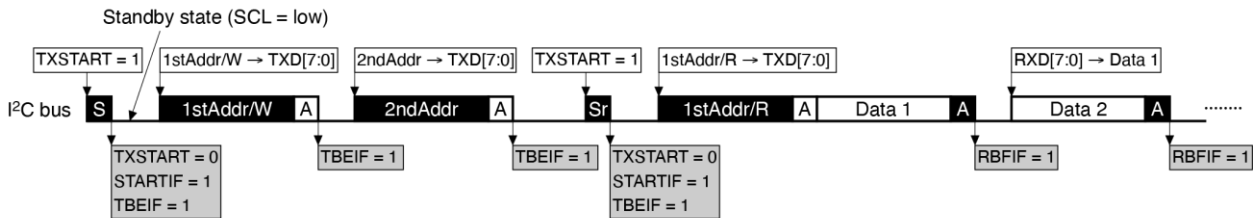


Figure 13.4.4.2 Example of Data Transfer Starting Operations in 10-bit Address Mode (Master Mode)

13.4.5 Data Transmission in Slave Mode

A data sending procedure in slave mode and the I2C Ch.n operations are shown below. Figure 13.4.5.1 and Figure 13.4.5.2 show an operation example and a flowchart, respectively.

Data sending procedure

1. Wait for a START condition interrupt (I2CnINTF.STARTIF bit = 1).
Clear the I2CnINTF.STARTIF bit by writing 1 after the interrupt has occurred.
2. Check to see if the I2CnINTF.TR bit = 1 (transmission mode).
(Start a data receiving procedure if the I2CnINTF.TR bit = 0.)
3. Write transmit data to the I2CnTXD register.
4. Wait for a transmit buffer empty interrupt (I2CnINTF.TBEIF bit = 1), a NACK reception interrupt (I2CnINTF.NACKIF bit = 1), or a STOP condition interrupt (I2CnINTF.STOPIF bit = 1).
 - i. Go to Step 3 when a transmit buffer empty interrupt has occurred.
 - ii. Go to Step 5 after clearing the I2CnINTF.NACKIF bit when a NACK reception interrupt has occurred.
 - iii. Go to Step 6 when a STOP condition interrupt has occurred.
5. Wait for a STOP condition interrupt (I2CnINTF.STOPIF bit = 1) or a START condition interrupt (I2CnINTF.STARTIF bit = 1).
 - i. Go to Step 6 when a STOP condition interrupt has occurred.
 - ii. Go to Step 2 when a START condition interrupt has occurred.
6. Clear the I2CnINTF.STOPIF bit and then terminate data sending operations.

Data sending operations

START condition detection and slave address check

While the I2CnCTL.MODEN bit = 1 and the I2CnCTL.MST bit = 0 (slave mode), the I2C Ch.n monitors the I²C bus. When the I2C Ch.n detects a START condition, it starts receiving of the slave address sent from the master. If the received address is matched with the own address set to the I2CnOADR.OADR[6:0] bits (when the I2CnMOD.OADR10 bit = 0 (7-bit address mode)) or the I2CnOADR.OADR[9:0] bits (when the I2CnMOD.OADR10 bit = 1 (10-bit address mode)), the I2CnINTF.STARTIF bit and the I2CnINTF.BSY bit are both set to 1. The I2C Ch.n sets the I2CnINTF.TR bit to the R/W bit value in the received address. If this value is 1, the I2C Ch.n sets the I2CnINTF.TBEIF bit to 1 and starts data sending operations.

Sending the first data byte

After the valid slave address has been received, the I2C Ch.n pulls down SCL to low and enters standby state until data is written to the I2CnTXD register. This puts the I²C bus into clock stretching state and the external master into standby state. When transmit data is written to the I2CnTXD register, the I2C Ch.n clears the I2CnINTF.TBEIF bit and sends an ACK to the master. The transmit data written in the I2CnTXD register is automatically transferred to the shift register and the I2CnINTF.TBEIF bit is set to 1. The data bits in the shift register are output in sequence to the I²C bus.

Sending subsequent data

If the I2CnINTF.TBEIF bit = 1, subsequent transmit data can be written during data transmission. If the I2CnINTF.TBEIF bit is still set to 1 when the data transmission from the shift register has completed, the I2C Ch.n pulls down SCL to low (sets the I²C bus into clock stretching state) until transmit data is written to the I2CnTXD register.

If the next transmit data already exists in the I2CnTXD register or data has been written after the above, the I2C Ch.n sends the subsequent eight-bit data when an ACK from the external master is received. At the same time, the I2CnINTF.BYTEENDIF bit is set to 1. If a NACK is received, the I2CnINTF.NACKIF bit is set to 1 without sending data.

STOP/repeated START condition detection

While the I2CnCTL.MST bit = 0 (slave mode) and the I2CnINTF.BSY = 1, the I2C Ch.n monitors the I²C bus. When the I2C Ch.n detects a STOP condition, it terminates data sending operations. At this time, the I2CnINTF.BSY bit is cleared to 0 and the I2CnINTF.STOPIF bit is set to 1. Also when the I2C Ch.n detects a repeated START condition, it terminates data sending operations. In this case, the I2CnINTF.STARTIF bit is set to 1.

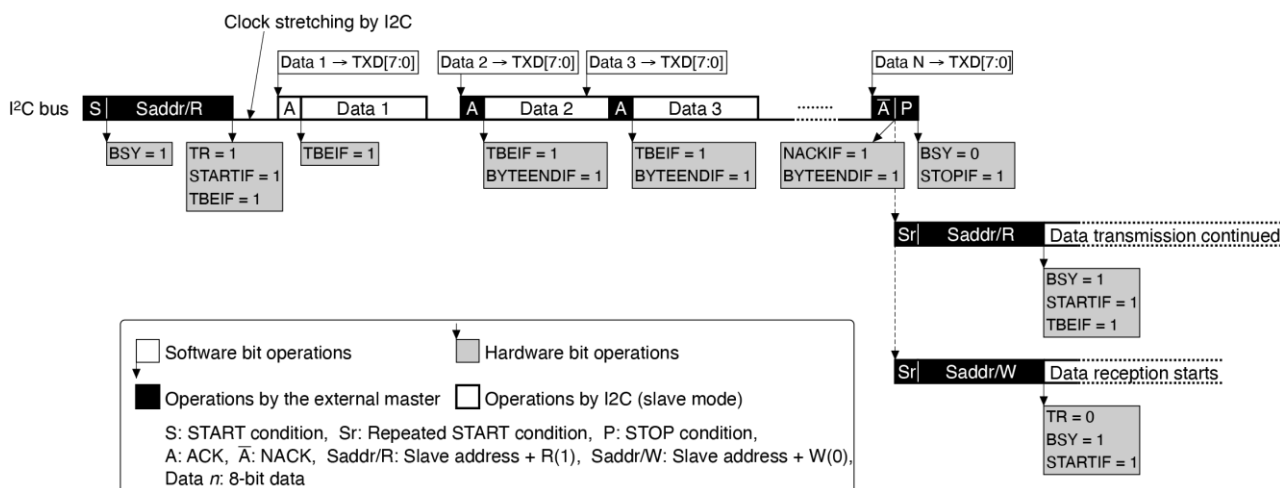


Figure 13.4.5.1 Example of Data Sending Operations in Slave Mode

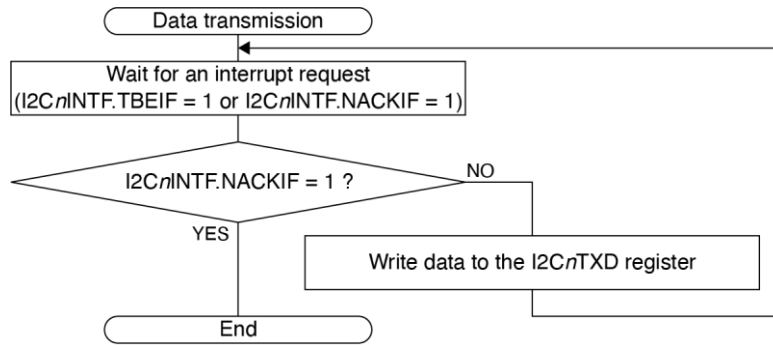


Figure 13.4.5.2 Slave Mode Data Transmission Flowchart

13.4.6 Data Reception in Slave Mode

A data receiving procedure in slave mode and the I2C Ch.*n* operations are shown below. Figure 13.4.6.1 and Figure 13.4.6.2 show an operation example and a flowchart, respectively.

Data receiving procedure

1. When receiving one-byte data, write 1 to the I2CnCTL.TXNACK bit.
2. Wait for a START condition interrupt (I2CnINTF.STARTIF bit = 1).
3. Check to see if the I2CnINTF.TR bit = 0 (reception mode).
(Start a data sending procedure if I2CnINTF.TR bit = 1.)
4. Clear the I2CnINTF.STARTIF bit by writing 1.
5. Wait for a receive buffer full interrupt (I2CnINTF.RBFIF bit = 1) generated when a one-byte reception has completed or an end of transfer interrupt (I2CnINTF.BYTEENDIF bit = 1).
Clear the I2CnINTF.BYTEENDIF bit by writing 1 after the interrupt has occurred.
6. If the next receive data is the last one, write 1 to the I2CnCTL.TXNACK bit to send a NACK after it is received.
7. Read the received data from the I2CnRXD register.
8. Repeat Steps 5 to 7 until the end of data reception.
9. Wait for a STOP condition interrupt (I2CnINTF.STOPIF bit = 1) or a START condition interrupt (I2CnINTF.STARTIF bit = 1).
 - i. Go to Step 10 when a STOP condition interrupt has occurred.
 - ii. Go to Step 3 when a START condition interrupt has occurred.
10. Clear the I2CnINTF.STOPIF bit and then terminate data receiving operations.

Data receiving operations

START condition detection and slave address check

It is the same as the data transmission in slave mode.

However, the I2CnINTF.TR bit is cleared to 0 and the I2CnINTF.TBEIF bit is not set.

If the I2CnMOD.GCEN bit is set to 1 (general call address response enabled), the I2C Ch.*n* starts data receiving operations when the general call address is received.

Slave mode can be operated even in SLEEP mode, it makes it possible to wake the CPU up using an interrupt when an address match is detected.

Receiving the first data byte

After the valid slave address has been received, the I2C Ch.*n* sends an ACK and pulls down SCL to low until 1 is written to the I2CnINTF.STARTIF bit. This puts the I²C bus into clock stretching state and the external master into standby state. When 1 is written to the I2CnINTF.STARTIF bit, the I2C Ch.*n* releases SCL and receives data sent from the external master into the shift register. After eight-bit data has been received, the I2C Ch.*n* sends an ACK and pulls down SCL to low. The received data in the shift register is transferred to the receive data buffer and the I2CnINTF.RBFIF and I2CnINTF.BYTEENDIF bits are both set to 1. After that, the received data can be read out from the I2CnRXD register.

Receiving subsequent data

When the received data is read out from the I2CnRXD register after the I2CnINTF.RBFIF bit has been set to 1, the I2C Ch.n clears the I2CnINTF.RBFIF bit to 0, releases SCL, and receives subsequent data sent from the external master. After eight-bit data has been received, the I2C Ch.n sends an ACK and pulls down SCL to low. The received data in the shift register is transferred to the receive data buffer and the I2CnINTF.RBFIF and I2CnINTF.BYTEENDIF bits are both set to 1.

To return a NACK after eight-bit data is received, such as when terminating data reception, write 1 to the I2CnCTL.TXNACK bit before the data reception is completed. The I2CnCTL.TXNACK bit is automatically cleared to 0 after a NACK has been sent.

STOP/repeated START condition detection

It is the same as the data transmission in slave mode.

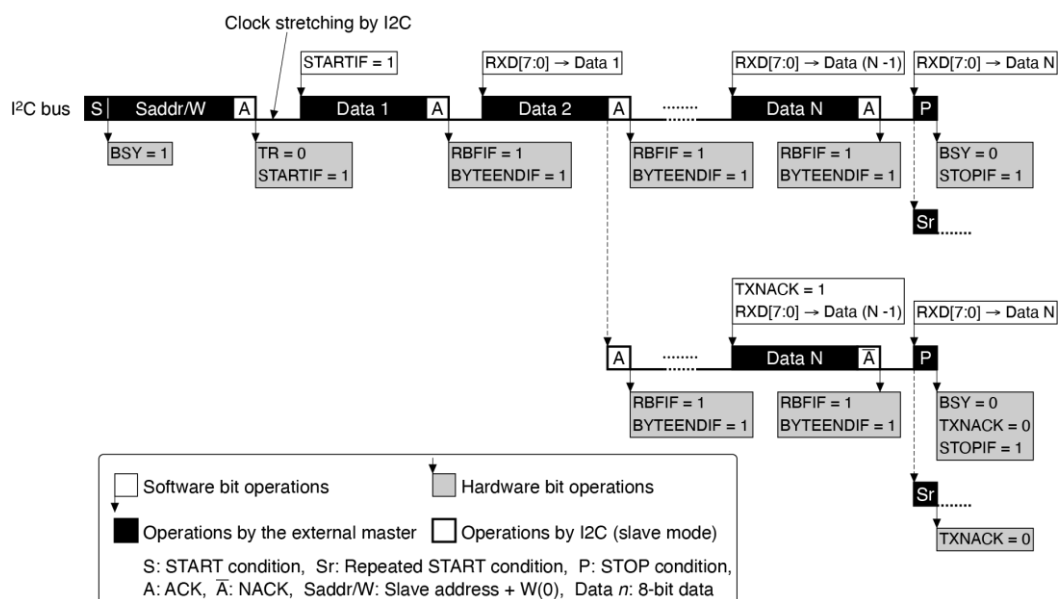


Figure 13.4.6.1 Example of Data Receiving Operations in Slave Mode

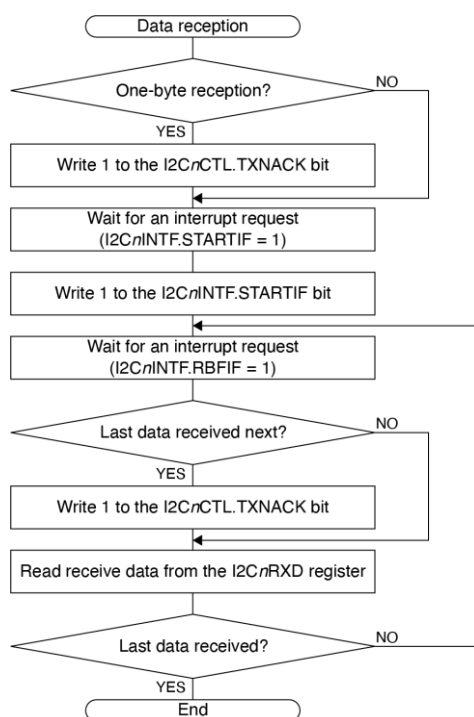


Figure 13.4.6.2 Slave Mode Data Reception Flowchart

13.4.7 Slave Operations in 10-bit Address Mode

The I2C Ch.*n* functions as a slave device in 10-bit address mode when the I2C*n*CTL.MST bit = 0 and the I2C*n*MOD.OADR10 bit = 1.

The following shows the address receiving operations in 10-bit address mode. Figure 13.4.7.1 shows an operation example. See Figure 13.4.4.1 for the 10-bit address configuration.

10-bit address receiving operations

After a START condition is issued, the master sends the first address that includes the two high-order slave address bits and the R/W bit (= 0). If the received two high-order slave address bits are matched with the I2C*n*OADR.OADR[9:8] bits, the I2C Ch.*n* returns an ACK. At this time, other slaves may return an ACK as the two high-order bits may be matched.

Then the master sends the eight low-order slave address bits as the second address. If this address is matched with the I2C*n*OADR.OADR[7:0] bits, the I2C Ch.*n* returns an ACK and starts data receiving operations.

If the master issues a request to the slave to send data (data reception in the master), the master generates a repeated START condition and sends the first address with the R/W bit set to 1. This reception switches the I2C Ch.*n* to data sending mode.

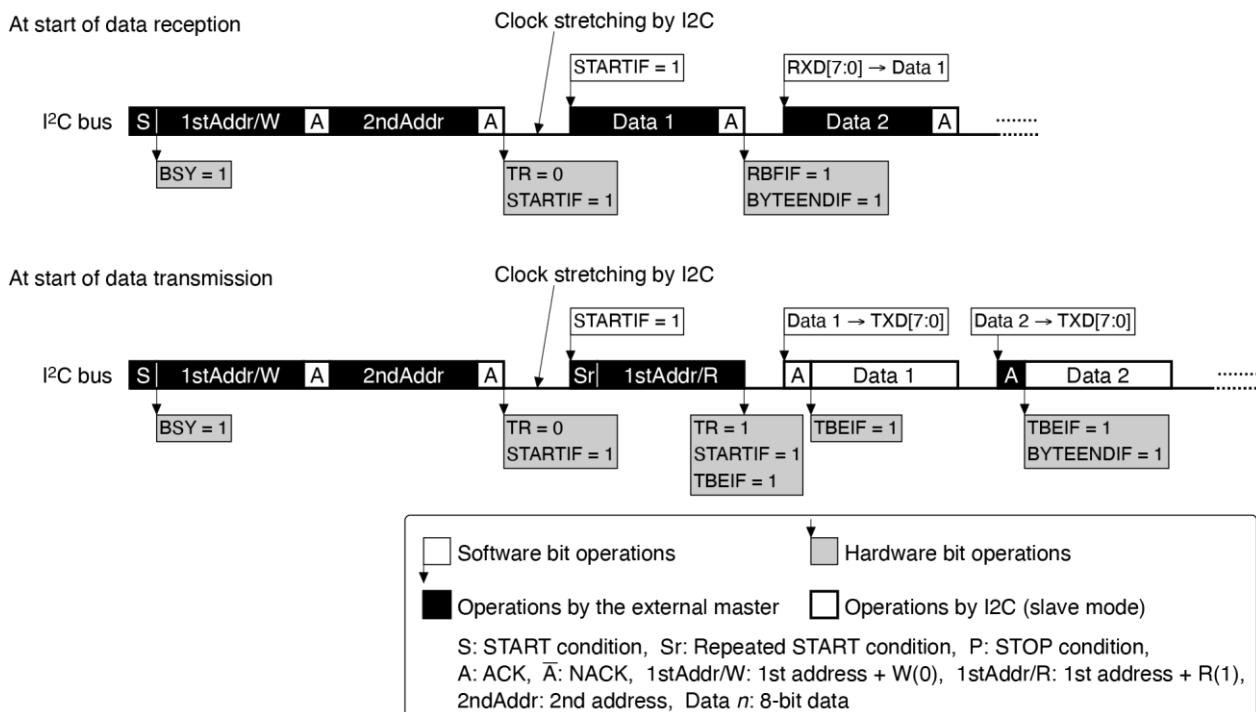


Figure 13.4.7.1 Example of Data Transfer Starting Operations in 10-bit Address Mode (Slave Mode)

13.4.8 Automatic Bus Clearing Operation

The I2C Ch.*n* set into master mode checks the SDA state immediately before generating a START condition. If SDA is set to a low level at this time, the I2C Ch.*n* automatically executes bus clearing operations that output up to ten clocks from the SCL*n* pin with SDA left free state.

When SDA goes high from low within nine clocks, the I2C Ch.*n* issues a START condition and starts normal operations. If SDA does not change from low when the I2C Ch.*n* outputs the ninth clock, it is regarded as an automatic bus clearing failure. In this case, the I2C Ch.*n* clears the I2C*n*CTL.TXSTART bit to 0 and sets both the I2C*n*INTF.ERRIF and I2C*n*INTF.STARTIF bits to 1.

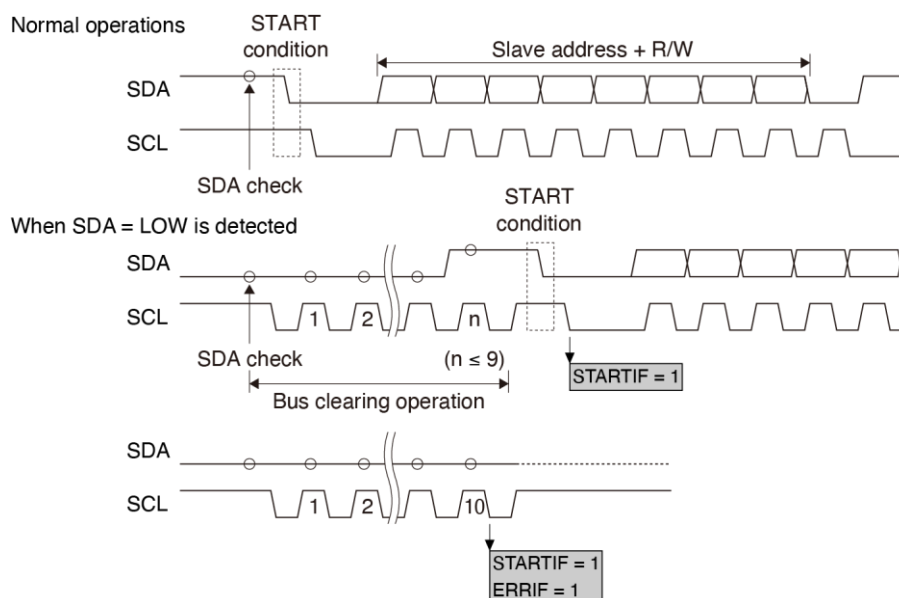


Figure 13.4.8.1 Automatic Bus Clearing Operation

13.4.9 Error Detection

The I²C includes a hardware error detection function.

Furthermore, the `I2CnINTF.SDALOW` and `I2CnINTF.SCLLOW` bits are provided to allow software to check whether the SDA and SCL lines are fixed at low. If unintended low level is detected on SDA or SCL, a software recovery processing, such as I²C Ch.*n* software reset, can be performed.

The table below lists the hardware error detection conditions and the notification method.

Table 13.4.9.1 Hardware Error Detection Function

No.	Error detecting period/timing	I ² C bus line monitored and error condition	Notification method
1	While the I ² C Ch. <i>n</i> controls SDA to high for sending address, data, or a NACK	SDA = low	<code>I2CnINTF.ERRIF = 1</code>
2	<Master mode only> When 1 is written to the <code>I2CnCTL.TXSTART</code> bit while the <code>I2CnINTF.BSY</code> bit = 0	SCL = low	<code>I2CnINTF.ERRIF = 1</code> <code>I2CnCTL.TXSTART = 0</code> <code>I2CnINTF.STARTIF = 1</code>
3	<Master mode only> When 1 is written to the <code>I2CnCTL.TXSTOP</code> bit while the <code>I2CnINTF.BSY</code> bit = 0	SCL = low	<code>I2CnINTF.ERRIF = 1</code> <code>I2CnCTL.TXSTOP = 0</code> <code>I2CnINTF.STOPIF = 1</code>
4	<Master mode only> When 1 is written to the <code>I2CnCTL.TXSTART</code> bit while the <code>I2CnINTF.BSY</code> bit = 0 (Refer to "Automatic Bus Clearing Operation.")	SDA Automatic bus clearing failure	<code>I2CnINTF.ERRIF = 1</code> <code>I2CnCTL.TXSTART = 0</code> <code>I2CnINTF.STARTIF = 1</code>

13.5 Interrupts

The I2C has a function to generate the interrupts shown in Table 13.5.1.

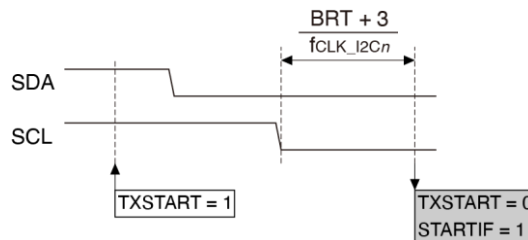
Table 13.5.1 I2C Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
End of data transfer	I2CnINTF.BYTEENDIF	When eight-bit data transfer and the following ACK/NACK transfer are completed	Writing 1, software reset
General call address reception	I2CnINTF.GCIF	Slave mode only: When the general call address is received	Writing 1, software reset
NACK reception	I2CnINTF.NACKIF	When a NACK is received	Writing 1, software reset
STOP condition	I2CnINTF.STOPIF	Master mode: When a STOP condition is generated and the bus free time (t_{BUF}) between STOP and START conditions has elapsed Slave mode: When a STOP condition is detected while the I2C Ch.n is selected as the slave currently accessed	Writing 1, software reset
START condition	I2CnINTF.STARTIF	Master mode: When a START condition is issued Slave mode: When an address match is detected (including general call)	Writing 1, software reset
Error detection	I2CnINTF.ERRIF	Refer to "Error Detection."	Writing 1, software reset
Receive buffer full	I2CnINTF.RBFIF	When received data is loaded to the receive data buffer	Reading received data (to empty the receive data buffer), software reset
Transmit buffer empty	I2CnINTF.TBEIF	Master mode: When a START condition is issued or when an ACK is received from the slave Slave mode: When transmit data written to the transmit data buffer is transferred to the shift register or when an address match is detected with R/W bit set to 1	Writing transmit data

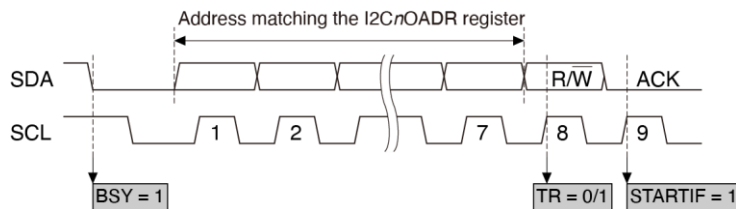
The I2C provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the "Interrupt Controller" chapter.

(1) START condition interrupt

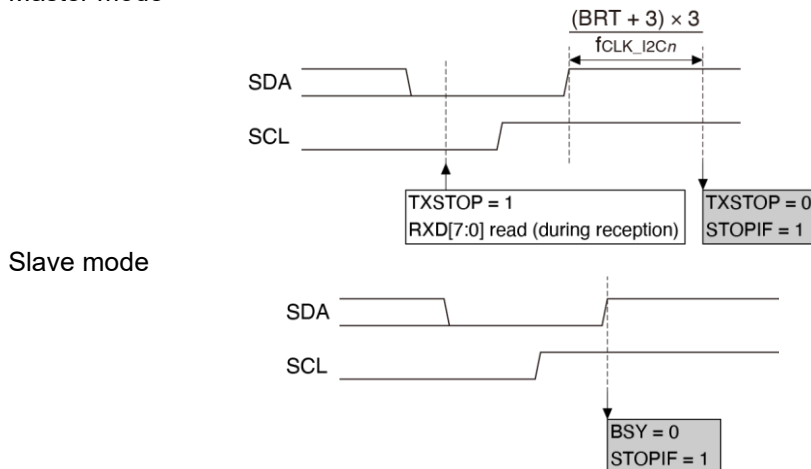
Master mode



Slave mode



(2) STOP condition interrupt
Master mode



(f_{CLK_I2Cn} : I2C operating clock frequency [Hz], BRT: I2CnBR.BRT[6:0] bits setting value (1 to 127))

Figure 13.5.1 START/STOP Condition Interrupt Timings

13.6 Control Registers

I2C Ch.n Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/W	
	3–2	–	0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the I2C operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

These bits select the division ratio of the I2C operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the I2C.

Table 13.6.1 Clock Source and Division Ratio Settings

I2CnCLK. CLKDIV[1:0] bits	I2CnCLK.CLKSRC[1:0] bits			
	0x0 IOSC	0x1 OSC1	0x2 OSC3	0x3 EXOSC
0x3	1/8	1/1	1/8	1/1
0x2	1/4		1/4	
0x1	1/2		1/2	
0x0	1/1		1/1	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The I2CnCLK register settings can be altered only when the I2CnCTL.MODEN bit = 0.

13 I²C (I2C)

I2C Ch.n Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnMOD	15–8	–	0x00	–	R	–
	7–3	–	0x00	–	R	
	2	OADR10	0	H0	R/W	
	1	GCEN	0	H0	R/W	
	0	–	0	–	R	

Bits 15–3 Reserved

Bit 2 OADR10

This bit sets the number of own address bits for slave mode.

1 (R/W): 10-bit address

0 (R/W): 7-bit address

Bit 1 GCEN

This bit sets whether to respond to master general calls in slave mode or not.

1 (R/W): Respond to general calls.

0 (R/W): Do not respond to general calls.

Bit 0 Reserved

Note: The I2CnMOD register settings can be altered only when the I2CnCTL.MODEN bit = 0.

I2C Ch.n Baud-Rate Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnBR	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6–0	BRT[6:0]	0x7f	H0	R/W	

Bits 15–7 Reserved

Bits 6–0 BRT[6:0]

These bits set the I2C Ch.n transfer rate for master mode. For more information, refer to “Baud Rate Generator.”

Notes: • The I2CnBR register settings can be altered only when the I2CnCTL.MODEN bit = 0.

- Be sure to avoid setting the I2CnBR register to 0.

I2C Ch.n Own Address Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnOADR	15–10	–	0x00	–	R	–
	9–0	OADR[9:0]	0x000	H0	R/W	

Bits 15–10 Reserved

Bits 9–0 OADR[9:0]

These bits set the own address for slave mode.

The I2CnOADR.OADR[9:0] bits are effective in 10-bit address mode (I2CnMOD.OADR10 bit = 1), or the I2CnOADR.OADR[6:0] bits are effective in 7-bit address mode (I2CnMOD.OADR10 bit = 0).

Note: The I2CnOADR register settings can be altered only when the I2CnCTL.MODEN bit = 0.

I2C Ch.n Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnCTL	15–8	–	0x00	–	R	–
	7–6	–	0x0	–	R	
	5	MST	0	H0	R/W	
	4	TXNACK	0	H0/S0	R/W	
	3	TXSTOP	0	H0/S0	R/W	
	2	TXSTART	0	H0/S0	R/W	
	1	SFTRST	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–6 Reserved**Bit 5 MST**

This bit selects the I2C Ch.*n* operating mode.

1 (R/W): Master mode

0 (R/W): Slave mode

Bit 4 TXNACK

This bit issues a request for sending a NACK at the next responding.

1 (W): Issue a NACK.

0 (W): Ineffective

1 (R): On standby or during sending a NACK

0 (R): NACK has been sent.

This bit is automatically cleared after a NACK has been sent.

Bit 3 TXSTOP

This bit issues a STOP condition in master mode. This bit is ineffective in slave mode.

1 (W): Issue a STOP condition.

0 (W): Ineffective

1 (R): On standby or during generating a STOP condition

0 (R): STOP condition has been generated.

This bit is automatically cleared when the bus free time (t_{BUF} defined in the I2C Specifications) has elapsed after the STOP condition has been generated.

Bit 2 TXSTART

This bit issues a START condition in master mode. This bit is ineffective in slave mode.

1 (W): Issue a START condition.

0 (W): Ineffective

1 (R): On standby or during generating a START condition

0 (R): START condition has been generated.

This bit is automatically cleared when a START condition has been generated.

Bit 1 SFTRST

This bit issues software reset to the I2C.

1 (W): Issue software reset

0 (W): Ineffective

1 (R): Software reset is executing.

0 (R): Software reset has finished. (During normal operation)

Setting this bit resets the I2C transmit/receive control circuit and interrupt flags. This bit is automatically cleared after the reset processing has finished.

Bit 0 MODEN

This bit enables the I2C operations.

1 (R/W): Enable I2C operations (The operating clock is supplied.)

0 (R/W): Disable I2C operations (The operating clock is stopped.)

Note: If the I2CnCTL.MODEN bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the I2CnCTL.MODEN bit to 1 again after that, be sure to write 1 to the I2CnCTL.SFTRST bit as well.

I2C Ch.*n* Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnTXD	15–8	–	0x00	–	R	–
	7–0	TXD[7:0]	0x00	H0	R/W	–

Bits 15–8 Reserved

13 I²C (I2C)

Bits 7–0 TXD[7:0]

Data can be written to the transmit data buffer through these bits. Make sure the I2CnINTF.TBEIF bit is set to 1 before writing data.

Note: Be sure to avoid writing to the I2CnTXD register when the I2CnINTF.TBEIF bit = 0, otherwise transmit data cannot be guaranteed.

I2C Ch.n Receive Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnRXD	15–8	–	0x00	–	R	–
	7–0	RXD[7:0]	0x00	H0	R	–

Bits 15–8 Reserved

Bits 7–0 RXD[7:0]

The receive data buffer can be read through these bits.

I2C Ch.n Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnINTF	15–13	–	0x0	–	R	–
	12	SDALOW	0	H0	R	Cleared by writing 1.
	11	SCLLOW	0	H0	R	
	10	BSY	0	H0/S0	R	
	9	TR	0	H0	R	
	8	–	0	–	R	
	7	BYTEENDIF	0	H0/S0	R/W	
	6	GCIF	0	H0/S0	R/W	
	5	NACKIF	0	H0/S0	R/W	
	4	STOPIF	0	H0/S0	R/W	
	3	STARTIF	0	H0/S0	R/W	
	2	ERRIF	0	H0/S0	R/W	
	1	RBFIF	0	H0/S0	R	Cleared by reading the I2CnRXD register.
0	TBEIF	0	H0/S0	R	Cleared by writing to the I2CnTXD register.	

Bits 15–13 Reserved

Bit 12 SDALOW

This bit indicates that SDA is set to low level.

1 (R): SDA = Low level

0 (R): SDA = High level

Bit 11 SCLLOW

This bit indicates that SCL is set to low level.

1 (R): SCL = Low level

0 (R): SCL = High level

Bit 10 BSY

This bit indicates that the I²C bus is placed into busy status.

1 (R): I²C bus busy

0 (R): I²C bus free

Bit 9 TR

This bit indicates whether the I2C is set in transmission mode or not.

1 (R): Transmission mode

0 (R): Reception mode

Bit 8 Reserved

Bit 7	BYTEENDIF
Bit 6	GCIF
Bit 5	NACKIF
Bit 4	STOPIF
Bit 3	STARTIF
Bit 2	ERRIF
Bit 1	RBFIF
Bit 0	TBEIF

These bits indicate the I2C interrupt cause occurrence status.

- 1 (R): Cause of interrupt occurred
 0 (R): No cause of interrupt occurred
 1 (W): Clear flag (BYTEENDIF, GCIF, NACKIF, STOPIF, STARTIF, ERRIF)
 Ineffective (RBFIF, TBEIF)
 0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

- I2CnINTF.BYTEENDIF bit: End of transfer interrupt
 I2CnINTF.GCIF bit: General call address reception interrupt
 I2CnINTF.NACKIF bit: NACK reception interrupt
 I2CnINTF.STOPIF bit: STOP condition interrupt
 I2CnINTF.STARTIF bit: START condition interrupt
 I2CnINTF.ERRIF bit: Error detection interrupt
 I2CnINTF.RBFIF bit: Receive buffer full interrupt
 I2CnINTF.TBEIF bit: Transmit buffer empty interrupt

I2C Ch.n Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2CnINTE	15–8	–	0x00	–	R	–
	7	BYTEENDIE	0	H0	R/W	
	6	GCIE	0	H0	R/W	
	5	NACKIE	0	H0	R/W	
	4	STOPIE	0	H0	R/W	
	3	STARTIE	0	H0	R/W	
	2	ERRIE	0	H0	R/W	
	1	RBFIE	0	H0	R/W	
	0	TBEIE	0	H0	R/W	

Bits 15–8 Reserved

Bit 7	BYTEENDIE
Bit 6	GCIE
Bit 5	NACKIE
Bit 4	STOPIE
Bit 3	STARTIE
Bit 2	ERRIE
Bit 1	RBFIE
Bit 0	TBEIE

These bits enable I2C interrupts.

- 1 (R/W): Enable interrupts
 0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

- I2CnINTE.BYTEENDIE bit: End of transfer interrupt
 I2CnINTE.GCIE bit: General call address reception interrupt
 I2CnINTE.NACKIE bit: NACK reception interrupt
 I2CnINTE.STOPIE bit: STOP condition interrupt
 I2CnINTE.STARTIE bit: START condition interrupt
 I2CnINTE.ERRIE bit: Error detection interrupt
 I2CnINTE.RBFIE bit: Receive buffer full interrupt
 I2CnINTE.TBEIE bit: Transmit buffer empty interrupt

14 16-bit PWM Timers (T16B)

14.1 Overview

T16B is a 16-bit PWM timer with comparator/capture functions. The features of T16B are listed below.

- Counter block
 - 16-bit up/down counter
 - A clock source and a clock division ratio for generating the count clock are selectable in each channel.
 - The count mode is configurable from combinations of up, down, or up/down count operations, and one-shot operations (counting for one cycle configured) or repeat operations (counting continuously until stopped via software).
 - Supports an event counter function using an external clock.
- Comparator/capture block
 - Supports up to six comparator/capture circuits to be included per one channel.
 - The comparator compares the counter value with the values specified via software to generate interrupt signals and a PWM waveform. (Can be used as an interval timer, PWM waveform generator, and external event counter.)
 - The capture circuit captures counter values using external/software trigger signals and generates interrupts. (Can be used to measure external event periods/cycles.)

Figure 14.1.1 shows the T16B configuration.

Table 14.1.1 T16B Channel Configuration of S1C17W11

Item	S1C17W11
Number of channels	2 channels (Ch.0 and Ch.1)
Event counter function	Ch.0: EXCL00 or EXCL01 pin input Ch.1: EXCL10 or EXCL11 pin input
Number of comparator/capture circuits per channel	2 systems (0 and 1)
Timer generating signal output	Ch.0: TOUT00 and TOUT01 pin outputs (2 systems) Ch.1: TOUT10 and TOUT11 pin outputs (2 systems)
Capture signal input	Ch.0: CAP00 and CAP01 pin inputs (2 systems) Ch.1: CAP10 and CAP11 pin inputs (2 systems)

Note: In this chapter, 'n' refers to a channel number, and 'm' refers to an input/output pin number or a comparator/capture circuit number in a channel.

14 16-bit PWM Timers (T16B)

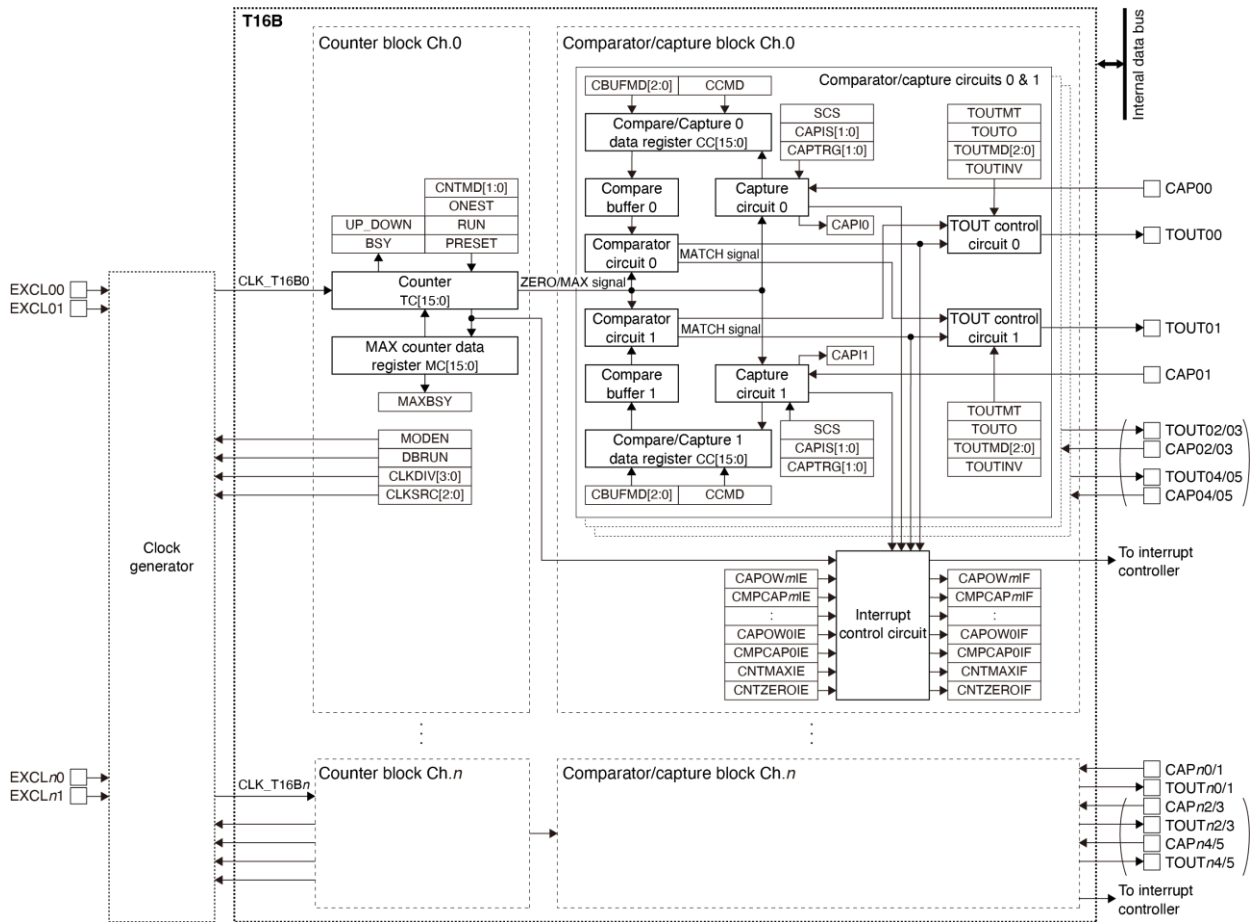


Figure 14.1.1 T16B Configuration

14.2 Input/Output Pins

Table 14.2.1 lists the T16B pins.

Table 14.2.1 List of T16B Pins

Pin name	I/O*	Initial status*	Function
EXCL nm	I	I (Hi-Z)	External clock input
TOUT nm /CAP nm	O or I	O (L)	TOUT signal output (in comparator mode) or capture trigger signal input (in capture mode)

* Indicates the status when the pin is configured for T16B.

If the port is shared with the T16B pin and other functions, the T16B input/output function must be assigned to the port before activating T16B. For more information, refer to the “I/O Ports” chapter.

14.3 Clock Settings

14.3.1 T16B Operating Clock

When using T16B Ch. n , the T16B Ch. n operating clock CLK_T16B n must be supplied to T16B Ch. n from the clock generator. The CLK_T16B n supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).

When an external clock is used, select the EXCL nm pin function (refer to the “I/O Ports” chapter).

2. Set the following T16B n CLK register bits:

- T16B n CLK.CLKSRC[2:0] bits (Clock source selection)
- T16B n CLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)

14.3.2 Clock Supply in SLEEP Mode

When using T16B during SLEEP mode, the T16B operating clock CLK_T16B n must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_T16B n clock source.

If the CLGOSC.xxxxSLPC bit for the CLK_T16B n clock source is 1, the CLK_T16B n clock source is deactivated during SLEEP mode and T16B stops with the register settings and counter value maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_T16B n is supplied and the T16B operation resumes.

14.3.3 Clock Supply in DEBUG Mode

The CLK_T16B n supply during DEBUG mode should be controlled using the T16B n CLK.DBRUN bit.

The CLK_T16B n supply to T16B Ch. n is suspended when the CPU enters DEBUG mode if the T16B n CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_T16B n supply resumes. Although T16B Ch. n stops operating when the CLK_T16B n supply is suspended, the counter and registers retain the status before DEBUG mode was entered. If the T16B n CLK.DBRUN bit = 1, the CLK_T16B n supply is not suspended and T16B Ch. n will keep operating in DEBUG mode.

14.3.4 Event Counter Clock

When EXCL nm is selected as the clock source using the T16B n CLK.CLKSRC[2:0] bits, the channel functions as a timer or event counter that counts the EXCL nm pin input clocks.

The counter counts rising edges of the input signal. This can be changed so that the counter will count falling edges of the original signal by selecting EXCL nm inverted input as the clock source.

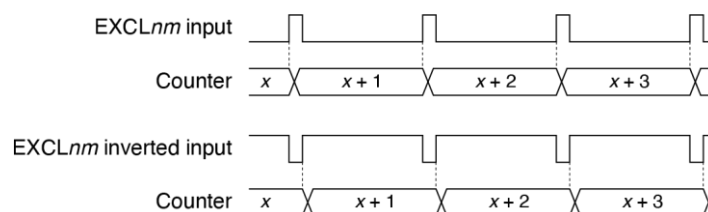


Figure 14.3.4.1 Count Timing (During Count Up Operation)

Note: When running the counter using the event counter clock, two dummy clocks must be input before the first counting up/down can be performed.

14.4 Operations

14.4.1 Initialization

T16B Ch. n should be initialized and started counting with the procedure shown below. Perform initial settings for comparator mode when using T16B as an interval timer, PWM waveform generator, or external event counter. Perform initial settings for capture mode when using T16B to measure external event periods/cycles.

Initial settings for comparator mode

1. Configure the T16B Ch. n operating clock.
 2. Set the T16B n CTL.MODEN bit to 1. (Enable T16B operations)
 3. Set the following T16B n CCCTL0 and T16B n CCCTL1 register bits:
 - Set the T16B n CCCTL m .CCMD bit to 0. * (Set comparator mode)
 - T16B n CCCTL m .CBUFMD[2:0] bits (Configure compare buffer)
- * Another circuit in the comparator/capture circuit pair (circuits 0 and 1, 2 and 3, 4 and 5) can be set to capture mode.

Set the following bits when the TOUT nm output is used.

- T16B n CCCTL m .TOUTMT bit (Select waveform generation signal)
 - T16B n CCCTL m .TOUTMD[2:0] bits (Select TOUT signal generation mode)
 - T16B n CCCTL m .TOUTINV bit (Select TOUT signal polarity)
4. Check to see if the T16B n CTL.MAXBSY bit is set to 0.

14 16-bit PWM Timers (T16B)

5. Set the T16BnCCR0 and T16BnCCR1 registers. (Set the counter comparison value)
6. Set the T16BnMC register. (Set MAX counter data)
7. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the T16BnINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the T16BnINTE register to 1. (Enable interrupts)
8. Check to see if the T16BnCTL.MAXBSY and the T16BnCTL.RUN bit are set to 0.
9. Set the following T16BnCTL register bits:
 - T16BnCTL.CNTMD[1:0] bits (Select count up/down operation)
 - T16BnCTL.ONEST bit (Select one-shot/repeat operation)
 - Set the T16BnCTL.PRESET bit to 1. (Reset counter)
 - Set the T16BnCTL.RUN bit to 1. (Start counting)

Initial settings for capture mode

1. Configure the T16B Ch.n operating clock.
2. Set the T16BnCTL.MODEN bit to 1. (Enable T16B operations)
3. Set the following T16BnCCCTL0 and T16BnCCCTL1 register bits:
 - Set the T16BnCCCTLm.CCMD bit to 1. * (Set capture mode)
 - T16BnCCCTLm.SCS bit (Set synchronous/asynchronous mode)
 - T16BnCCCTLm.CAPIS[1:0] bits (Set trigger signal)
 - T16BnCCCTLm.CAPTRG[1:0] bits (Select trigger edge)

* Another circuit in the comparator/capture circuit pair (circuits 0 and 1, 2 and 3, 4 and 5) can be set to comparator mode.
4. Check to see if the T16BnCTL.MAXBSY bit is set to 0.
5. Set the T16BnMC register. (Set MAX counter data)
6. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the T16BnINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the T16BnINTE register to 1. (Enable interrupts)
7. Check to see if the T16BnCTL.MAXBSY and the T16BnCTL.RUN bit are set to 0.
8. Set the following T16BnCTL register bits:
 - T16BnCTL.CNTMD[1:0] bits (Select count up/down operation)
 - T16BnCTL.ONEST bit (Select one-shot/repeat operation)
 - Set the T16BnCTL.PRESET bit to 1. (Reset counter)
 - Set the T16BnCTL.RUN bit to 1. (Start counting)

14.4.2 Counter Block Operations

The counter in each counter block channel is a 16-bit up/down counter that counts the selected operating clock (count clock).

Count mode

The T16BnCTL.CNTMD[1:0] bits allow selection of up, down, and up/down mode. The T16BnCTL.ONEST bit allows selection of repeat and one-shot mode. The counter operates in six counter modes specified with a combination of these modes.

Repeat mode enables the counter to continue counting until stopped via software. Select this mode to generate periodic interrupts at desired intervals or to generate timer output waveforms.

One-shot mode enables the counter to stop automatically. Select this mode to stop the counter after an interrupt has occurred once, such as for measuring pulse width or external event intervals and checking a specific lapse of time.

Up, down, and up/down mode configures the counter as an up counter, down counter and up/down counter, respectively.

MAX counter data register

The MAX counter data register (T16BnMC.MC[15:0] bits) is used to set the maximum value of the counter (hereafter referred to as MAX value). This setting limits the count range to 0x0000–MAX value and determines the

count and interrupt cycles. When the counter is set to repeat mode, the MAX value can be rewritten in the procedure shown below even if the counter is running.

1. Check to see if the T16BnCTL.MAXBSY bit is set to 0.
2. Write the MAX value to the T16BnMC.MC[15:0] bits.

Note: When rewriting the MAX value, the new MAX value should be written after the counter has been reset to the previously set MAX value.

Counter reset

Setting the T16BnCTL.PRESET bit to 1 resets the counter. This clears the counter to 0x0000 in up or up/down mode, or presets the MAX value to the counter in down mode.

The counter is also cleared to 0x0000 when the counter value exceeds the MAX value during count up operation.

Counting start

To start counting, set the T16BnCTL.RUN bit to 1. The counting stop control depends on the count mode set.

Counter value read

The counter value can be read out from the T16BnTC.TC[15:0] bits. However, since T16B operates on CLK_T16Bn, one of the operations shown below is required to read correctly by the CPU.

- Read the counter value twice or more and check to see if the same value is read.
- Stop the timer and then read the counter value.

Counter status check

The counter operating status can be checked using the T16BnCS.BSY bit. The T16BnCS.BSY bit is set to 1 while the counter is running or 0 while the counter is idle.

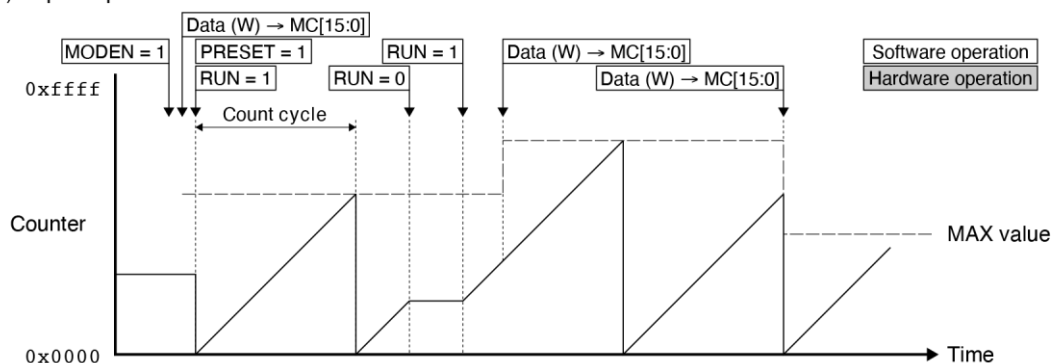
The current count direction can also be checked using the T16BnCS.UP_DOWN bit. The T16BnCS.UP_DOWN bit is set to 1 during count up operation or 0 during count down operation.

Operations in repeat up count and one-shot up count modes

In these modes, the counter operates as an up counter and counts from 0x0000 (or current value) to the MAX value. In repeat up count mode, the counter returns to 0x0000 if it exceeds the MAX value and continues counting until the T16BnCTL.RUN bit is set to 0. If the MAX value is altered to a value larger than the current counter value during counting, the counter keeps counting up to the new MAX value. If the MAX value is altered to a value smaller than the current counter value, the counter is cleared to 0x0000 and continues counting up to the new MAX value.

In one-shot up count mode, the counter returns to 0x0000 if it exceeds the MAX value and stops automatically at that point.

(1) Repeat up count mode



14 16-bit PWM Timers (T16B)

(2) One-shot up count mode

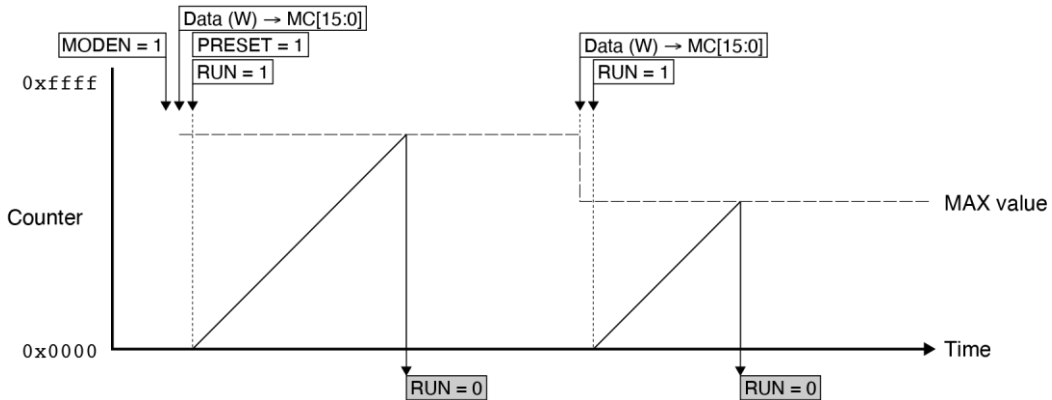


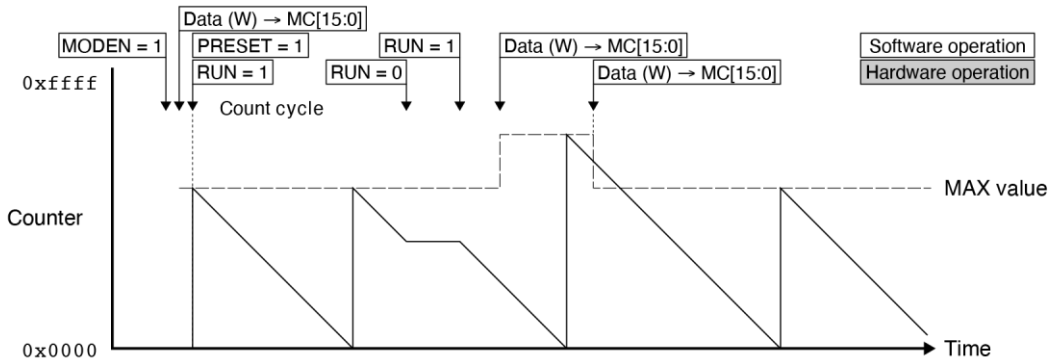
Figure 14.4.2.1 Operations in Repeat Up Count and One-shot Up Count Modes

Operations in repeat down count and one-shot down count modes

In these modes, the counter operates as a down counter and counts from the MAX value (or current value) to 0x0000.

In repeat down count mode, the counter returns to the MAX value if a counter underflow occurs and continues counting until the T16BnCTL.RUN bit is set to 0. If the MAX value is altered during counting, the counter keeps counting down to 0x0000 and continues counting down from the new MAX value after a counter underflow occurs. In one-shot down count mode, the counter returns to the MAX value if a counter underflow occurs and stops automatically at that point.

(1) Repeat down count mode



(2) One-shot down count mode

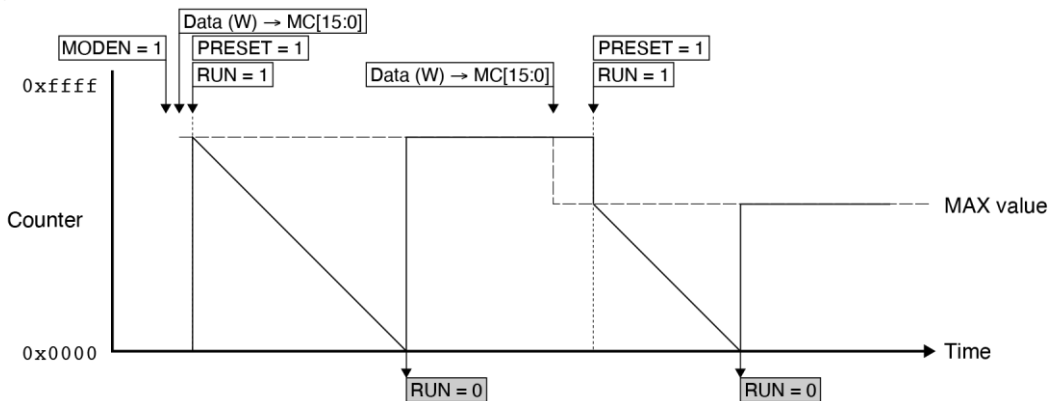


Figure 14.4.2.2 Operations in Repeat Down Count and One-shot Down Count Modes

Operations in repeat up/down count and one-shot up/down count modes

In these modes, the counter operates as an up/down counter and counts as 0x0000 (or current value) → the MAX value → 0x0000.

In repeat up/down count mode, the counter repeats counting up from 0x0000 to the MAX value and counting down from the MAX value to 0x0000 until the T16BnCTL.RUN bit is set to 0. If the MAX value is altered to a value larger than the current counter value during count up operation, the counter keeps counting up to the new MAX value. If the MAX value is altered to a value smaller than the current counter value, the counter is cleared to 0x0000 and continues counting up to the new MAX value. If the MAX value is altered during count down operation, the counter keeps counting down to 0x0000 and then starts counting up to the new MAX value.

In one-shot up/down count mode, the counter stops automatically when it reaches 0x0000 during count down operation.

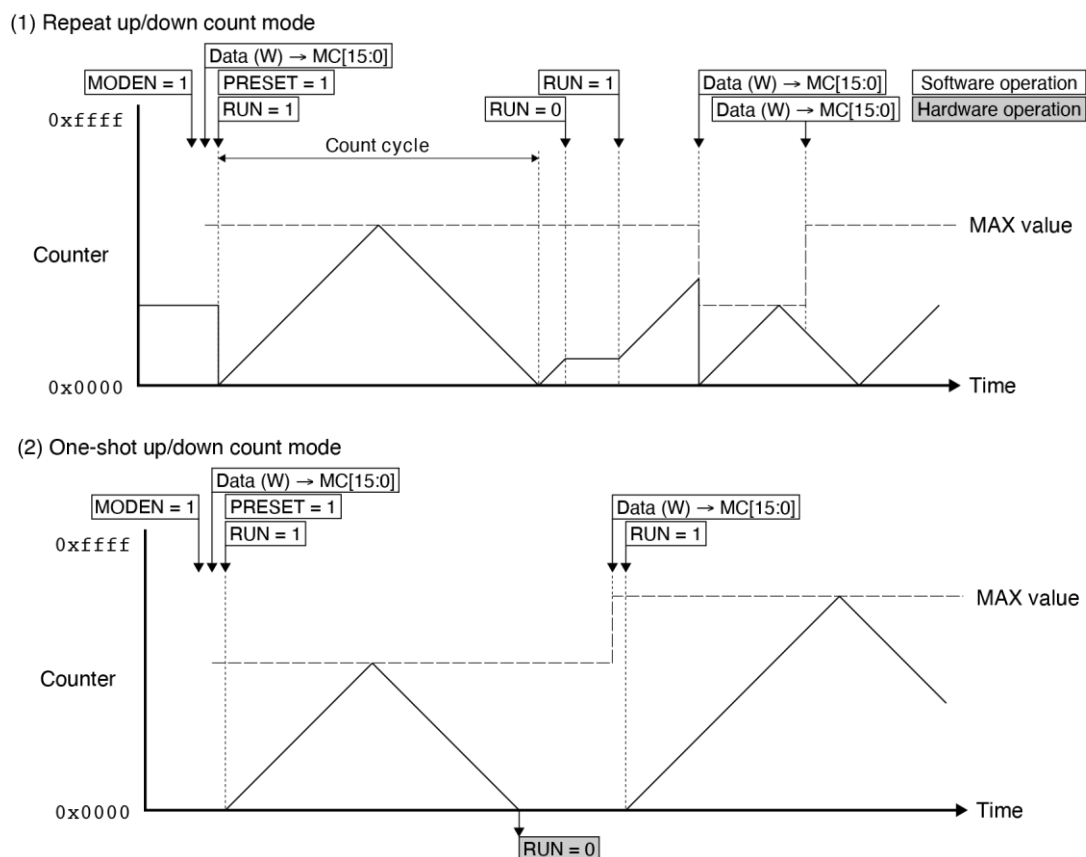


Figure 14.4.2.3 Operations in Repeat Up/Down Count and One-shot Up/Down Count Modes

14.4.3 Comparator/Capture Block Operations

The comparator/capture block functions as a comparator to compare the counter value with the register value set or a capture circuit to capture counter values using the external/software trigger signals.

Comparator/capture block operating mode

The comparator/capture block includes two systems (four or six systems) of comparator/capture circuits and each system can be set to comparator mode or capture mode, individually.

Set the T16BnCCCTLm.CCMD bit to 0 to set the comparator/capture circuit m to comparator mode or 1 to set it to capture mode.

Operations in comparator mode

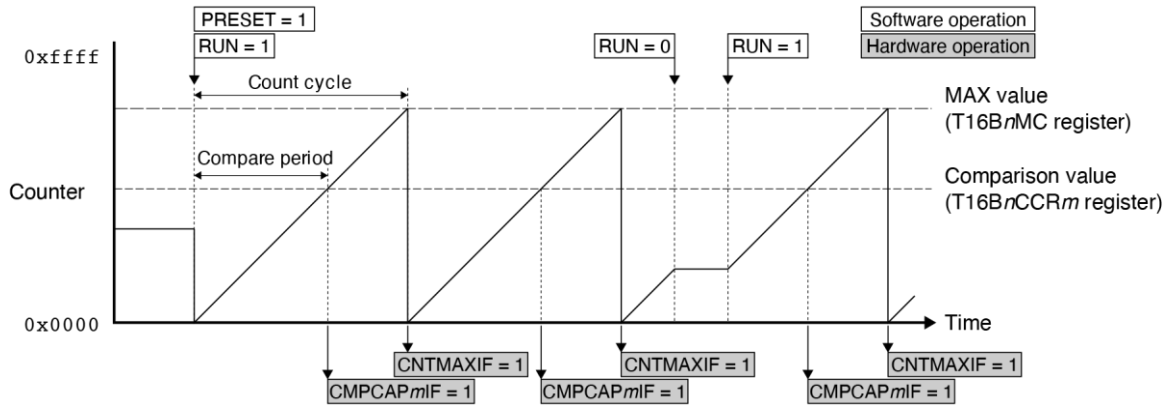
The comparator mode compares the counter value and the value set via software. It generates an interrupt and toggles the timer output signal level when the values are matched. The T16BnCCRm register functions as the compare data register used for setting a comparison value in this mode. The TOUTnm/CAPnm pin is configured to the TOUTnm pin.

14 16-bit PWM Timers (T16B)

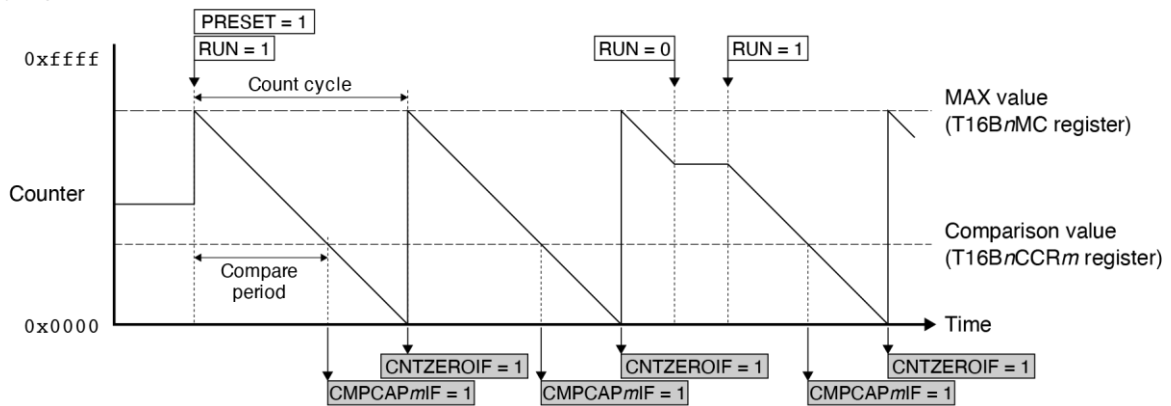
When the counter reaches the value set in the T16BnCCRm register during counting, the comparator asserts the MATCH signal and sets the T16BnINTF.COMPCAPmIF bit (compare interrupt flag) to 1.

When the counter reaches the MAX value in comparator mode, the T16BnINTF.CNTMAXIF bit (counter MAX interrupt flag) is set to 1. When the counter reaches 0x0000, the T16BnINTF.CNTZEROIF bit (counter zero interrupt flag) is set to 1.

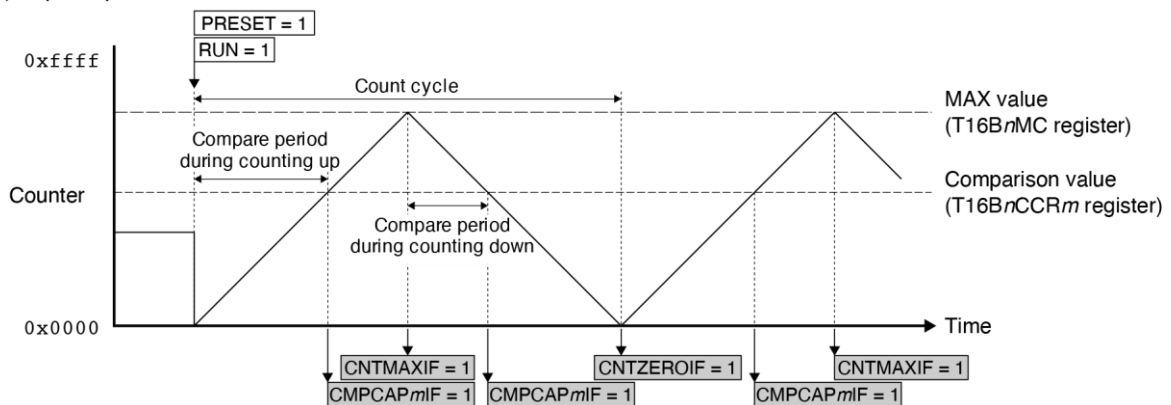
(1) Repeat up count mode



(2) Repeat down count mode



(3) Repeat up/down count mode



(Note that the T16BnINTF.COMPCAPmIF/CNTMAXIF/CNTZEROIF bit clearing operations via software are omitted from the figure.)

Figure 14.4.3.1 Operation Examples in Comparator Mode

The time from counter = 0x0000 or MAX value to occurrence of a compare interrupt (compare period) and the time to occurrence of a counter MAX or counter zero interrupt (count cycle) can be calculated as follows:

During counting up

$$\text{Compare period} = \frac{(CC + 1)}{f_{\text{CLK_T16B}}} \text{ [s]} \quad \text{Count cycle} = \frac{(\text{MAX} + 1)}{f_{\text{CLK_T16B}}} \text{ [s]} \quad (\text{Eq. 14.1})$$

During counting down

$$\text{Compare period} = \frac{(\text{MAX} - CC + 1)}{f_{\text{CLK_T16B}}} \text{ [s]} \quad \text{Count cycle} = \frac{(\text{MAX} + 1)}{f_{\text{CLK_T16B}}} \text{ [s]} \quad (\text{Eq. 14.2})$$

Where

CC: T16BnCCRm register setting value (0 to 65,535)

MAX: T16BnMC register setting value (0 to 65,535)

f_{CLK_T16B}: Count clock frequency [Hz]

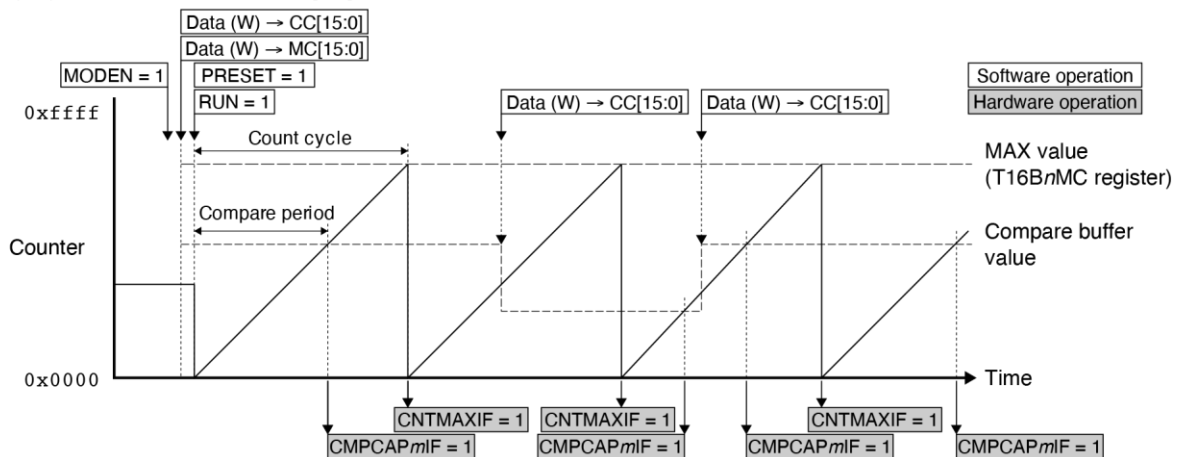
The comparator MATCH signal and counter MAX/ZERO signals are also used to generate a timer output waveform (TOUT). Refer to “TOUT Output Control” for more information.

Compare buffer

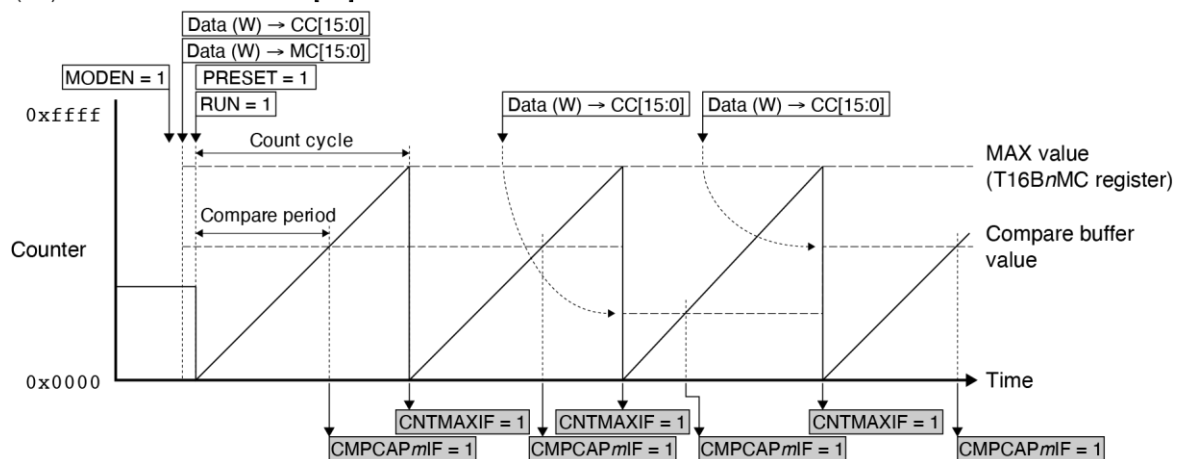
The comparator loads the comparison value, which has been written to the T16BnCCRm register, to the compare buffer before comparing it with the counter value. For example, when generating a PWM waveform, the waveform with the desired duty ratio may not be generated if the comparison value is altered asynchronous to the count operation. To avoid this problem, the timing to load the comparison value to the compare buffer can be configured using the T16BnCCCTLm.CBUFMD[2:0] bits for synchronization with the count operation.

(1) Repeat up count mode

(1.1) T16BnCCCTLm.CBUFMD[2:0] bits = 0x0

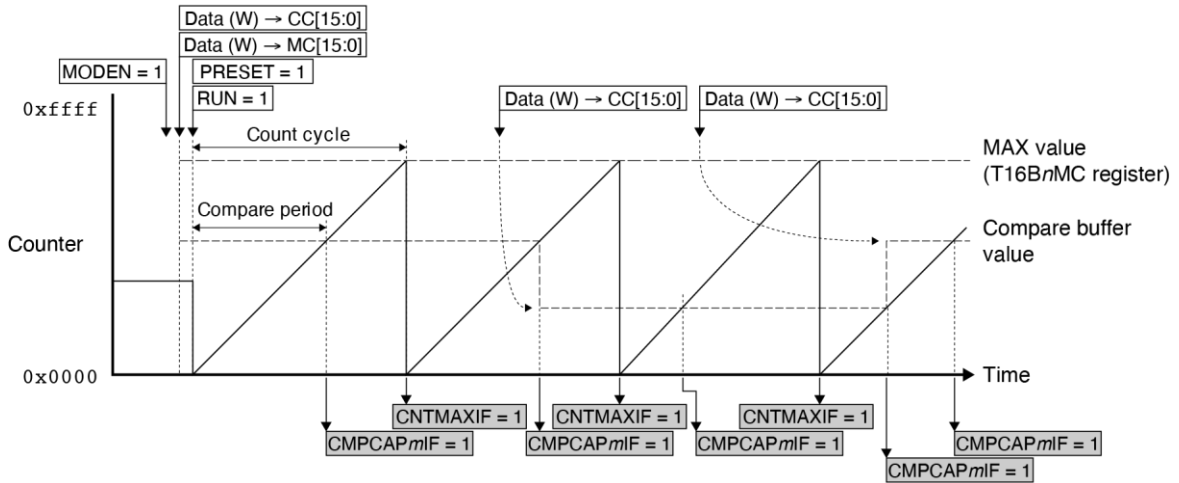


(1.2) T16BnCCCTLm.CBUFMD[2:0] bits = 0x1

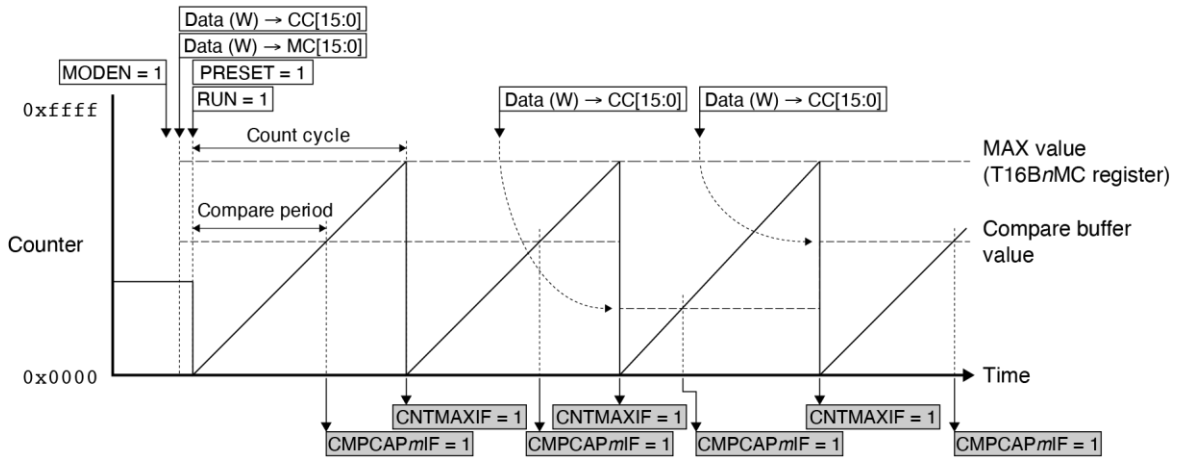


14 16-bit PWM Timers (T16B)

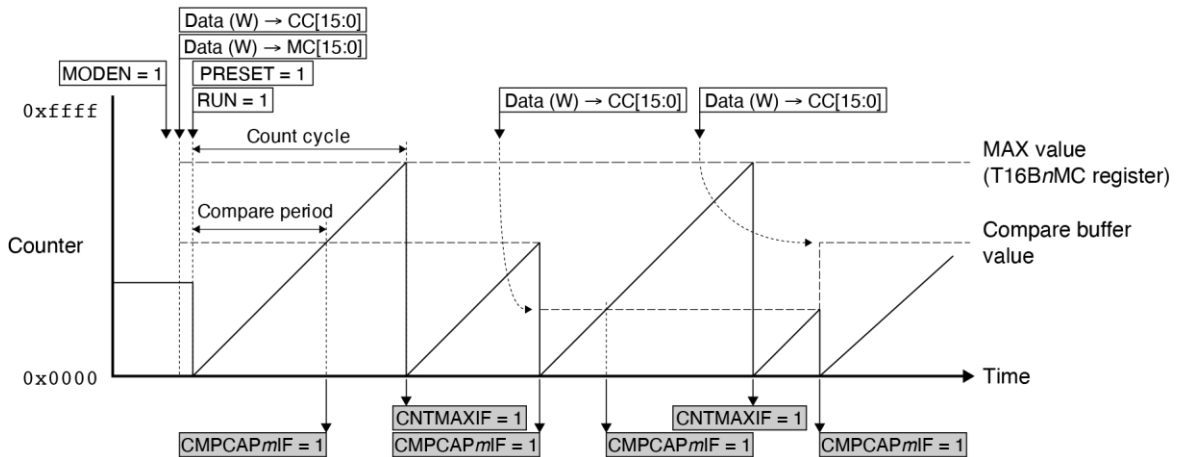
(1.3) T16BnCCCTLm.CBUFMD[2:0] bits = 0x2



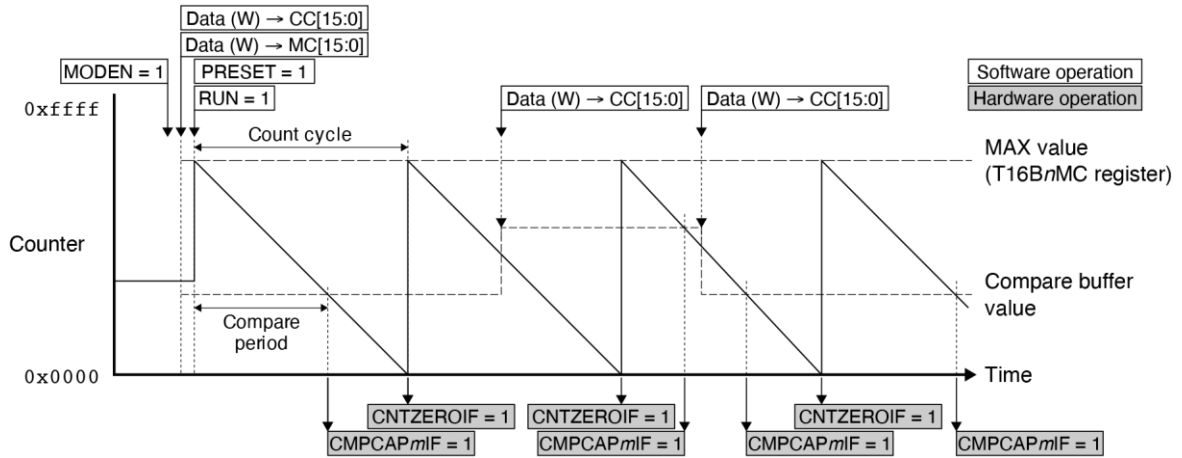
(1.4) T16BnCCCTLm.CBUFMD[2:0] bits = 0x3



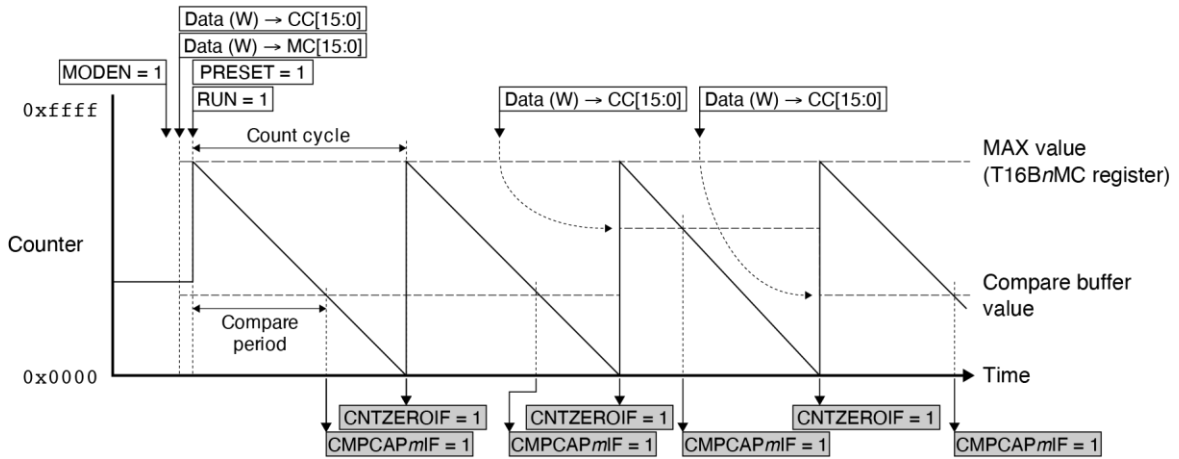
(1.5) T16BnCCCTLm.CBUFMD[2:0] bits = 0x4



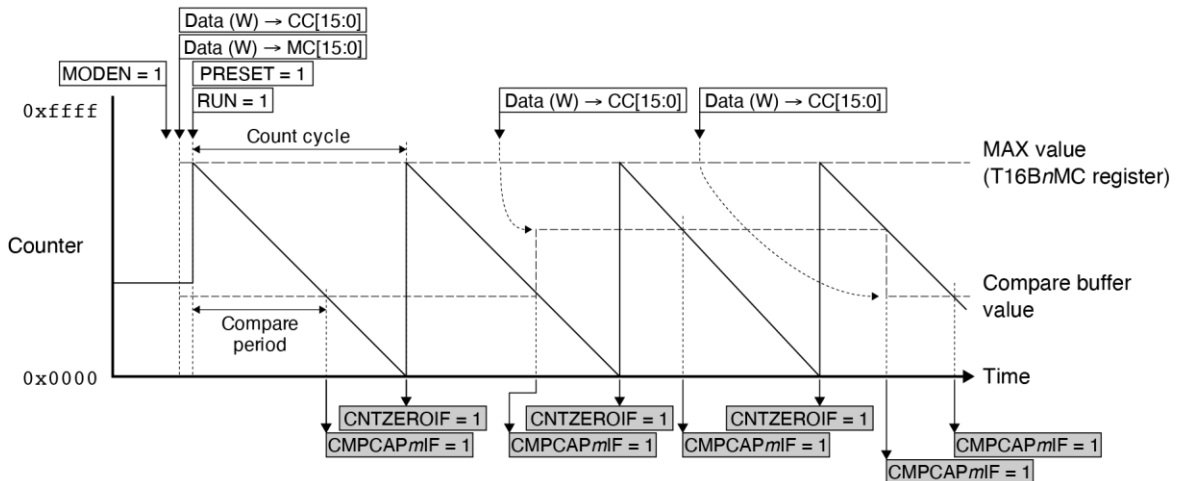
(2) Repeat down count mode
 (2.1) T16BnCCCTLm.CBUFMD[2:0] bits = 0x0



(2.2) T16BnCCCTLm.CBUFMD[2:0] bits = 0x1

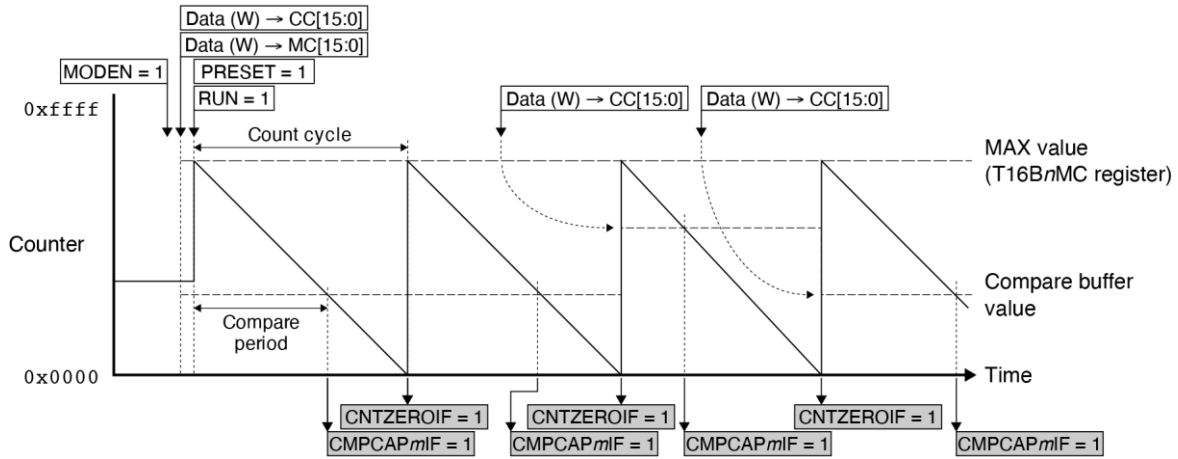


(2.3) T16BnCCCTLm.CBUFMD[2:0] bits = 0x2

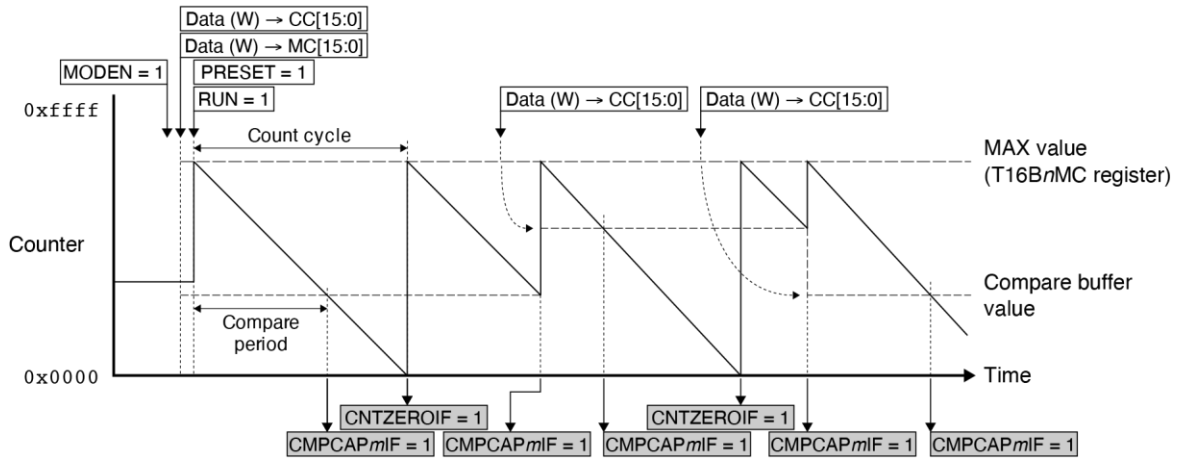


14 16-bit PWM Timers (T16B)

(2.4) T16BnCCCTLm.CBUFMD[2:0] bits = 0x3

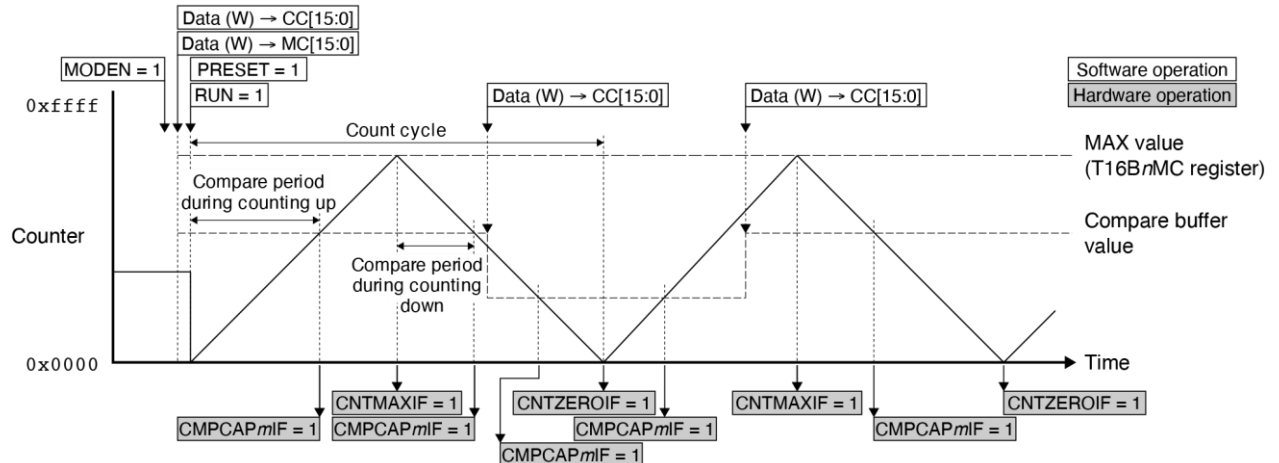


(2.5) T16BnCCCTLm.CBUFMD[2:0] bits = 0x4

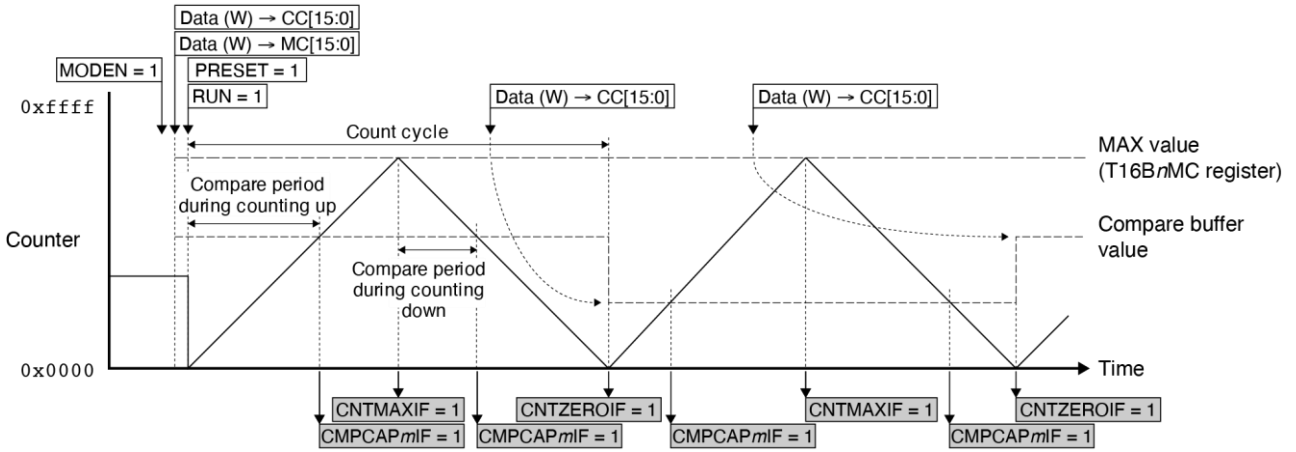


(3) Repeat up/down count mode

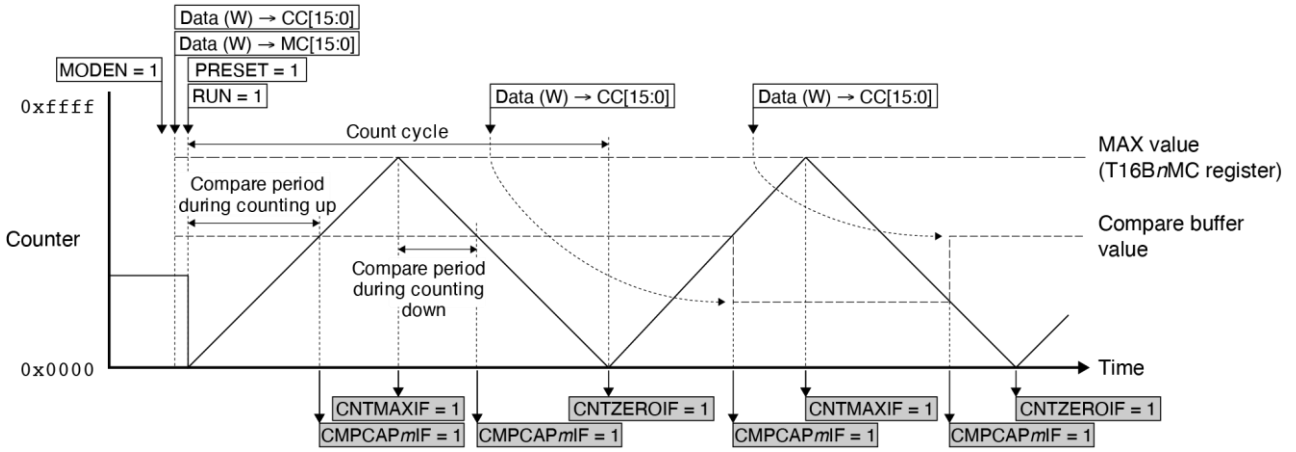
(3.1) T16BnCCCTLm.CBUFMD[2:0] bits = 0x0



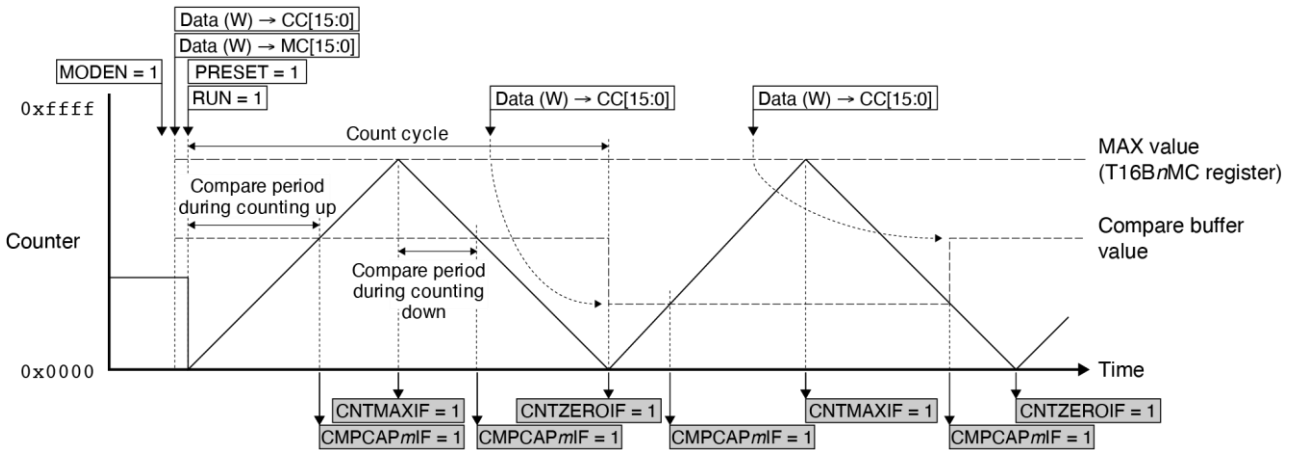
(3.2) T16BnCCCTLm.CBUFMD[2:0] bits = 0x1



(3.3) T16BnCCCTLm.CBUFMD[2:0] bits = 0x2

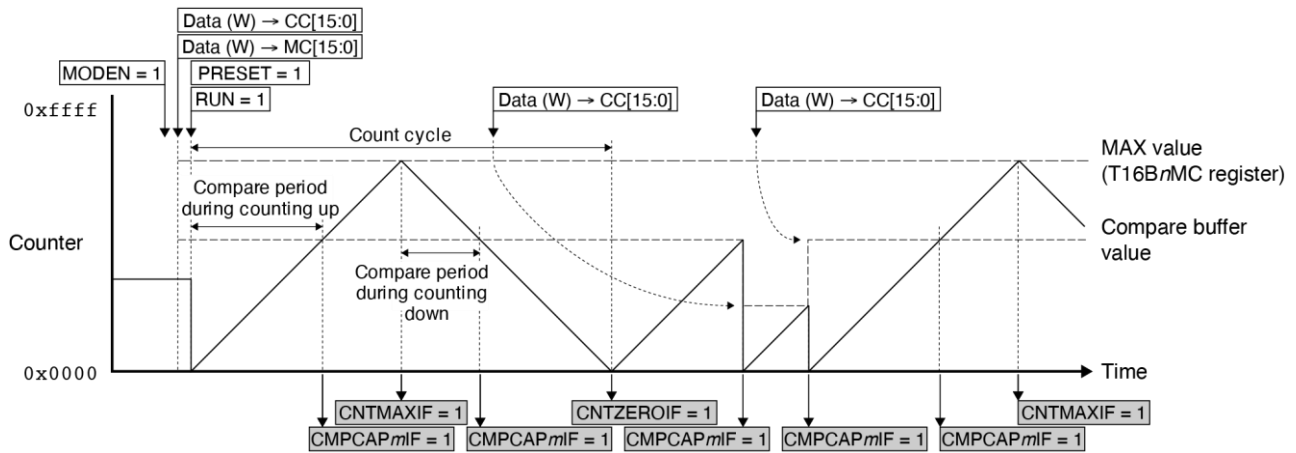


(3.4) T16BnCCCTLm.CBUFMD[2:0] bits = 0x3



14 16-bit PWM Timers (T16B)

(3.5) T16BnCCCTLm.CBUFMD[2:0] bits = 0x4



(Note that the T16BnINTF.CMPCAPmIF/CNTMAXIF/CNTZEROIF bit clearing operations via software are omitted from the figure.)

Figure 14.4.3.2 Compare Buffer Operations

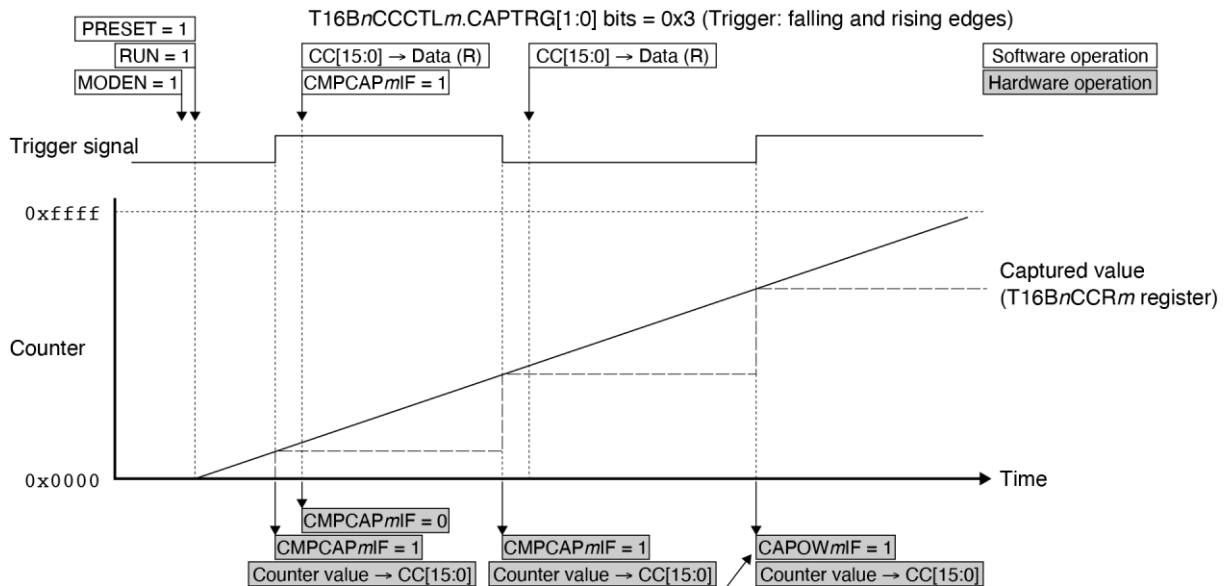
Operations in capture mode

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/software trigger signal). In this mode, the T16BnCCCRm register functions as the capture register from which the captured data is read. Furthermore, the TOUTnm/CAPnm pin is configured to the CAPnm pin.

The trigger signal and the trigger edge to capture the counter value are selected using the T16BnCCCTLm.CAPIS[1:0] bits and the T16BnCCCTLm.CAPTRG[1:0] bits, respectively.

When a specified trigger edge is input during counting, the current counter value is loaded to the T16BnCCCRm register. At the same time the T16BnINTF.CMPCAPmIF bit is set. The interrupt occurred by this bit can be used to read the captured data from the T16BnCCCRm 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 stored in the T16BnCCCRm register is overwritten by the next trigger when the T16BnINTF.CMPCAPmIF bit is still set, an overwrite error occurs (the T16BnINTF.CAPOWmIF bit is set).



An overwrite error occurs as the T16BnINTF.CMPCAPmIF bit has not been cleared.

Figure 14.4.3.3 Operations in Capture Mode (Example in One-shot Up Count Mode)

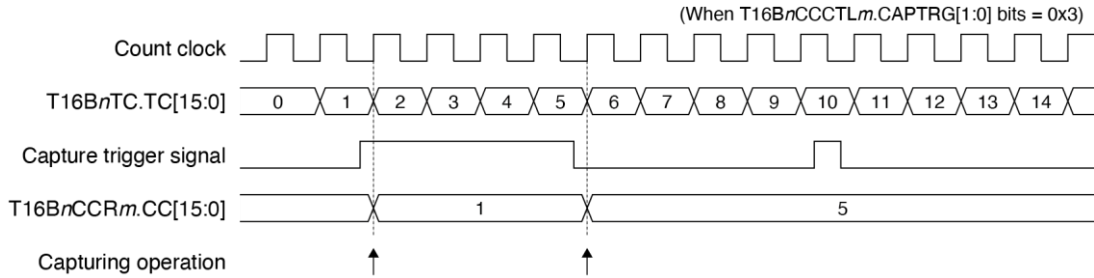
Synchronous capture mode/asynchronous capture mode

The capture circuit can operate in two operating modes: synchronous capture mode and asynchronous capture mode.

Synchronous capture mode is provided to avoid the possibility of invalid data reading by capturing counter data simultaneously with the counter being counted up/down. Set the T16BnCCCTLm.SCS bit to 1 to set the capture circuit to synchronous capture mode. This mode captures counter data by synchronizing the capture signal with the counter clock.

On the other hand, asynchronous capture mode can capture counter data by detecting a trigger pulse even if the pulse is shorter than the counter clock cycle that becomes invalid in synchronous capture mode. Set the T16BnCCCTLm.SCS bit to 0 to set the capture circuit to asynchronous capture mode.

(1) Synchronous capture mode



(2) Asynchronous capture mode

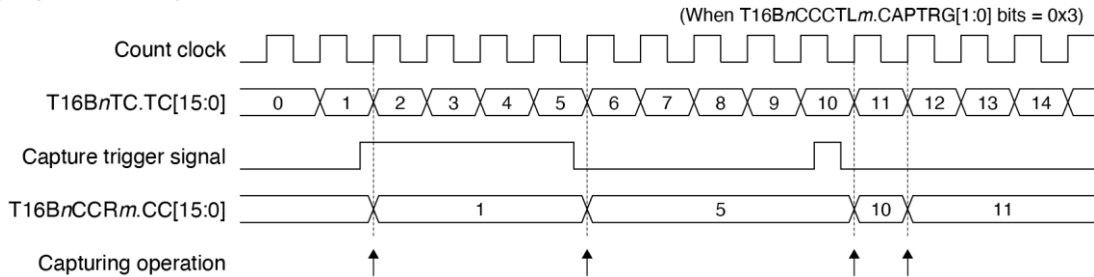


Figure 14.4.3.4 Synchronous Capture Mode/Asynchronous Capture Mode

14.4.4 TOUT Output Control

Comparator mode can generate TOUT signals using the comparator MATCH and counter MAX/ZERO signals. The generated signals can be output to outside the IC. Figure 14.4.4.1 shows the TOUT output circuits (circuits 0 and 1).

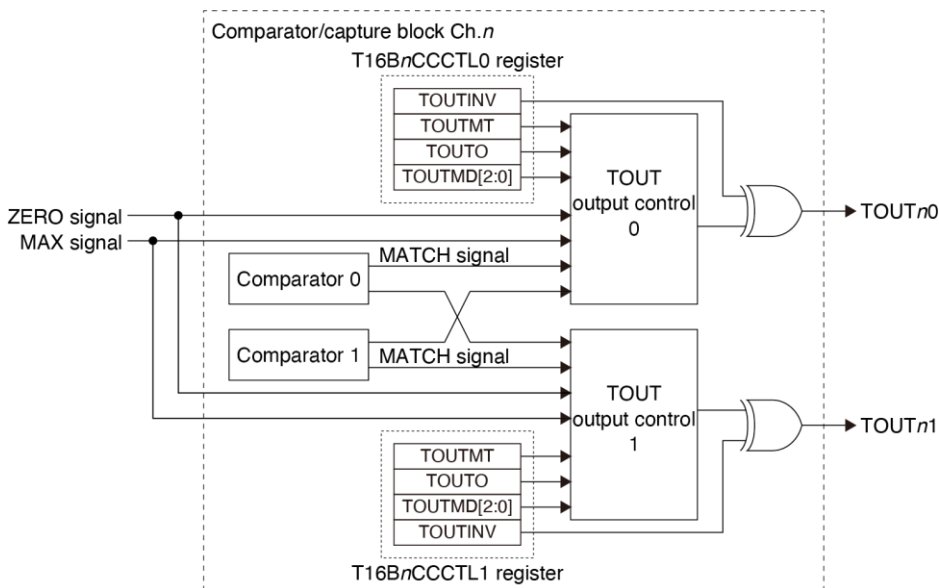


Figure 14.4.4.1 TOUT Output Circuits (Circuits 0 and 1)

14 16-bit PWM Timers (T16B)

Each timer channel includes two (four, or six) TOUT output circuits and their signal generation and output can be controlled individually.

TOUT generation mode

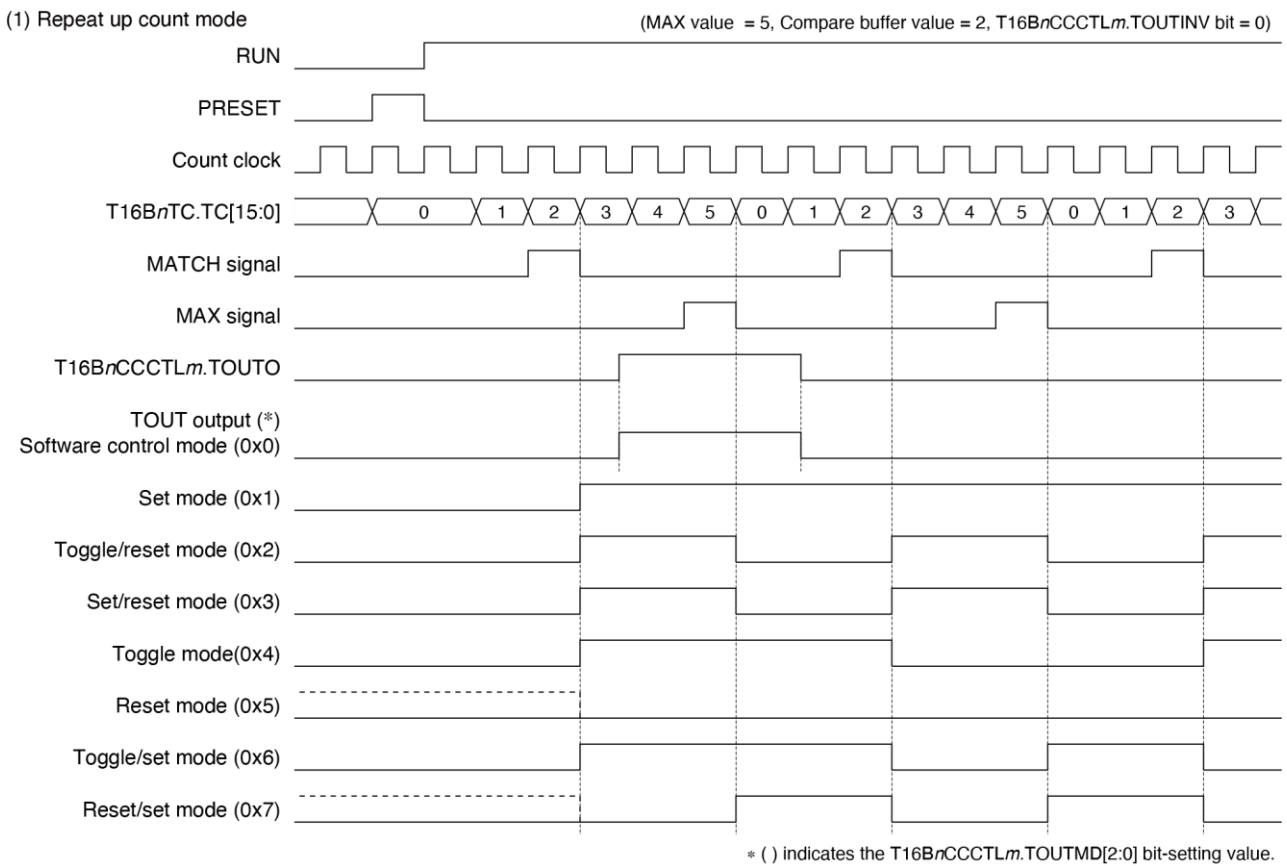
The $T16BnCCCTLm.TOUTMD[2:0]$ bits are used to set how the TOUT signal waveform is changed by the MATCH and MAX/ZERO signals.

Furthermore, when the $T16BnCCCTLm.TOUTMT$ bit is set to 1, the TOUT circuit uses the MATCH signal output from another system in the circuit pair (0 and 1, 2 and 3, 4 and 5). This makes it possible to change the signal twice within a counter cycle.

TOUT signal polarity

The TOUT signal polarity (active level) can be set using the $T16BnCCCTLm.TOUTINV$ bit. It is set to active high by setting the $T16BnCCCTLm.TOUTINV$ bit to 0 and active low by setting to 1.

Figure 14.4.4.2 and Figure 14.4.4.3 show the TOUT output waveforms.



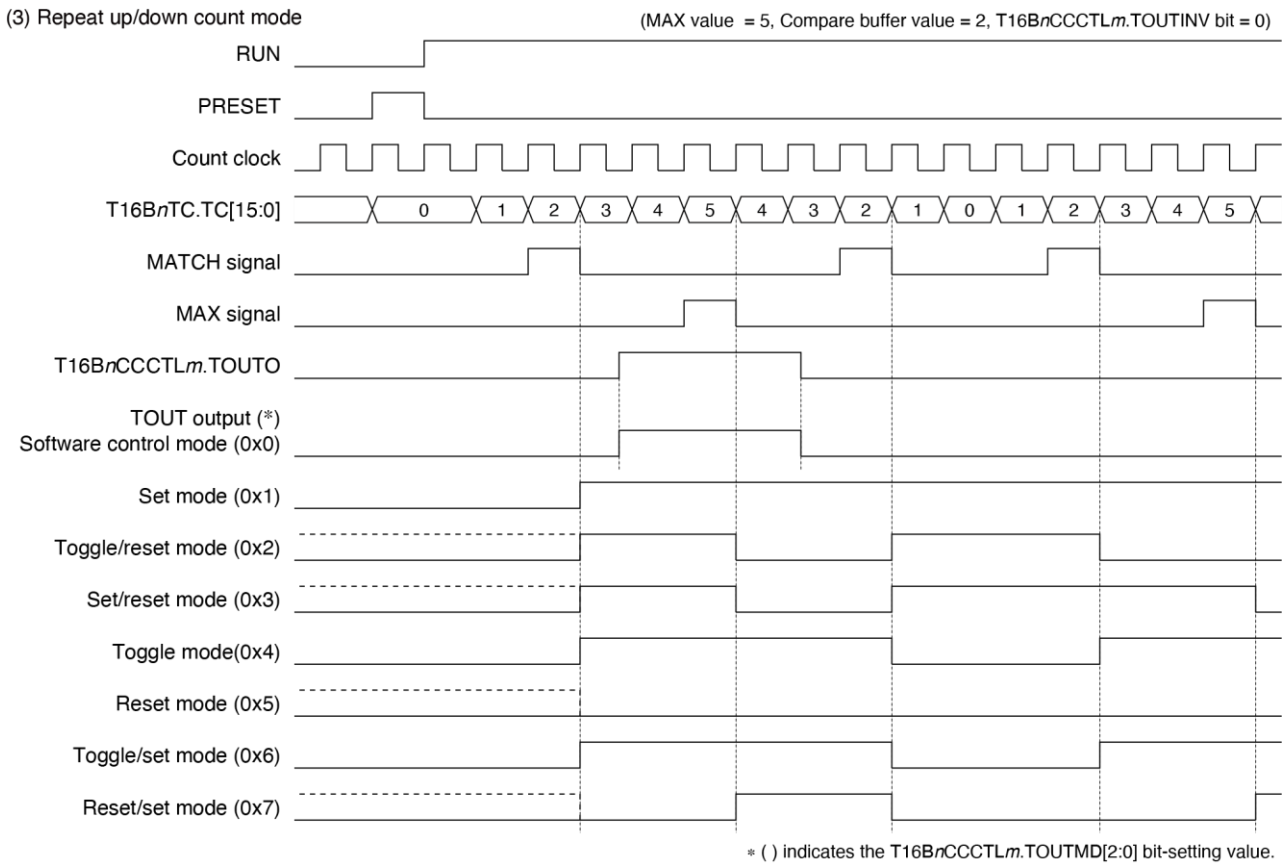
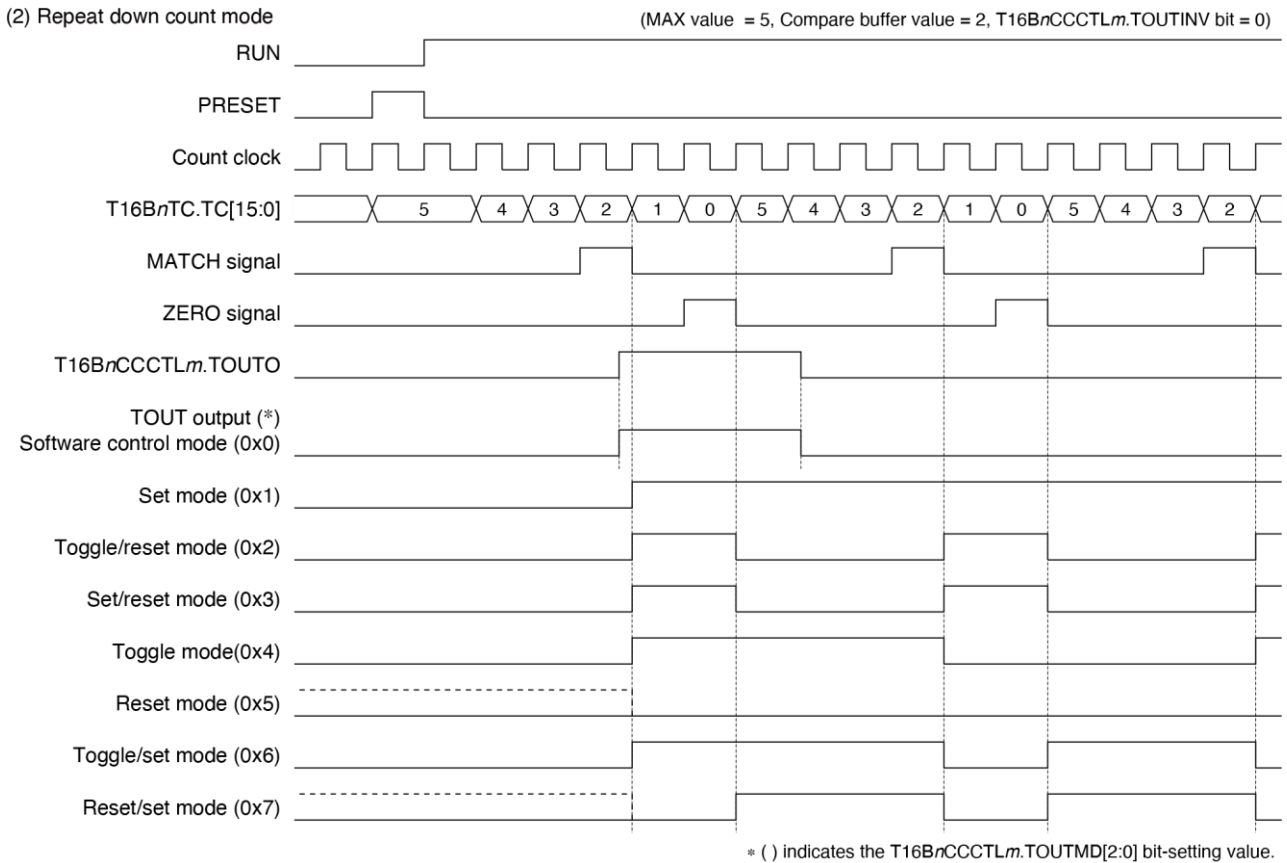
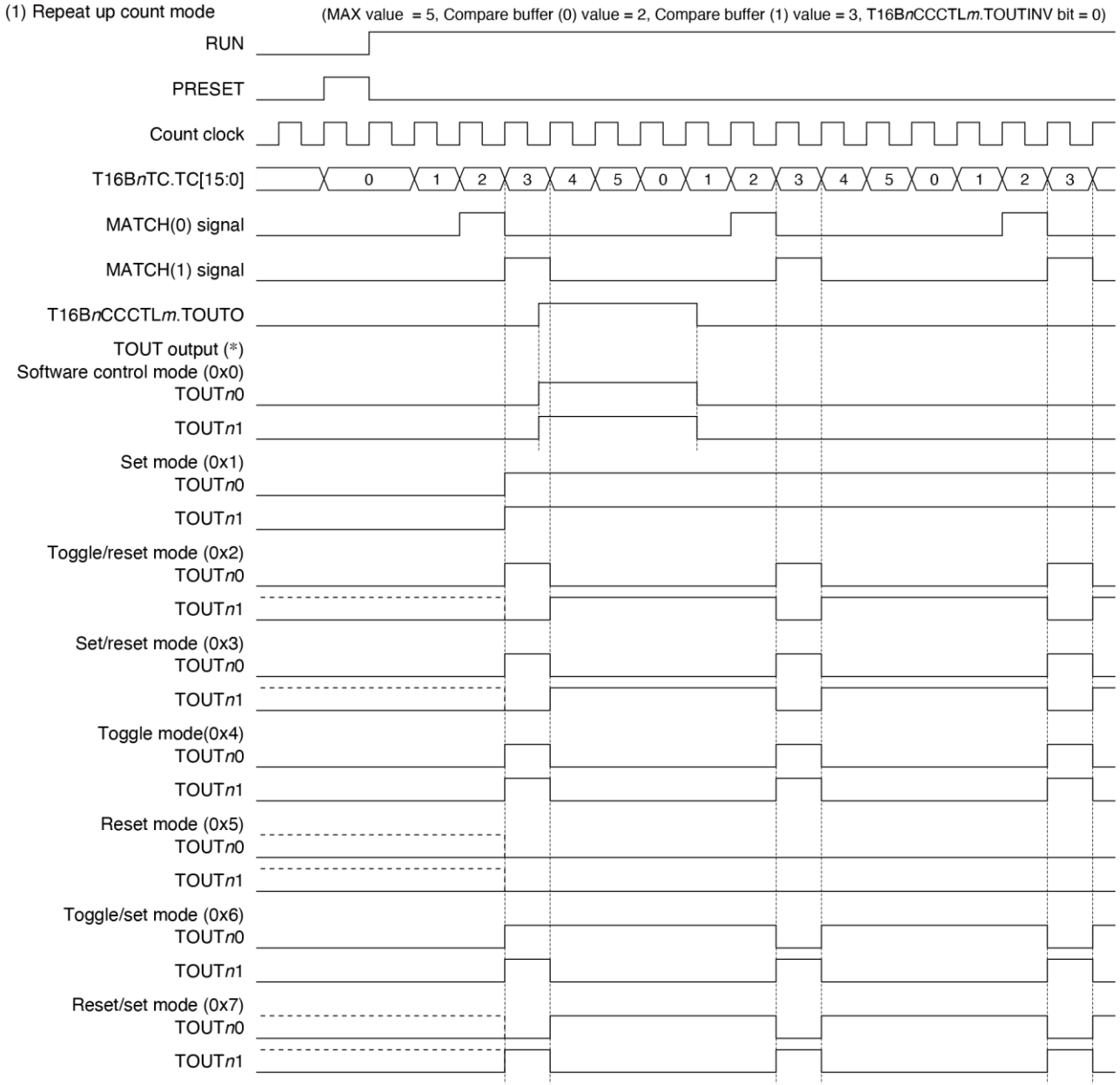
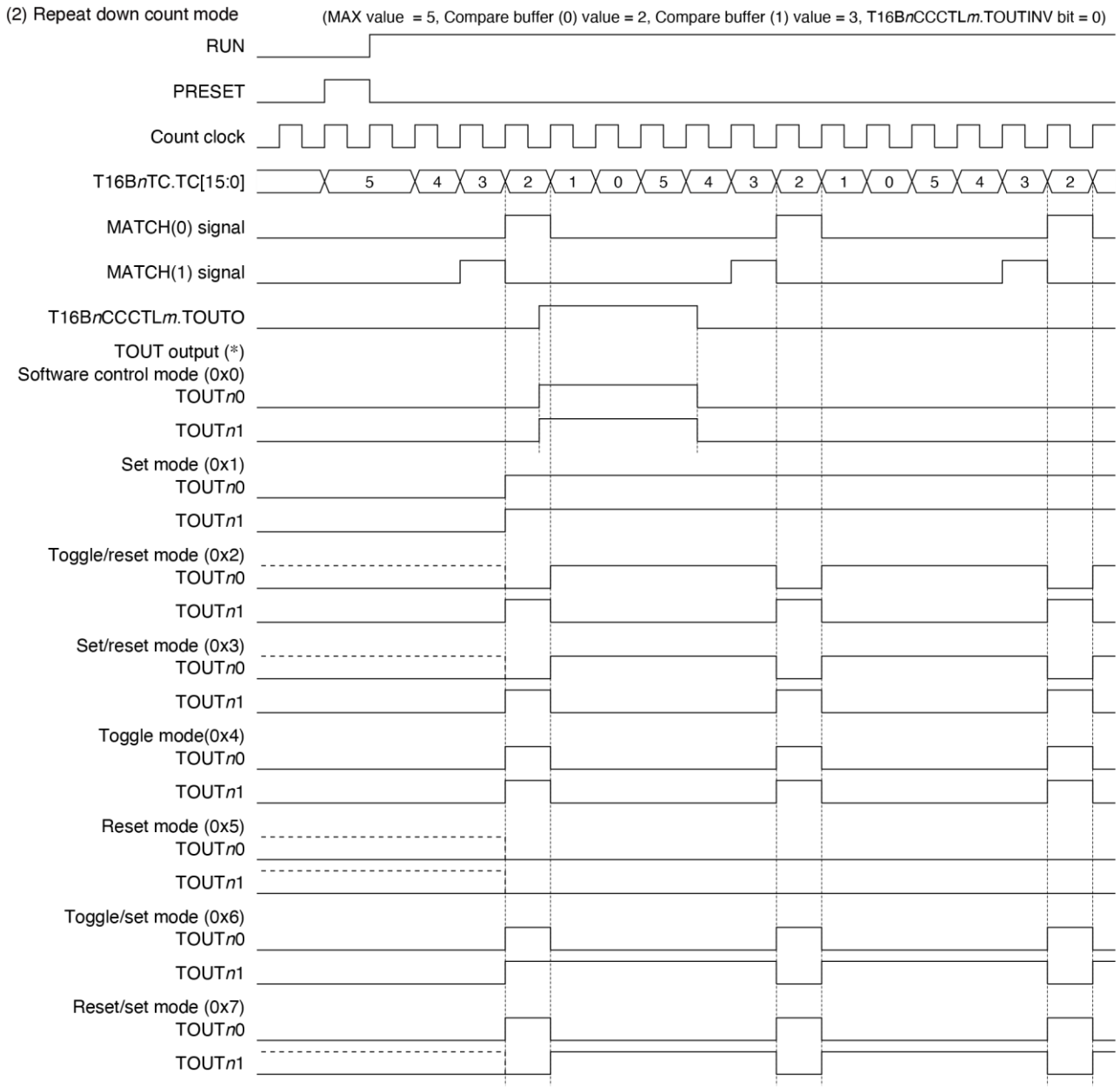


Figure 14.4.4.2 TOUT Output Waveform (T16BnCCCTLm.TOUTMT bit = 0)

14 16-bit PWM Timers (T16B)



* () indicates the T16BrCCCTLm.TOUTMD[2:0] bit-setting value.



* () indicates the T16BrCCCTLm.TOUTMD[2:0] bit-setting value.

14 16-bit PWM Timers (T16B)

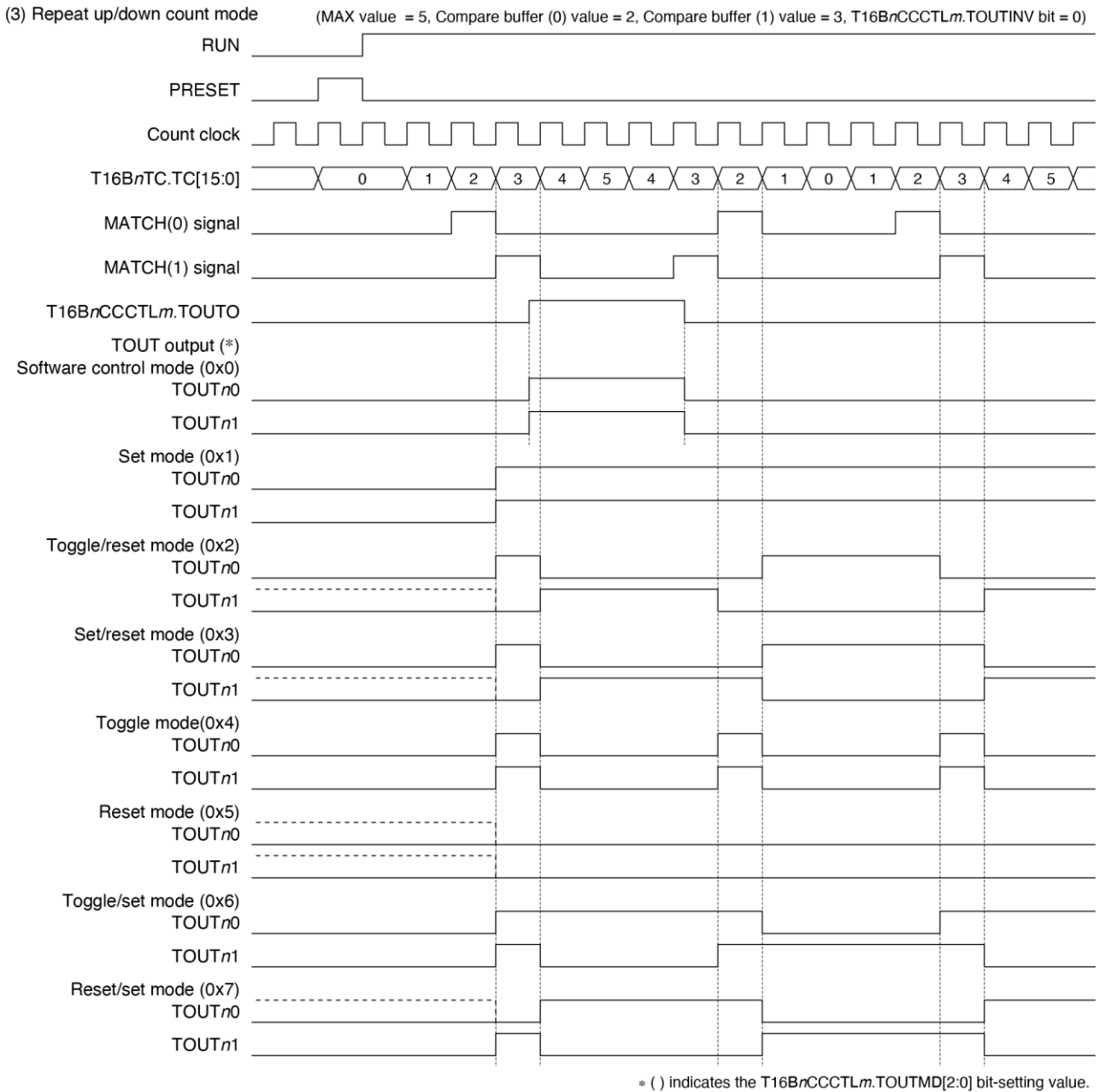


Figure 14.4.4.3 TOUT Output Waveform (T16BnCCCTL0.TOUTMT bit = 1, T16BnCCCTL1.TOUTMT bit = 0)

14.5 Interrupt

Each T16B channel has a function to generate the interrupt shown in Table 14.5.1.

Table 14.5.1 T16B Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Capture overwrite	T16BnINTF.CAPOWmIF	When the T16BnINTF.CMPCAPmIF bit = 1 and the T16BnCCRm register is overwritten with new captured data in capture mode	Writing 1
Compare/capture	T16BnINTF.CMPCAPmIF	When the counter value becomes equal to the compare buffer value in comparator mode When the counter value is loaded to the T16BnCCRm register by a capture trigger input in capture mode	Writing 1
Counter MAX	T16BnINTF.CNTMAXIF	When the counter reaches the MAX value	Writing 1
Counter zero	T16BnINTF.CNTZEROIF	When the counter reaches 0x0000	Writing 1

T16B provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

14.6 Control Registers

T16B Ch.n Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B _n CLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–4	CLKDIV[3:0]	0x0	H0	R/W	
	3	–	0	–	R	
	2–0	CLKSRC[2:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 **DBRUN**

This bit sets whether the T16B Ch.n operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–4 **CLKDIV[3:0]**

These bits select the division ratio of the T16B Ch.n operating clock (counter clock).

Bit 3 Reserved

Bits 2–0 **CLKSRC[2:0]**

These bits select the clock source of T16B Ch.n.

Table 14.6.1 Clock Source and Division Ratio Settings

T16B _n CLK. CLKDIV[3:0] bits	T16B _n CLK.CLKSRC[2:0] bits							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
	IOSC	OSC1	OSC3	EXOSC	EXCL _n 0	EXCL _n 1	EXCL _n 0 inverted input	EXCL _n 1 inverted input
0xf	1/32,768	1/1	1/32,768	1/1	1/1	1/1	1/1	1/1
0xe	1/16,384		1/16,384					
0xd	1/8,192		1/8,192					
0xc	1/4,096		1/4,096					
0xb	1/2,048		1/2,048					
0xa	1/1,024		1/1,024					
0x9	1/512		1/512					
0x8	1/256	1/256	1/256					
0x7	1/128	1/128	1/128					
0x6	1/64	1/64	1/64					
0x5	1/32	1/32	1/32					
0x4	1/16	1/16	1/16					
0x3	1/8	1/8	1/8					
0x2	1/4	1/4	1/4					
0x1	1/2	1/2	1/2					
0x0	1/1	1/1	1/1					

(Note) The oscillator circuits/external inputs that are not supported in this IC cannot be selected as the clock source.

14 16-bit PWM Timers (T16B)

T16B n Counter Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B n CTL	15–9	–	0x00	–	R	–
	8	MAXBSY	0	H0	R	
	7–6	–	0x0	–	R	
	5–4	CNTMD[1:0]	0x0	H0	R/W	
	3	ONEST	0	H0	R/W	
	2	RUN	0	H0	R/W	
	1	PRESET	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 MAXBSY

This bit indicates whether data can be written to the T16B n MC register or not.

1 (R): Busy status (cannot be written)

0 (R): Idle (can be written)

While this bit is 1, the T16B n MC register is loading the MAX value. Data writing is prohibited during this period.

Bits 7–6 Reserved

Bits 5–4 CNTMD[1:0]

These bits select the counter up/down mode. The count mode is configured with this selection and the T16B n CTL.ONEST bit setting (see Table 14.6.2).

Bit 3 ONEST

This bit selects the counter repeat/one-shot mode. The count mode is configured with this selection and the T16B n CTL.CNTMD[1:0] bit settings (see Table 14.6.2).

Table 14.6.2 Count Mode

T16B n CTL.CNTMD[1:0] bits	Count mode	
	T16B n CTL.ONEST bit = 1	T16B n CTL.ONEST bit = 0
0x3	Reserved	
0x2	One-shot up/down count mode	Repeat up/down count mode
0x1	One-shot down count mode	Repeat down count mode
0x0	One-shot up count mode	Repeat up count mode

Bit 2 RUN

This bit starts/stops counting.

1 (W): Start counting

0 (W): Stop counting

1 (R): Counting

0 (R): Idle

By writing 1 to this bit, the counter block starts count operations. However, the T16B n CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance. While the timer is running, writing 0 to the T16B n CTL.RUN bit stops count operations. When the counter stops by the counter MAX/ZERO signal in one-shot mode, this bit is automatically cleared to 0.

Bit 1 PRESET

This bit resets the counter.

1 (W): Reset

0 (W): Ineffective

1 (R): Resetting in progress

0 (R): Resetting finished or normal operation

In up mode or up/down mode, the counter is cleared to 0x0000 by writing 1 to this bit. In down mode, the MAX value, which has been set to the T16B n MC register, is preset to the counter. However, the T16B n CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance.

Bit 0 MODEN

This bit enables the T16B Ch.*n* operations.

1 (R/W): Enable (Start supplying operating clock)

0 (R/W): Disable (Stop supplying operating clock)

Note: The counter reset operation using the T16B*n*CTL.PRESET bit and the counting start operation using the T16B*n*CTL.RUN bit take effect only when the T16B*n*CTL.MODEN bit = 1.

T16B Ch.*n* Max Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B <i>n</i> MC	15–0	MC[15:0]	0xffff	H0	R/W	–

Bits 15–0 MC[15:0]

These bits are used to set the MAX value to preset to the counter. For more information, refer to “Counter Block Operations - MAX counter data register.”

- Notes:
- When one-shot mode is selected, do not alter the T16B*n*MC.MC[15:0] bits (MAX value) during counting.
 - Make sure the T16B*n*CTL.MODEN bit is set to 1 before writing data to the T16B*n*MC.MC[15:0] bits. If the T16B*n*CTL.MODEN bit = 0 when writing to the T16B*n*MC.MC[15:0] bits, set the T16B*n*CTL.MODEN bit to 1 until the T16B*n*CS.BSY bit is set to 0 from 1.
 - Do not set the T16B*n*MC.MC[15:0] bits to 0x0000.

T16B Ch.*n* Timer Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B <i>n</i> TC	15–0	TC[15:0]	0x0000	H0	R	–

Bits 15–0 TC[15:0]

The current counter value can be read out through these bits.

T16B Ch.*n* Counter Status Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B <i>n</i> CS	15–8	–	0x00	–	R	–
	7	CAP15	0	H0	R	
	6	CAP14	0	H0	R	
	5	CAP13	0	H0	R	
	4	CAP12	0	H0	R	
	3	CAP11	0	H0	R	
	2	CAP10	0	H0	R	
	1	UP_DOWN	1	H0	R	
0	BSY	0	H0	R		

Bits 15–8 Reserved**Bit 7 CAP15****Bit 6 CAP14****Bit 5 CAP13****Bit 4 CAP12****Bit 3 CAP11****Bit 2 CAP10**

These bits indicate the signal level currently input to the CAP*nm* pin.

1 (R): Input signal = High level

0 (R): Input signal = Low level

14 16-bit PWM Timers (T16B)

The following shows the correspondence between the bit and the CAP_{nm} pin:

T16B n CS.CAPI5 bit: CAP n 5 pin

T16B n CS.CAPI4 bit: CAP n 4 pin

T16B n CS.CAPI3 bit: CAP n 3 pin

T16B n CS.CAPI2 bit: CAP n 2 pin

T16B n CS.CAPI1 bit: CAP n 1 pin

T16B n CS.CAPI0 bit: CAP n 0 pin

Note: The configuration of the T16B n CS.CAPI m bits depends on the model. The bits corresponding to the CAP nm pins that do not exist are read-only bits and are always fixed at 0.

Bit 1 UP_DOWN

This bit indicates the currently set count direction.

1 (R): Count up

0 (R): Count down

Bit 0 BSY

This bit indicates the counter operating status.

1 (R): Running

0 (R): Idle

T16B Ch. n Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B n INTF	15–14	–	0x0	–	R	–
	13	CAPOW5IF	0	H0	R/W	Cleared by writing 1.
	12	CMPCAP5IF	0	H0	R/W	
	11	CAPOW4IF	0	H0	R/W	
	10	CMPCAP4IF	0	H0	R/W	
	9	CAPOW3IF	0	H0	R/W	
	8	CMPCAP3IF	0	H0	R/W	
	7	CAPOW2IF	0	H0	R/W	
	6	CMPCAP2IF	0	H0	R/W	
	5	CAPOW1IF	0	H0	R/W	
	4	CMPCAP1IF	0	H0	R/W	
	3	CAPOW0IF	0	H0	R/W	
	2	CMPCAP0IF	0	H0	R/W	
1	CNTMAXIF	0	H0	R/W		
0	CNTZEROIF	0	H0	R/W		

Bits 15–14 Reserved

Bit 13 CAPOW5IF

Bit 12 CMPCAP5IF

Bit 11 CAPOW4IF

Bit 10 CMPCAP4IF

Bit 9 CAPOW3IF

Bit 8 CMPCAP3IF

Bit 7 CAPOW2IF

Bit 6 CMPCAP2IF

Bit 5 CAPOW1IF

Bit 4 CMPCAP1IF

Bit 3 CAPOW0IF

Bit 2 CMPCAP0IF

Bit 1 CNTMAXIF

Bit 0 CNTZEROIF

These bits indicate the T16B Ch. n interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

T16B n INTF.CAPOW5IF bit: Capture 5 overwrite interrupt
 T16B n INTF.CMPCAP5IF bit: Compare/capture 5 interrupt
 T16B n INTF.CAPOW4IF bit: Capture 4 overwrite interrupt
 T16B n INTF.CMPCAP4IF bit: Compare/capture 4 interrupt
 T16B n INTF.CAPOW3IF bit: Capture 3 overwrite interrupt
 T16B n INTF.CMPCAP3IF bit: Compare/capture 3 interrupt
 T16B n INTF.CAPOW2IF bit: Capture 2 overwrite interrupt
 T16B n INTF.CMPCAP2IF bit: Compare/capture 2 interrupt
 T16B n INTF.CAPOW1IF bit: Capture 1 overwrite interrupt
 T16B n INTF.CMPCAP1IF bit: Compare/capture 1 interrupt
 T16B n INTF.CAPOW0IF bit: Capture 0 overwrite interrupt
 T16B n INTF.CMPCAP0IF bit: Compare/capture 0 interrupt
 T16B n INTF.CNTMAXIF bit: Counter MAX interrupt
 T16B n INTF.CNTZEROIF bit: Counter zero interrupt

Note: The configuration of the T16B n INTF.CAPOW m IF and T16B n INTF.CMPCAP m IF bits depends on the model. The bits corresponding to the comparator/capture circuits that do not exist are read-only bits and are always fixed at 0.

T16B Ch. n Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B n INTE	15–14	–	0x0	–	R	–
	13	CAPOW5IE	0	H0	R/W	
	12	CMPCAP5IE	0	H0	R/W	
	11	CAPOW4IE	0	H0	R/W	
	10	CMPCAP4IE	0	H0	R/W	
	9	CAPOW3IE	0	H0	R/W	
	8	CMPCAP3IE	0	H0	R/W	
	7	CAPOW2IE	0	H0	R/W	
	6	CMPCAP2IE	0	H0	R/W	
	5	CAPOW1IE	0	H0	R/W	
	4	CMPCAP1IE	0	H0	R/W	
	3	CAPOW0IE	0	H0	R/W	
	2	CMPCAP0IE	0	H0	R/W	
1	CNTMAXIE	0	H0	R/W		
0	CNTZEROIE	0	H0	R/W		

Bits 15–14 Reserved

Bit 13 CAPOW5IE
 Bit 12 CMPCAP5IE
 Bit 11 CAPOW4IE
 Bit 10 CMPCAP4IE
 Bit 9 CAPOW3IE
 Bit 8 CMPCAP3IE
 Bit 7 CAPOW2IE
 Bit 6 CMPCAP2IE
 Bit 5 CAPOW1IE
 Bit 4 CMPCAP1IE
 Bit 3 CAPOW0IE
 Bit 2 CMPCAP0IE
 Bit 1 CNTMAXIE
 Bit 0 CNTZEROIE

These bits enable T16B Ch. n interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

14 16-bit PWM Timers (T16B)

The following shows the correspondence between the bit and interrupt:

T16BnINTE.CAPOW5IE bit: Capture 5 overwrite interrupt
 T16BnINTE.CMPCAP5IE bit: Compare/capture 5 interrupt
 T16BnINTE.CAPOW4IE bit: Capture 4 overwrite interrupt
 T16BnINTE.CMPCAP4IE bit: Compare/capture 4 interrupt
 T16BnINTE.CAPOW3IE bit: Capture 3 overwrite interrupt
 T16BnINTE.CMPCAP3IE bit: Compare/capture 3 interrupt
 T16BnINTE.CAPOW2IE bit: Capture 2 overwrite interrupt
 T16BnINTE.CMPCAP2IE bit: Compare/capture 2 interrupt
 T16BnINTE.CAPOW1IE bit: Capture 1 overwrite interrupt
 T16BnINTE.CMPCAP1IE bit: Compare/capture 1 interrupt
 T16BnINTE.CAPOW0IE bit: Capture 0 overwrite interrupt
 T16BnINTE.CMPCAP0IE bit: Compare/capture 0 interrupt
 T16BnINTE.CNTMAXIE bit: Counter MAX interrupt
 T16BnINTE.CNTZEROIE bit: Counter zero interrupt

- Notes:
- The configuration of the T16BnINTE.CAPOWmIE and T16BnINTE.CMPCAPmIE bits depends on the model. The bits corresponding to the comparator/capture circuits that do not exist are read-only bits and are always fixed at 0.
 - To prevent generating unnecessary interrupts, the corresponding interrupt flag should be cleared before enabling interrupts.

T16B Ch.n Comparator/Capture m Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16BnCCCTLm	15	SCS	0	H0	R/W	-
	14–12	CBUFMD[2:0]	0x0	H0	R/W	
	11–10	CAPIS[1:0]	0x0	H0	R/W	
	9–8	CAPTRG[1:0]	0x0	H0	R/W	
	7	–	0	–	R	
	6	TOUTMT	0	H0	R/W	
	5	TOUTO	0	H0	R/W	
	4–2	TOUTMD[2:0]	0x0	H0	R/W	
	1	TOUTINV	0	H0	R/W	
0	CCMD	0	H0	R/W		

Bit 15 SCS

This bit selects either synchronous capture mode or asynchronous capture mode.

- 1 (R/W): Synchronous capture mode
 0 (R/W): Asynchronous capture mode

For more information, refer to “Comparator/Capture Block Operations - Synchronous capture mode/asynchronous capture mode.” The T16BnCCCTLm.SCS bit is control bit for capture mode and is ineffective in comparator mode.

Bits 14–12 CBUFMD[2:0]

These bits select the timing to load the comparison value written in the T16BnCCRm register to the compare buffer. The T16BnCCCTLm.CBUFMD[2:0] bits are control bits for comparator mode and are ineffective in capture mode.

Table 14.6.3 Timings to Load Comparison Value to Compare Buffer

T16BnCCCTLm.CBUFMD[2:0] bits	Count mode	Comparison Value load timing
0x7–0x5	Reserved	
0x4	Up mode	When the counter becomes equal to the comparison value set previously Also the counter is reset to 0x0000 simultaneously.
	Down mode	When the counter becomes equal to the comparison value set previously Also the counter is reset to the MAX value simultaneously.
	Up/down mode	When the counter becomes equal to the comparison value set previously Also the counter is reset to 0x0000 simultaneously.
0x3	Up mode	When the counter reverts to 0x0000
	Down mode	When the counter reverts to the MAX value
	Up/down mode	When the counter becomes equal to the comparison value set previously or when the counter reverts to 0x0000
0x2	Up mode	When the counter becomes equal to the comparison value set previously
	Down mode	
	Up/down mode	
0x1	Up mode	When the counter reaches the MAX value
	Down mode	When the counter reaches 0x0000
	Up/down mode	When the counter reaches 0x0000 or the MAX value
0x0	Up mode	At the CLK_T16Bn rising edge after writing to the T16BnCCRm register
	Down mode	
	Up/down mode	

Bits 11–10 CAPIS[1:0]

These bits select the trigger signal for capturing (see Table 14.6.4). The T16BnCCCTLm.CAPIS[1:0] bits are control bits for capture mode and are ineffective in comparator mode.

Bits 9–8 CAPTRG[1:0]

These bits select the trigger edge(s) of the trigger signal at which the counter value is captured in the T16BnCCRm register in capture mode (see Table 14.6.4). The T16BnCCCTLm.CAPTRG[1:0] bits are control bits for capture mode and are ineffective in comparator mode.

Table 14.6.4 Trigger Signal/Edge for Capturing Counter Value

T16BnCCCTLm.CAPTRG[1:0] bits (Trigger edge)	Trigger condition		
	T16BnCCCTLm.CAPIS[1:0] bits (Trigger signal)		
	0x0 (External trigger signal)	0x2 (Software trigger signal = L)	0x3 (Software trigger signal = H)
0x3 (↑ & ↓)	Rising or falling edge of the CAPnm pin input signal	Altering the T16BnCCCTLm.CAPIS[1:0] bits from 0x2 to 0x3, or from 0x3 to 0x2	
0x2 (↓)	Falling edge of the CAPnm pin input signal	Altering the T16BnCCCTLm.CAPIS[1:0] bits from 0x3 to 0x2	
0x1 (↑)	Rising edge of the CAPnm pin input signal	Altering the T16BnCCCTLm.CAPIS[1:0] bits from 0x2 to 0x3	
0x0	Not triggered (disable capture function)		

Bit 7 Reserved**Bit 6 TOUTMT**

This bit selects whether the comparator MATCH signal of another system is used for generating the TOUTnm signal or not.

1 (R/W): Generate TOUT using two comparator MATCH signals of the comparator circuit pair (0 and 1, 2 and 3, 4 and 5)

0 (R/W): Generate TOUT using one comparator MATCH signal of comparator m and the counter MAX or ZERO signals

The T16BnCCCTLm.TOUTMT bit is control bit for comparator mode and is ineffective in capture mode.

Bit 5 TOUTO

This bit sets the TOUTnm signal output level when software control mode (T16BnCCCTLm.TOUTMD[2:0] = 0x0) is selected for the TOUTnm output.

1 (R/W): High level output

0 (R/W): Low level output

The T16BnCCCTLm.TOUTO bit is control bit for comparator mode and is ineffective in capture mode.

14 16-bit PWM Timers (T16B)

Bits 4–2 TOUTMD[2:0]

These bits configure how the TOUT_{nm} signal waveform is changed by the comparator MATCH and counter MAX/ZERO signals.

The T16B_nCCCTL_m.TOUTMD[2:0] bits are control bits for comparator mode and are ineffective in capture mode.

Table 14.6.5 TOUT Generation Mode

T16B _n CCCTL _m . TOUTMD[2:0] bits	TOUT generation mode and operations			
	T16B _n CCCTL _m . TOUTMT bit	Count mode	Output signal	Change in the signal
0x7	Reset/set mode			
	0	Up count mode	TOUT _{nm}	The signal becomes inactive by the MATCH signal and it becomes active by the MAX signal.
		Up/down count mode	TOUT _{nm}	The signal becomes inactive by the MATCH signal and it becomes active by the ZERO signal.
	1	All count modes	TOUT _{nm}	The signal becomes inactive by the MATCH _m signal and it becomes active by the MATCH _{m+1} signal.
TOUT _{nm+1}			The signal becomes inactive by the MATCH _{m+1} signal and it becomes active by the MATCH _m signal.	
0x6	Toggle/set mode			
	0	Up count mode	TOUT _{nm}	The signal is inverted by the MATCH signal and it becomes active by the MAX signal.
		Up/down count mode	TOUT _{nm}	The signal is inverted by the MATCH signal and it becomes active by the ZERO signal.
	1	All count modes	TOUT _{nm}	The signal is inverted by the MATCH _m signal and it becomes active by the MATCH _{m+1} signal.
TOUT _{nm+1}			The signal is inverted by the MATCH _{m+1} signal and it becomes active by the MATCH _m signal.	
0x5	Reset mode			
	0	All count modes	TOUT _{nm}	The signal becomes inactive by the MATCH signal.
	1	All count modes	TOUT _{nm}	The signal becomes inactive by the MATCH _m or MATCH _{m+1} signal.
TOUT _{nm+1}			The signal becomes inactive by the MATCH _{m+1} or MATCH _m signal.	
0x4	Toggle mode			
	0	All count modes	TOUT _{nm}	The signal is inverted by the MATCH signal.
	1	All count modes	TOUT _{nm}	The signal is inverted by the MATCH _m or MATCH _{m+1} signal.
TOUT _{nm+1}			The signal is inverted by the MATCH _{m+1} or MATCH _m signal.	
0x3	Set/reset mode			
	0	Up count mode	TOUT _{nm}	The signal becomes active by the MATCH signal and it becomes inactive by the MAX signal.
		Up/down count mode	TOUT _{nm}	The signal becomes active by the MATCH signal and it becomes inactive by the ZERO signal.
	1	All count modes	TOUT _{nm}	The signal becomes active by the MATCH _m signal and it becomes inactive by the MATCH _{m+1} signal.
TOUT _{nm+1}			The signal becomes active by the MATCH _{m+1} signal and it becomes inactive by the MATCH _m signal.	
0x2	Toggle/reset mode			
	0	Up count mode	TOUT _{nm}	The signal is inverted by the MATCH signal and it becomes inactive by the MAX signal.
		Up/down count mode	TOUT _{nm}	The signal is inverted by the MATCH signal and it becomes inactive by the ZERO signal.
	1	All count modes	TOUT _{nm}	The signal is inverted by the MATCH _m signal and it becomes inactive by the MATCH _{m+1} signal.
TOUT _{nm+1}			The signal is inverted by the MATCH _{m+1} signal and it becomes inactive by the MATCH _m signal.	
0x1	Set mode			
	0	All count modes	TOUT _{nm}	The signal becomes active by the MATCH signal.
	1	All count modes	TOUT _{nm}	The signal becomes active by the MATCH _m or MATCH _{m+1} signal.
TOUT _{nm+1}			The signal becomes active by the MATCH _{m+1} or MATCH _m signal.	
0x0	Software control mode			
	*	All count modes	TOUT _{nm}	The signal becomes active by setting the T16B _n CCCTL _m .TOUTO bit to 1 and it becomes inactive by setting to 0.

Bit 1 TOUTINV

This bit selects the TOUT nm signal polarity.

1 (R/W): Inverted (active low)

0 (R/W): Normal (active high)

The T16B n CCCTL m .TOUTINV bit is control bit for comparator mode and is ineffective in capture mode.

Bit 0 CCMD

This bit selects the operating mode of the comparator/capture circuit m .

1 (R/W): Capture mode (T16B n CCRM register = capture register)

0 (R/W): Comparator mode (T16B n CCRM register = compare data register)

T16B Ch. n Compare/Capture m Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B n CCRM	15–0	CC[15:0]	0x0000	H0	R/W	–

Bits 15–0 CC[15:0]

In comparator mode, this register is configured as the compare data register and used to set the comparison value to be compared with the counter value.

In capture mode, this register is configured as the capture register and the counter value captured by the capture trigger signal is loaded.

15 Sound Generator (SNDA2)

15.1 Overview

SNDA2 is a sound generator that generates melodies and buzzer signals. The features of the SNDA2 are listed below.

- Sound output mode is selectable from three types.
 1. Normal buzzer mode (for normal buzzer output of which the output duration is controlled via software)
 - Output frequency: Can be set within the range of 512 Hz to 16,384 Hz.
 - Duty ratio: Can be set within the range of 0 % to 100 %.
 2. One-shot buzzer mode (for short buzzer output such as a clicking sound)
 - Output frequency: Can be set within the range of 512 Hz to 16,384 Hz.
 - Duty ratio: Can be set within the range of 0 % to 100 %.
 - One-shot output duration: Can be set within the range of 15.6 ms to 250 ms. (16 types)
 3. Melody mode (for playing single note melody)
 - Pitch: Can be set within the range of 128 Hz to 16,384 Hz.
(Scale: 3 octave from C3 to C6 with reference to A4 = 443 Hz)
 - Duration: Can be set within the range of half note/rest to thirty-second note/rest. (7 types)
 - Tempo: Can be set within the range of 30 to 480. (16 types)
 - Other: Tie can be specified.
- A piezoelectric buzzer can be driven with the inverted and non-inverted output pins.
- Can control the non-inverted output pin status while sound stops.

Figure 15.1.1 shows the SNDA2 configuration.

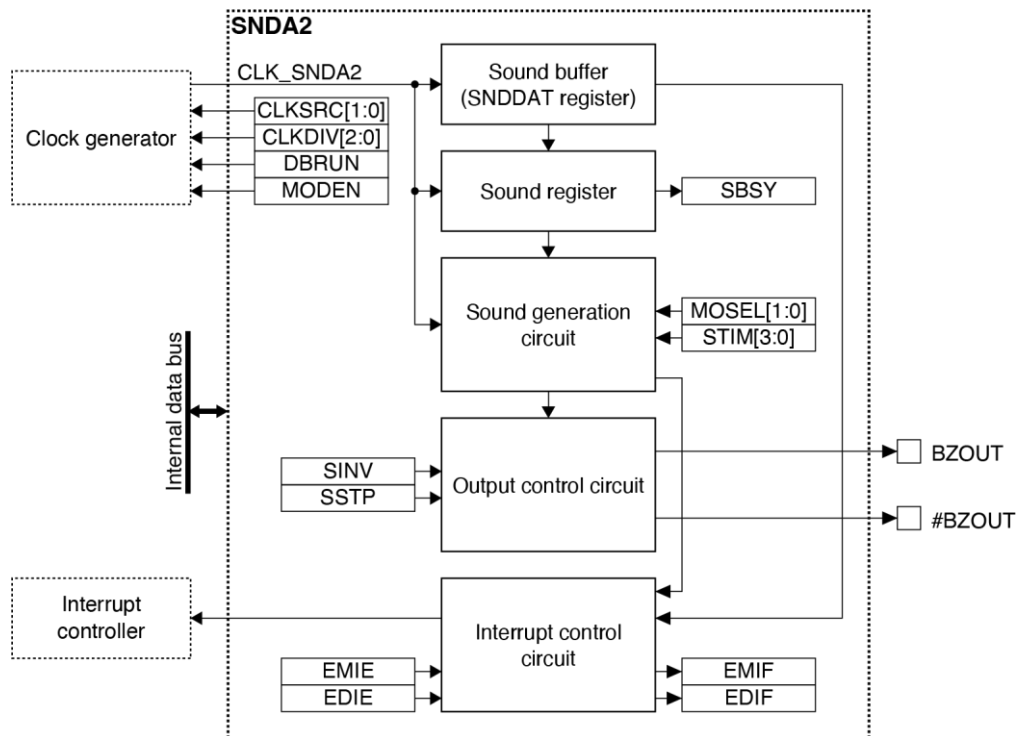


Figure 15.1.1 SNDA2 Configuration

15 Sound Generator (SNDA2)

15.2 Output Pins and External Connections

15.2.1 List of Output Pins

Table 15.2.1 lists the SNDA2 pins.

Table 15.2.1 List of SNDA2 Pins

Pin name	I/O*	Initial status*	Function
BZOUT	O	O (Low)	Non-inverted buzzer output pin
#BZOUT	O	O (Low)	Inverted buzzer output pin

* Indicates the status when the pin is configured for SNDA2

If the port is shared with the SNDA2 pin and other functions, the SNDA2 output function must be assigned to the port before activating the SNDA2. For more information, refer to the “I/O Ports” chapter.

15.2.2 Output Pin Drive Mode

The drive mode of the BZOUT and #BZOUT pins can be set to one of the two types shown below using the SNDSEL.SINV bit.

Direct drive mode (SNDSEL.SINV bit = 0)

This mode drives both the BZOUT and #BZOUT pins to low while the buzzer signal output is off to prevent the piezoelectric buzzer from applying unnecessary bias.

Normal drive mode (SNDSEL.SINV bit = 1)

In this mode, the #BZOUT pin always outputs the inverted signal of the BZOUT pin even when the buzzer output is off.

15.2.3 External Connections

Figure 15.2.3.1 and Figure 15.2.3.2 show connection diagrams between SNDA2 and a piezoelectric buzzer.

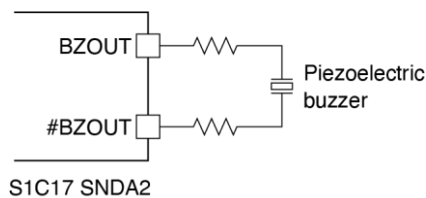


Figure 15.2.3.1 Connection between SNDA2 and Piezoelectric Buzzer (Direct Drive)

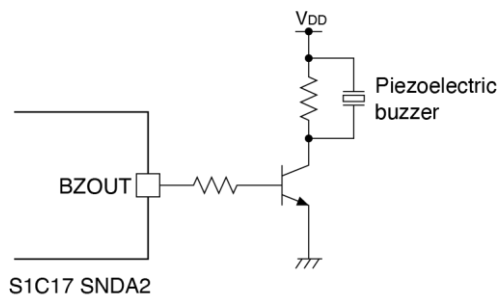


Figure 15.2.3.2 Connection between SNDA2 and Piezoelectric Buzzer (Single Pin Drive)

15.3 Clock Settings

15.3.1 SNDA2 Operating Clock

When using SNDA2, the SNDA2 operating clock CLK_SNDA2 must be supplied to SNDA2 from the clock generator. The CLK_SNDA2 supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
2. Set the following SNDCLK register bits:
 - SNDCLK.CLKSRC[1:0] bits (Clock source selection)
 - SNDCLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)

The CLK_SNDA2 frequency should be set to around 32,768 Hz.

15.3.2 Clock Supply in SLEEP Mode

When using SNDA2 during SLEEP mode, the SNDA2 operating clock CLK_SNDA2 must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_SNDA2 clock source.

If the CLGOSC.xxxxSLPC bit for the CLK_SNDA2 clock source is 1, the CLK_SNDA2 clock source is deactivated during SLEEP mode and SNDA2 stops with the register settings maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_SNDA2 is supplied and the SNDA2 operation resumes.

15.3.3 Clock Supply in DEBUG Mode

The CLK_SNDA2 supply during DEBUG mode should be controlled using the SNDCLK.DBRUN bit.

The CLK_SNDA2 supply to SNDA2 is suspended when the CPU enters DEBUG mode if the SNDCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_SNDA2 supply resumes. Although SNDA2 stops operating when the CLK_SNDA2 supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the SNDCLK.DBRUN bit = 1, the CLK_SNDA2 supply is not suspended and SNDA2 will keep operating in DEBUG mode.

15.4 Operations

15.4.1 Initialization

SNDA2 should be initialized with the procedure shown below.

1. Assign the SNDA2 output function to the ports. (Refer to the “I/O Ports” chapter.)
2. Configure the SNDA2 operating clock.
3. Set the SNDCTL.MODEN bit to 1. (Enable SNDA2 operations)
4. Set the SNDSSEL.SINV bit. (Set output pin drive mode)
5. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the SNDINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the SNDINTE register to 1. (Enable interrupts)

15.4.2 Buzzer Output in Normal Buzzer Mode

Normal buzzer mode generates a buzzer signal with the software specified frequency and duty ratio, and outputs the generated signal to outside the IC. The buzzer output duration can also be controlled via software.

An output start/stop procedure and the SNDA2 operations are shown below.

Normal buzzer output start/stop procedure

1. Set the SNDSSEL.MOSEL[1:0] bits to 0x0. (Set normal buzzer mode)
2. Write data to the following sound buffer (SNDDAT register) bits. (Start buzzer output)
 - SNDDAT.SLEN[5:0] bits (Set buzzer output signal duty ratio)
 - SNDDAT.SFRQ[7:0] bits (Set buzzer output signal frequency)
3. Write 1 to the SNDCTL.SSTP bit after the output period has elapsed. (Stop buzzer output)

15 Sound Generator (SNDA2)

Normal buzzer output operations

When data is written to the sound buffer (SNDDAT register), SNDA2 clears the SNDINTF.EMIF bit (sound buffer empty interrupt flag) to 0 and starts buzzer output operations.

The data written to the sound buffer is loaded into the sound register in sync with the CLK_SNDA2 clock. At the same time, the SNDINTF.EMIF bit and SNDINTF.SBSY bit are both set to 1. The output pin outputs the buzzer signal with the frequency/duty ratio specified.

Writing 1 to the SNDCTL.SSTP bit stops buzzer output and sets the SNDINTF.EDIF bit (sound output completion interrupt flag) to 1. The SNDINTF.SBSY bit is cleared to 0.

Figure 15.4.2.1 shows a buzzer output timing chart in normal buzzer mode.

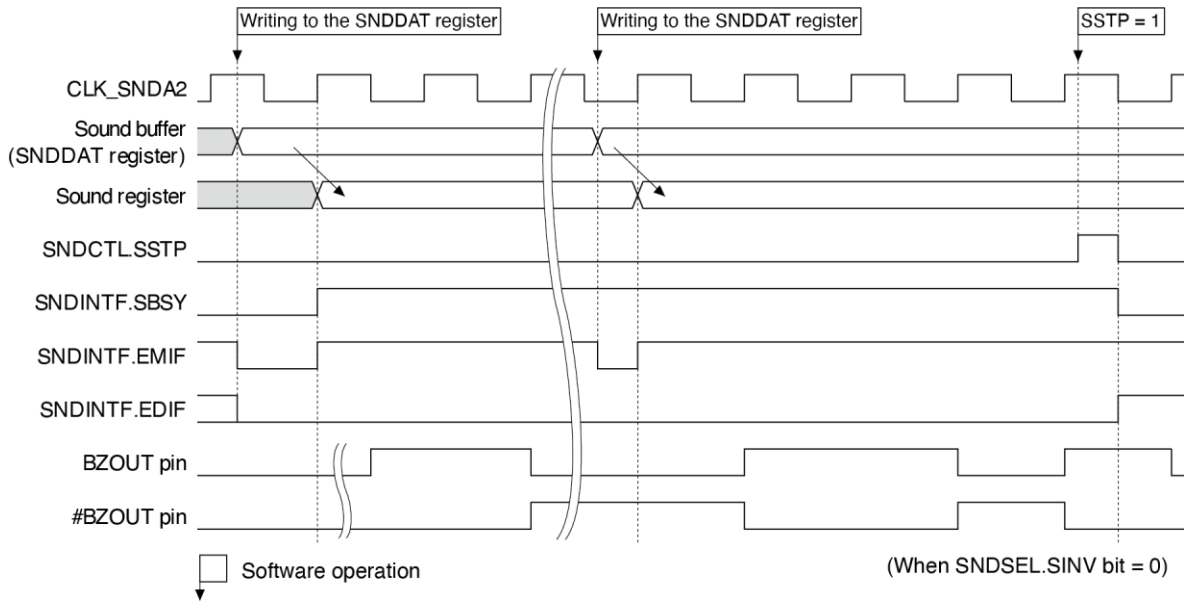


Figure 15.4.2.1 Buzzer Output Timing Chart in Normal Buzzer Mode

Buzzer output waveform configuration (normal buzzer mode/one-shot buzzer mode)

Set the buzzer signal frequency and duty ratio (high period/cycle) using the SNDDAT.SFRQ[7:0] and SNDDAT.SLEN[5:0] bits, respectively. Use the following equations to calculate these setting values.

$$\text{SNDDAT.SFRQ}[7:0] \text{ bits} = \frac{f_{\text{CLK_SNDA2}}}{f_{\text{BZOUT}}} - 1 \quad (\text{Eq. 15.1})$$

$$\text{SNDDAT.SLEN}[5:0] \text{ bits} = \left(\frac{f_{\text{CLK_SNDA2}}}{f_{\text{BZOUT}}} \times \frac{\text{DUTY}}{100} \right) - 1 \quad (\text{Eq. 15.2})$$

Where

- $f_{\text{CLK_SNDA2}}$: CLK_SNDA2 frequency [Hz]
- f_{BZOUT} : Buzzer signal frequency [Hz]
- DUTY: Buzzer signal duty ratio [%]

However, the following settings are prohibited:

- Settings as SNDDAT.SFRQ[7:0] bits \leq SNDDAT.SLEN[5:0] bits
- Settings as SNDDAT.SFRQ[7:0] bits = 0x00

Table 15.4.2.1 Buzzer Frequency Settings (when $f_{CLK_SNDA2} = 32$ kHz)

SNDDAT. SFRQ[7:0] bits	Frequency [Hz]	SNDDAT. SFRQ[7:0] bits	Frequency [Hz]	SNDDAT. SFRQ[7:0] bits	Frequency [Hz]	SNDDAT. SFRQ[7:0] bits	Frequency [Hz]
0x3f	500.0	0x2f	666.7	0x1f	1,000.0	0x0f	2,000.0
0x3e	507.9	0x2e	680.9	0x1e	1,032.3	0x0e	2,133.3
0x3d	516.1	0x2d	695.7	0x1d	1,066.7	0x0d	2,285.7
0x3c	524.6	0x2c	711.1	0x1c	1,103.4	0x0c	2,461.5
0x3b	533.3	0x2b	727.3	0x1b	1,142.9	0x0b	2,666.7
0x3a	542.4	0x2a	744.2	0x1a	1,185.2	0x0a	2,909.1
0x39	551.7	0x29	761.9	0x19	1,230.8	0x09	3,200.0
0x38	561.4	0x28	780.5	0x18	1,280.0	0x08	3,555.6
0x37	571.4	0x27	800.0	0x17	1,333.3	0x07	4,000.0
0x36	581.8	0x26	820.5	0x16	1,391.3	0x06	4,571.4
0x35	592.6	0x25	842.1	0x15	1,454.5	0x05	5,333.3
0x34	603.8	0x24	864.9	0x14	1,523.8	0x04	6,400.0
0x33	615.4	0x23	888.9	0x13	1,600.0	0x03	8,000.0
0x32	627.5	0x22	914.3	0x12	1,684.2	0x02	10,666.7
0x31	640.0	0x21	941.2	0x11	1,777.8	0x01	16,000.0
0x30	653.1	0x20	969.7	0x10	1,882.4	0x00	Cannot be set

Table 15.4.2.2 Buzzer Duty Ratio Setting Examples (when $f_{CLK_SNDA2} = 32$ kHz)

SNDDAT. SLEN[5:0] bits	Duty ratio by buzzer frequency					
	16,384 Hz	8,192 Hz	4,096 Hz	2,048 Hz	1,024 Hz	512 Hz
0x3f	-	-	-	-	-	-
0x3e	-	-	-	-	-	-
0x3d	-	-	-	-	-	99.2
0x3c	-	-	-	-	-	97.6
0x3b	-	-	-	-	-	96.0
0x3a	-	-	-	-	-	94.4
0x39	-	-	-	-	-	92.8
0x38	-	-	-	-	-	91.2
0x37	-	-	-	-	-	89.6
0x36	-	-	-	-	-	88.0
0x35	-	-	-	-	-	86.4
0x34	-	-	-	-	-	84.8
0x33	-	-	-	-	-	83.2
0x32	-	-	-	-	-	81.6
0x31	-	-	-	-	-	80.0
0x30	-	-	-	-	-	78.4
0x2f	-	-	-	-	-	76.8
0x2e	-	-	-	-	-	75.2
0x2d	-	-	-	-	-	73.6
0x2c	-	-	-	-	-	72.0
0x2b	-	-	-	-	-	70.4
0x2a	-	-	-	-	-	68.8
0x29	-	-	-	-	-	67.2
0x28	-	-	-	-	-	65.6
0x27	-	-	-	-	-	64.0
0x26	-	-	-	-	-	62.4
0x25	-	-	-	-	-	60.8
0x24	-	-	-	-	-	59.2
0x23	-	-	-	-	-	57.6
0x22	-	-	-	-	-	56.0
0x21	-	-	-	-	-	54.4
0x20	-	-	-	-	-	52.8
0x1f	-	-	-	-	-	51.2
0x1e	-	-	-	-	99.2	49.6
0x1d	-	-	-	-	96.0	48.0
0x1c	-	-	-	-	92.8	46.4
0x1b	-	-	-	-	89.6	44.8
0x1a	-	-	-	-	86.4	43.2
0x19	-	-	-	-	83.2	41.6
0x18	-	-	-	-	80.0	40.0
0x17	-	-	-	-	76.8	38.4

15 Sound Generator (SNDA2)

SNDDAT. SLEN[5:0] bits	Duty ratio by buzzer frequency					
	16,384 Hz	8,192 Hz	4,096 Hz	2,048 Hz	1,024 Hz	512 Hz
0x16	-	-	-	-	73.6	36.8
0x15	-	-	-	-	70.4	35.2
0x14	-	-	-	-	67.2	33.6
0x13	-	-	-	-	64.0	32.0
0x12	-	-	-	-	60.8	30.4
0x11	-	-	-	-	57.6	28.8
0x10	-	-	-	-	54.4	27.2
0x0f	-	-	-	-	51.2	25.6
0x0e	-	-	-	96.0	48.0	24.0
0x0d	-	-	-	89.6	44.8	22.4
0x0c	-	-	-	83.2	41.6	20.8
0x0b	-	-	-	76.8	38.4	19.2
0x0a	-	-	-	70.4	35.2	17.6
0x09	-	-	-	64.0	32.0	16.0
0x08	-	-	-	57.6	28.8	14.4
0x07	-	-	-	51.2	25.6	12.8
0x06	-	-	89.6	44.8	22.4	11.2
0x05	-	-	76.8	38.4	19.2	9.6
0x04	-	-	64.0	32.0	16.0	8.0
0x03	-	-	51.2	25.6	12.8	6.4
0x02	-	76.8	38.4	19.2	9.6	4.8
0x01	-	51.2	25.6	12.8	6.4	3.2
0x00	51.2	25.6	12.8	6.4	3.2	1.6

15.4.3 Buzzer Output in One-shot Buzzer Mode

One-shot buzzer mode is provided for clicking sound and short-duration buzzer output. This mode generates a buzzer signal with the software specified frequency and duty ratio, and outputs the generated signal for the short duration specified.

An output start procedure and the SNDA2 operations are shown below. For the buzzer output waveform, refer to “Buzzer Output in Normal Buzzer Mode.”

One-shot buzzer output start procedure

- Set the following SNDSEL register bits:
 - Set the SNDSEL.MOSEL[1:0] bits to 0x1. (Set one-shot buzzer mode)
 - SNDSEL.STIM[3:0] bits (Set output duration)
- Write data to the following sound buffer (SNDDAT register) bits. (Start buzzer output)
 - SNDDAT.SLEN[5:0] bits (Set buzzer output signal duty ratio)
 - SNDDAT.SFRQ[7:0] bits (Set buzzer output signal frequency)

One-shot buzzer output operations

When data is written to the sound buffer (SNDDAT register), SNDA2 clears the SNDINTF.EMIF bit (sound buffer empty interrupt flag) to 0 and starts buzzer output operations.

The data written to the sound buffer is loaded into the sound register in sync with the CLK_SNDA2 clock. At the same time, the SNDINTF.EMIF bit and SNDINTF.SBSY bit are both set to 1. The output pin outputs the buzzer signal with the frequency/duty ratio specified.

The buzzer output automatically stops when the duration specified by the SNDSEL.STIM[3:0] bits has elapsed. At the same time, the SNDINTF.EDIF bit (sound output completion interrupt flag) is set to 1 and the SNDINTF.SBSY bit is cleared to 0.

Figure 15.4.3.1 shows a buzzer output timing chart in one-shot buzzer mode.

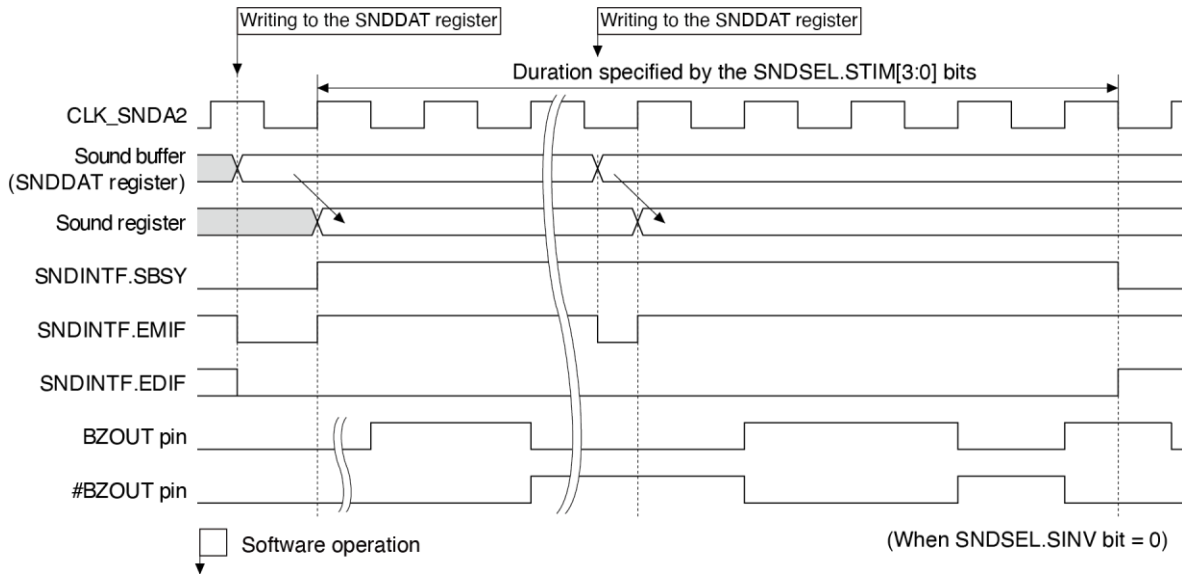


Figure 15.4.3.1 Buzzer Output Timing Chart in One-shot Buzzer Mode

15.4.4 Output in Melody Mode

Melody mode generates the buzzer signal with a melody according to the data written to the sound buffer (SNDDAT register) successively, and outputs the generated signal to outside the IC. An output start procedure and the SNDA2 operations are shown below.

Melody output start procedure

- Set the following SNDSSEL register bits:
 - Set the SNDSSEL.MOSEL[1:0] bits to 0x2. (Set melody mode)
 - SNDSSEL.STIM[3:0] bits (Set tempo)
- Write data to the following sound buffer (SNDDAT register) bits. (Start sound output)
 - SNDDAT.MDTI bit (Set tie)
 - SNDDAT.MDRS bit (Set note/rest)
 - SNDDAT.SLEN[5:0] bits (Set duration)
 - SNDDAT.SFRQ[7:0] bits (Set scale)
- Check to see if the SNDINTF.EMIF bit is set to 1 (an interrupt can be used).
- Repeat Steps 2 and 3 until the end of the melody.

Melody output operations

When data is written to the sound buffer (SNDDAT register), SNDA2 clears the SNDINTF.EMIF bit (sound buffer empty interrupt flag) to 0 and starts sound output operations.

The data written to the sound buffer is loaded into the sound register by the internal trigger signal. At the same time, the SNDINTF.EMIF bit and SNDINTF.SBSY bit are both set to 1. The output pin outputs the sound specified. The sound output stops if data is not written to the sound buffer (SNDDAT register) until the next trigger is issued. At the same time, the SNDINTF.EDIF bit (sound output completion interrupt flag) is set to 1 and the SNDINTF.SBSY bit is cleared to 0.

Figure 15.4.4.1 shows a melody mode operation timing chart.

15 Sound Generator (SNDA2)

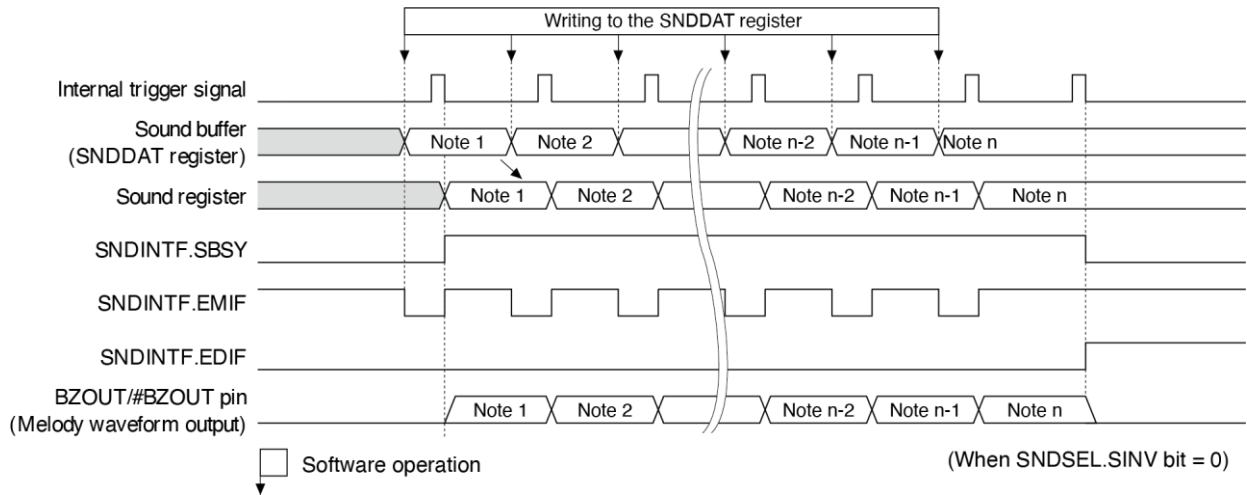


Figure 15.4.4.1 Melody Mode Operation Timing Chart

Melody output waveform configuration

Note/rest (duration) specification

Notes and rests can be specified using the SNDDAT.MDRS and SNDDAT.SLEN[5:0] bits.

Table 15.4.4.1 Note/Rest Specification

SNDDAT.SLEN[5:0] bits	SNDDAT.MDRS bit	
	0: Note	1: Rest
0x0f	Half note	Half rest
0x0b	Dotted quarter note	Dotted quarter rest
0x07	Quarter note	Quarter rest
0x05	Dotted eighth note	Dotted eighth rest
0x03	Eighth note	Eighth rest
0x01	Sixteenth note	Sixteenth rest
0x00	Thirty-second note	Thirty-second rest
Other	Setting not allowed	

Tie specification

A tie takes effect by setting the SNDDAT.MDTI bit to 1 and the previous note and the current note are played continuously. Slurs cannot be specified.

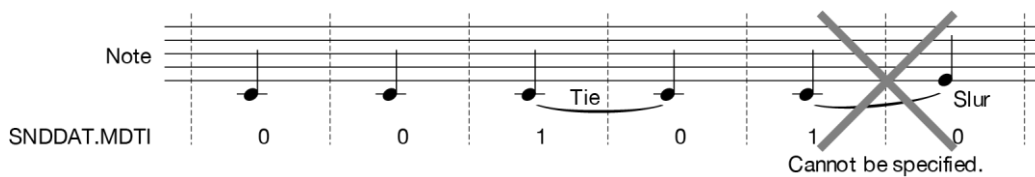


Figure 15.4.4.2 Tie

Scale specification

Scales can be specified using the SNDDAT.SFRQ[7:0] bits.

Table 15.4.4.2 Scale Specification (when $f_{CLK_SNDA2} = 32k$ Hz)

SNDDAT.SFRQ[7:0] bits	Scale	Frequency [Hz]
0xf5	C3	131.15
0xe7	C#3/Db3	139.13
0xda	D3	147.47
0xce	D#3/Eb3	156.10
0xc2	E3	165.80
0xb7	F3	175.82
0xad	F#3/Gb3	186.05
0xa3	G3	196.32
0x9a	G#3/Ab3	209.15
0x91	A3	220.69

SNDDAT.SFRQ[7:0] bits	Scale	Frequency [Hz]
0x89	A#3/Bb3	131.15
0x81	B3	139.13
0x7a	C4	147.47
0x73	C#4/Db4	156.10
0x6c	D4	165.80
0x66	D#4/Eb4	175.82
0x61	E4	186.05
0x5b	F4	196.32
0x56	F#4/Gb4	209.15
0x51	G4	220.69
0x4c	G#4/Ab4	131.15
0x48	A4	139.13
0x44	A#4/Bb4	147.47
0x40	B4	156.10
0x3c	C5	165.80
0x39	C#5/Db5	175.82
0x36	D5	186.05
0x33	D#5/Eb5	196.32
0x30	E5	209.15
0x2d	F5	220.69
0x2a	F#5/Gb5	131.15
0x28	G5	139.13
0x26	G#5/Ab5	147.47
0x24	A5	156.10
0x21	A#5/Bb5	165.80
0x20	B5	175.82
0x1e	C6	186.05

15.5 Interrupts

SNDA2 has a function to generate the interrupts shown in Table 15.5.1.

Table 15.5.1 SNDA2 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Sound buffer empty	SNDINTF.EMIF	When data in the sound buffer (SNDDAT register) is transferred to the sound register or 1 is written to the SNDCTL.SSTP bit	Writing to the SNDDAT register
Sound output completion	SNDINTF.EDIF	When a sound output has completed	Writing 1 or writing to the SNDDAT register

SNDA2 provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

15.6 Control Registers

SNDA2 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDCLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7	–	0	–	R	
	6–4	CLKDIV[2:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the SNDA2 operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

15 Sound Generator (SNDA2)

Bit 7 **Reserved**

Bits 6–4 **CLKDIV[2:0]**

These bits select the division ratio of the SNDA2 operating clock.

Bits 3–2 **Reserved**

Bits 1–0 **CLKSRC[1:0]**

These bits select the clock source of SNDA2.

Table 15.6.1 Clock Source and Division Ratio Settings

SNDCLK. CLKDIV[2:0] bits	SNDCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x7	Reserved	1/1	Reserved	1/1
0x6				
0x5	1/128		1/128	
0x4	1/64		1/64	
0x3	1/32		1/32	
0x2	1/16		1/16	
0x1	1/8		1/8	
0x0	1/4		1/4	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The SNDCLK register settings can be altered only when the SNDCTL.MODEN bit = 0.

SNDA2 Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDSEL	15–12	–	0x0	–	R	–
	11–8	STIM[3:0]	0x0	H0	R/W	
	7–3	–	0x00	–	R	
	2	SINV	0	H0	R/W	
	1–0	MOSEL[1:0]	0x0	H0	R/W	

Bits 15–12 **Reserved**

Bits 11–8 **STIM[3:0]**

These bits select a tempo (when melody mode is selected) or a one-shot buzzer output duration (when one-shot buzzer mode is selected).

Table 15.6.2 Tempo/One-shot Buzzer Output Duration Selections (when f_{CLK_SNDA2} = 32 kHz)

SNDSEL. STIM[3:0] bits	Tempo (= Quarter note/minute)	One-shot buzzer output duration [ms]
0xf	30	253.6
0xe	32	237.8
0xd	34.3	221.9
0xc	36.9	206.1
0xb	40	190.2
0xa	43.6	174.4
0x9	48	158.5
0x8	53.3	142.7
0x7	60	126.8
0x6	68.6	111.0
0x5	80	95.1
0x4	96	79.3
0x3	120	63.4
0x2	160	47.6
0x1	240	31.7
0x0	480	15.9

Note: Be sure to avoid altering these bits when SNDINTF.SBSY bit = 1.

Bits 7–3 **Reserved**

Bit 2 SINV

This bit selects an output pin drive mode.

1 (R/W): Normal drive mode

0 (R/W): Direct drive mode

For more information, refer to “Output Pin Drive Mode.”

Bits 1–0 MOSEL[1:0]

These bits select a sound output mode.

Table 15.6.3 Sound Output Mode Selection

SNDSEL.MOSEL[1:0] bits	Sound output mode
0x3	Reserved
0x2	Melody mode
0x1	One-shot buzzer mode
0x0	Normal buzzer mode

SNDA2 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDCTL	15–9	–	0x00	–	R	–
	8	SSTP	0	H0	R/W	
	7–1	–	0x00	–	R	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved**Bit 8 SSTP**

This bit stops sound output.

1 (W): Stop sound output

0 (W): Ineffective

1 (R): In stop process

0 (R): Stop process completed/Idle

The SNDCTL.SSTP bit is used to stop buzzer output in normal buzzer mode. After 1 is written, this bit is cleared to 0 when the sound output has completed. Also in one-shot buzzer mode/melody mode, writing 1 to this bit can forcibly terminate the sound output.

Bits 7–1 Reserved**Bit 0 MODEN**

This bit enables the SNDA2 operations.

1 (R/W): Enable SNDA2 operations (The operating clock is supplied.)

0 (R/W): Disable SNDA2 operations (The operating clock is stopped.)

SNDA2 Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDDAT	15	MDTI	0	H0	R/W	–
	14	MDRS	0	H0	R/W	
	13–8	SLEN[5:0]	0x00	H0	R/W	
	7–0	SFRQ[7:0]	0xff	H0	R/W	

This register functions as a sound buffer. Writing data to this register starts sound output. For detailed information on the setting data, refer to “Buzzer output waveform configuration (normal buzzer mode/one-shot buzzer mode)” and “Melody output waveform configuration.”

Bit 15 MDTI

This bit specifies a tie (continuous play between the current note with this bit set to 1 and the subsequent note) in melody mode.

1 (R/W): Enable tie

0 (R/W): Disable tie

15 Sound Generator (SNDA2)

The successive notes must be the same scale. Therefore, a slur cannot be specified. If a continuous play is specified to the notes with a different scale, the setting of this bit is ignored.
This bit is also ignored in normal buzzer mode/one-shot buzzer mode.

Bit 14 MDRS

This bit selects the output type in melody mode from a note or a rest .

1 (R/W): Rest

0 (R/W): Note

When a rest is selected, the BZOUT pin goes low and the #BZOUT pin goes high during the output duration. This bit is ignored in normal buzzer mode/one-shot buzzer mode.

Bits 13–8 SLEN[5:0]

These bits select a duration (when melody mode is selected) or a buzzer signal duty ratio (when normal buzzer mode/one-shot buzzer mode is selected).

Bits 7–0 SFRQ[7:0]

These bits select a scale (when melody mode is selected) or a buzzer signal frequency (when normal buzzer mode/one-shot buzzer mode is selected).

- Notes:
- In normal buzzer mode/one-shot buzzer mode, only the low-order 6 bits (SNDDAT.SFRQ[5:0] bits) are effective within the SNDDAT.SFRQ[7:0] bits. Always set the SNDDAT.SFRQ[7:6] bits to 0x0.
 - The SNDDAT register allows 16-bit data writing only. Data writings in 8-bit size will be ignored.

SNDA2 Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDINTF	15–9	–	0x00	–	R	–
	8	SBSY	0	H0	R	
	7–2	–	0x00	–	R	
	1	EMIF	1	H0	R	Cleared by writing to the SNDDAT register.
	0	EDIF	0	H0	R/W	Cleared by writing 1 or writing to the SNDDAT register.

Bits 15–9 Reserved

Bit 8 SBSY

This bit indicates the sound output status. (See Figure 15.4.2.1, Figure 15.4.3.1, and Figure 15.4.4.1.)

1 (R): Outputting

0 (R): Idle

Bits 7–2 Reserved

Bit 1 EMIF

Bit 0 EDIF

These bits indicate the SNDA2 interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

SNDINTF.EMIF bit: Sound buffer empty interrupt

SNDINTF.EDIF bit: Sound output completion interrupt

SNDA2 Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDINTE	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	EMIE	0	H0	R/W	
	0	EDIE	0	H0	R/W	

Bits 15–2 Reserved

Bit 1 **EMIE**
Bit 0 **EDIE**

These bits enable SNDA2 interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

SNDINTE.EMIE bit: Sound buffer empty interrupt

SNDINTE.EDIE bit: Sound output completion interrupt

16 LCD Driver (LCD4B)

16.1 Overview

LCD4B is an LCD driver to drive an LCD panel. The features of the LCD4B are listed below.

- The frame frequency is configurable into 16 steps.
- Provides all on, all off, and inverse display functions as well as normal display.
- The segment and common pin assignments can be inverted.
- Provides a partial common output drive function.
- Provides an n-segment-line inverse AC drive function.
- The LCD contrast is adjustable into 29 steps.
- Includes a power supply for 1/3 bias driving (allows external voltages to be applied, voltage-dividing resistors included).
- Provides a frame signal monitor output pin.
- Can generate interrupts every frame.

Figure 16.1.1 shows the LCD4B configuration.

Table 16.1.1 LCD4B Configuration of S1C17W11

Item	S1C17W11
Number of segments supported	Max. 80 segments (20SEG × 4COM)
SEG/COM outputs	20SEG × 1–4COM
Drive bias	1/3 bias
Embedded display data RAM	20 bytes
LCD power supply	Internal generation mode/External voltage application mode 1/2/3
LFRO output	Available

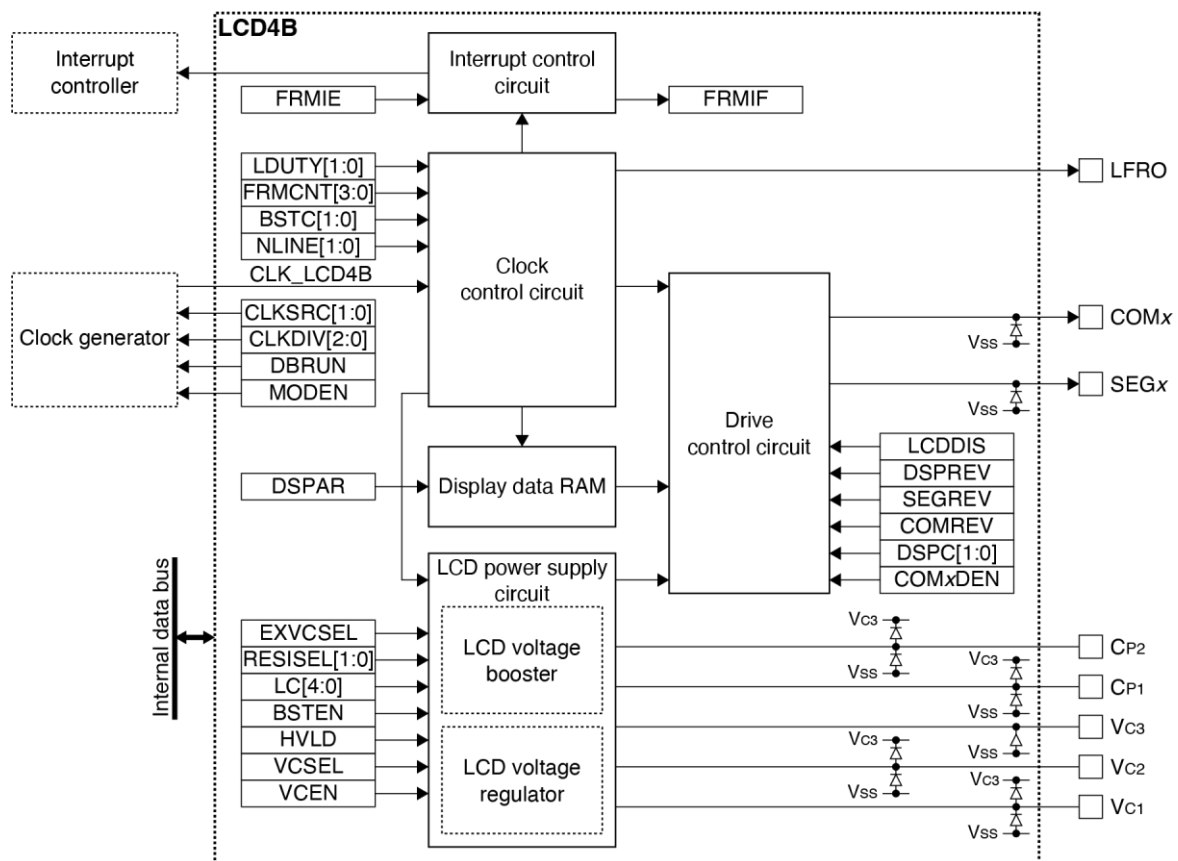


Figure 16.1.1 LCD4B Configuration

16.2 Output Pins and External Connections

16.2.1 List of Output Pins

Table 16.2.1.1 lists the LCD4B pins.

Table 16.2.1.1 List of LCD4B Pins

Pin name	I/O ^{*1}	Initial status ^{*1}	Function
COM0-3	A	Hi-Z / O (L) ^{*2}	Common data output-only pins
SEG0-1, 4-7	A	Hi-Z / O (L) ^{*2}	Segment data output-only pins
SEG2-3, 8-19	A	Hi-Z / O (L) ^{*2}	General-purpose IO/segment data output pins
LFRO	O	O (L)	Frame signal monitoring output pin
V _{C1}	P	-	LCD panel drive power supply pin
V _{C2}	P	-	LCD panel drive power supply pin
V _{C3}	P	-	LCD panel drive power supply pin
C _{P1}	A	-	LCD voltage booster capacitor connecting pin
C _{P2}	A	-	LCD voltage booster capacitor connecting pin

^{*1}: Indicates the status when the pin is configured for LCD4B. ^{*2}: When LCD4CTL.LCDDIS bit = 1

If the port is shared with the LCD4B pin and other functions, the LCD4B output function must be assigned to the port before activating the LCD4B. For more information, refer to the “I/O Ports” chapter.

Notes: • Be sure to avoid using the V_{C1} to V_{C3} pin outputs for driving external circuits.

- When an LCD panel is connected, the LCD4CTL.LCDDIS bit should be set to 1. If it is set to 0, the LCD panel characteristics may fluctuate.

16.2.2 External Connections

Figure 16.2.2.1 shows a connection diagram between LCD4B and an LCD panel.

Note: When the panel is connected, the LCD4CTL.LCDDIS bit must be set to 1 to bias the panel even if display is turned off.

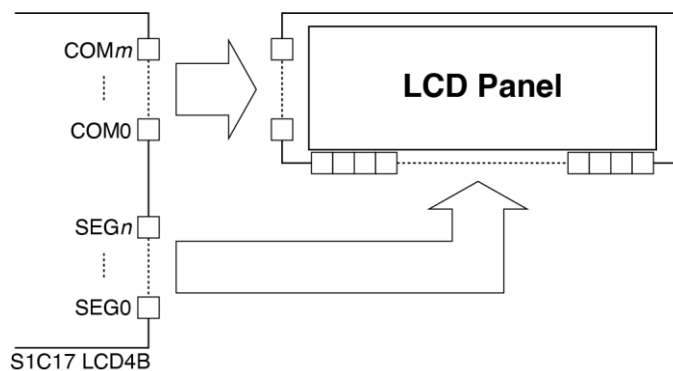


Figure 16.2.2.1 Connections between LCD4B and an LCD Panel

16.3 Clock Settings

16.3.1 LCD4B Operating Clock

When using LCD4B, the LCD4B operating clock CLK_LCD4B must be supplied to LCD4B from the clock generator. The CLK_LCD4B supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
2. Set the following LCD4CLK register bits:
 - LCD4CLK.CLKSRC[1:0] bits (Clock source selection)
 - LCD4CLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)

The CLK_LCD4B frequency should be set to around 32 kHz.

16.3.2 Clock Supply in SLEEP Mode

When using LCD4B during SLEEP mode, the LCD4B operating clock CLK_LCD4B must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_LCD4B clock source.

16.3.3 Clock Supply in DEBUG Mode

The CLK_LCD4B supply during DEBUG mode should be controlled using the LCD4CLK.DBRUN bit.

The CLK_LCD4B supply to LCD4B is suspended when the CPU enters DEBUG mode if the LCD4CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_LCD4B supply resumes. Although LCD4B stops operating and the display is turned off when the CLK_LCD4B supply is suspended, the registers retain the status before DEBUG mode was entered. If the LCD4CLK.DBRUN bit = 1, the CLK_LCD4B supply is not suspended and LCD4B will keep operating in DEBUG mode.

16.3.4 Frame Frequency

The LCD4B frame signal is generated by dividing CLK_LCD4B. The frame frequency is determined by selecting a division ratio from 16 variations depending on the drive duty using the LCD4TIM1.FRMCNT[3:0] bits. Use the following equation to calculate the frame frequency.

$$f_{FR} = \frac{f_{CLK_LCD4B}}{16 \times (FRMCNT + 1) \times (LDUTY + 1)} \quad (\text{Eq. 16.1})$$

Where

f_{FR} : Frame frequency [Hz]

f_{CLK_LCD4B} : LCD4B operating clock frequency [Hz]

FRMCNT: LCD4TIM1.FRMCNT[3:0] setting value (0 to 15)

LDUTY: LCD4TIM1.LDUTY[1:0] setting value (0 to 3)

Table 16.3.4.1 lists frame frequency settings when $f_{CLK_LCD4B} = 32$ kHz as an example.

Table 16.3.4.1 Frame Frequency Settings (when $f_{CLK_LCD4B} = 32$ kHz)

LCD4TIM1. FRMCNT[3:0] bits	Frame frequency [Hz]			
	1/4 duty	1/3 duty	1/2 duty	Static
0xf	31.3	41.7	62.5	125.0
0xe	33.3	44.4	66.7	133.3
0xd	35.7	47.6	71.4	142.9
0xc	38.5	51.3	76.9	153.8
0xb	41.7	55.6	83.3	166.7
0xa	45.5	60.6	90.9	181.8
0x9	50.0	66.7	100.0	200.0
0x8	55.6	74.1	111.1	222.2
0x7	62.5	83.3	125.0	250.0
0x6	71.4	95.2	142.9	285.7
0x5	83.3	111.1	166.7	333.3
0x4	100.0	133.3	200.0	400.0
0x3	125.0	166.7	250.0	500.0
0x2	166.7	222.2	333.3	666.7
0x1	250.0	333.3	500.0	1000.0
0x0	500.0	666.7	1000.0	2000.0

16.4 LCD Power Supply

The LCD drive voltages V_{C1} to V_{C3} can be generated by the internal LCD power supply circuit (LCD voltage regulator and LCD voltage booster). One or all voltages can also be applied from outside the IC.

16.4.1 Internal Generation Mode

This mode generates all the LCD drive voltages V_{C1} to V_{C3} on the chip. To put LCD4B into internal generation mode, set both the LCD4PWR.VCEN and LCD4PWR.BSTEN bits to 1 to turn both the LCD voltage regulator and LCD voltage booster on after setting the LCD4PWR.EXVCSEL bit to 0. The LCD4PWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 16.4.1.1 shows an external connection example for internal generation mode.

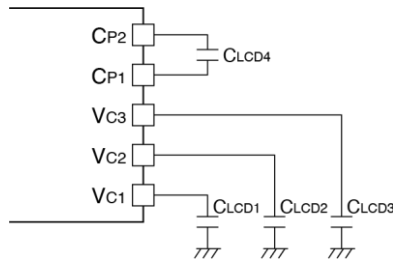


Figure 16.4.1.1 External Connection Example for Internal Generation Mode

16.4.2 External Voltage Application Mode 1

In this mode, all the LCD drive voltages V_{C1} to V_{C3} are applied from outside the IC. To put LCD4B into external voltage application mode 1, set both the LCD4PWR.VCEN and LCD4PWR.BSTEN bits to 0 to turn both the LCD voltage regulator and LCD voltage booster off after setting the LCD4PWR.EXVCSEL bit to 1. The LCD4PWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 16.4.2.1 shows an external connection example for external voltage application mode 1.

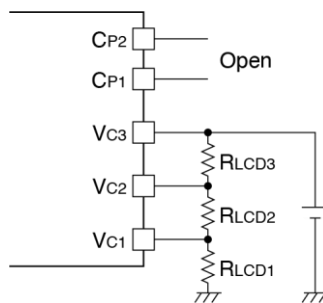


Figure 16.4.2.1 External Connection Example for External Voltage Application Mode 1 (resistor divider)

16.4.3 External Voltage Application Mode 2

In this mode, one of the LCD drive voltages V_{C1} to V_{C2} are applied from outside the IC and other voltages are internally generated. To put LCD4B into external voltage application mode 2, set the LCD4PWR.VCEN bit to 0 to turn the LCD voltage regulator off and set the LCD4PWR.BSTEN bit to 1 to turn the LCD voltage booster on after setting the LCD4PWR.EXVCSEL bit to 1. The LCD4PWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 16.4.3.1 shows an external connection example for external voltage application mode 2.

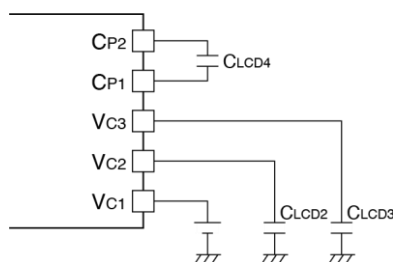


Figure 16.4.3.1 External Connection Example for External Voltage Application Mode 2 (when V_{C1} is applied)

16.4.4 External Voltage Application Mode 3

In this mode, the LCD drive voltage V_{C3} is applied from outside the IC and the V_{C1} and V_{C2} voltages are generated using the internal LCD voltage dividing resistors. To put LCD4B into external voltage application mode 3, set both the LCD4PWR.VCEN and LCD4PWR.BSTEN bits to 0 to turn both the LCD voltage regulator and LCD voltage booster off after setting the LCD4PWR.EXVCSEL bit to 1. Also set the LCD4PWR.RESISEL[1:0] bits to 0x1, 0x2, or 0x3 to use the internal LCD voltage dividing resistors according to the LCD panel load. A capacitor should be connected to the V_{C1} to V_{C3} pins while taking fluctuation of LCD load into consideration. Figure 16.4.4.1 shows an external connection example for external voltage application mode 3.

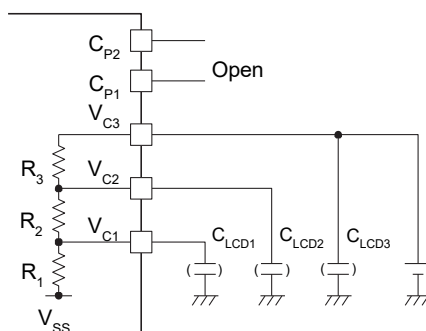


Figure 16.4.4.1 External Connection Example for External Voltage Application Mode 3

16.4.5 LCD Power Supply Circuit Settings

When using internal generation mode

Select the reference voltage for boosting voltage generated by the LCD voltage regulator according to the power supply voltage V_{DD} . Refer to “LCD Driver (LCD4B) Characteristics” in the “Electrical Characteristics” chapter and set the LCD4PWR.VCSEL bit. Current consumption can be reduced by selecting reference voltage V_{C2} as compared with reference voltage V_{C1} . By setting the LCD4PWR.HVLD bit to 1, the LCD voltage regulator enters heavy load protection mode and ensures stable V_{C1} to V_{C3} outputs. Heavy load protection mode should be set when the display has inconsistencies in density. Current consumption increases in heavy load protection mode, therefore do not set heavy load protection mode if unnecessary.

When using internal generation mode or external voltage application mode 2

Set the booster clock frequency used in the LCD voltage booster using the LCD4TIM2.BSTC[1:0] bits. Set it to the frequency that provides the best V_{C1} – V_{C3} output stability after being evaluated using the actual circuit board.

When using external voltage application mode 3

LCD4B includes voltage dividing resistors to generate the LCD drive voltages V_{C1} and V_{C2} from the V_{C3} that is applied externally. The resistance values can be adjusted according to the external LCD panel load by setting the LCD4PWR.REGISEL[1:0] bits.

LCD contrast adjustment

The LCD panel contrast can only be adjusted in internal generation mode using the LCD4PWR.LC[4:0] bits. For the adjustment range, refer to “LCD Driver (LCD4B) Characteristics” in the “Electrical Characteristics” chapter.

16.5 Operations

16.5.1 Initialization

The LCD4B should be initialized with the procedure shown below.

1. Assign the LCD4B output function to the ports. (Refer to the “I/O Ports” chapter.)
2. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).

16 LCD Driver (LCD4B)

3. Set the following LCD4CLK register bits:
 - LCD4CLK.CLKSRC[1:0] bits (Clock source selection)
 - LCD4CLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)
4. Configure the following LCD4CTL register bits:
 - Write 1 to the LCD4CTL.MODEN bit. (Enable LCD4B operating clock)
 - Write 1 to the LCD4CTL.LCDDIS bit. (Enable LCD driver pin discharge at display off)
5. Configure the following LCD4TIM1 register bits:
 - LCD4TIM1.LDUTY[1:0] bits (Set drive duty)
 - LCD4TIM1.FRMCNT[3:0] bits (Set frame frequency)
6. Configure the following LCD4TIM2 register bits:
 - LCD4TIM2.NLINE[1:0] bits (Set n-line inverse AC drive)
 - LCD4TIM2.BSTC[1:0] bits (Set booster clock frequency)
7. Set the LCD4PWR.EXVCSEL bit. (Select external voltage application mode/internal generation mode)
8. Configure the following LCD4PWR register bits:
 - LCD4PWR.RESISEL[1:0] bits (Select internal voltage dividing resistors)
 - LCD4PWR.VCEN bit (Enable LCD voltage regulator)
 - LCD4PWR.VCSEL bit (Set reference voltage for boosting)
 - LCD4PWR.BSTEN bit (Enable LCD voltage booster)
 - LCD4PWR.LC[4:0] bits (Set LCD contrast initial value)
9. Configure the following LCD4DSP register bits:
 - LCD4DSP.DSPAR bit (Select display area)
 - LCD4DSP.COMREV bit (Select COM pin assignment direction)
 - LCD4DSP.SEGREV bit (Select SEG pin assignment direction)
10. Write display data to the display data RAM.
11. Set the following bits when using the interrupt:
 - Write 1 to the LCD4INTF.FRMIIF bit. (Clear interrupt flag)
 - Set the LCD4INTE.FRMIIE bit to 1. (Enable LCD4B interrupt)

16.5.2 Display On/Off

The LCD display state is controlled using the LCD4DSP.DSPC[1:0] bits.

Table 16.5.2.1 LCD Display Control

LCD4DSP.DSPC[1:0] bits	LCD display
0x3	All off (static drive)
0x2	All on
0x1	Normal display
0x0	Display off

Selecting “Display off” stops the drive voltage supply and the LCD driver pin outputs are all set to V_{SS} level when the LCD4CTL.LCDDIS bit = 1.

Since “All on” and “All off” directly control the driving waveform output by the LCD driver, data in the display data RAM is not altered. The common pins are set to dynamic drive for “All on” and to static drive for “All off.” This function can be used to make the display flash on and off without altering the display memory.

Note: When “Display off” is selected while the external LCD drive voltages are being supplied in an external voltage application mode, the electric charges of V_{C3} must be discharged in the following procedure.

1. Turn the external power supply off.
2. Set the LCD4PWR.EXVCSEL bit to 0. (Select internal generation mode)
3. Set the LCD4PWR.EXVCSEL bit to 1. (Select external voltage application mode)

16.5.3 Inverted Display

The LCD panel display can be inverted (black/white inversion) using merely control bit manipulation, without rewriting the display data RAM. Setting the LCD4DSP.DSPREV bit to 0 inverts the display; setting it to 1 returns the display to normal status. Note that the display will not be inverted when the LCD4DSP.DSPC[1:0] bits = 0x3 (All off).

16.5.4 Drive Duty Switching

Drive duty can be set to 1/4 to 1/2 or static drive using the LCD4TIM1.LDUTY[1:0] bits. Table 16.5.4.1 shows the correspondence between the LCD4TIM1.LDUTY[1:0] bit settings, drive duty, and maximum number of display segments.

Table 16.5.4.1 Drive Duty Settings

LCD4TIM1.LDUTY[1:0] bits	Duty	Valid COM pins	Valid SEG pins	Max. number of display dots/segments
0x3	1/4	COM0-3	SEG0-19	80
0x2	1/3	COM0-2		60
0x1	1/2	COM0-1		40
0x0	Static	COM0		20

Unused common pins output an OFF waveform that turns the segments off.

16.5.5 Drive Waveforms

Figure 16.5.5.1 to Figure 16.5.5.4 show drive waveform examples by drive duty.

16 LCD Driver (LCD4B)

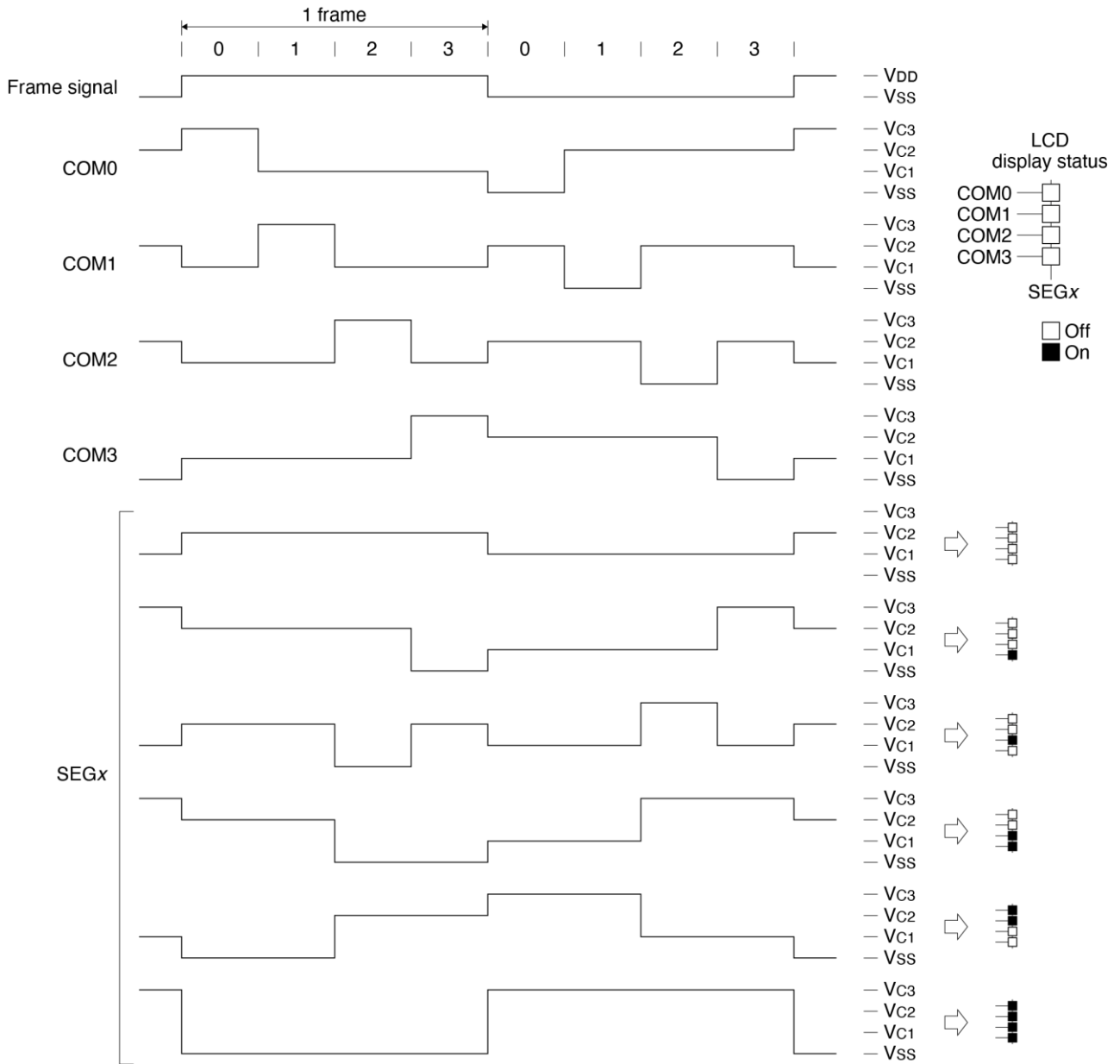


Figure 16.5.5.1 1/4 Duty Drive Waveform

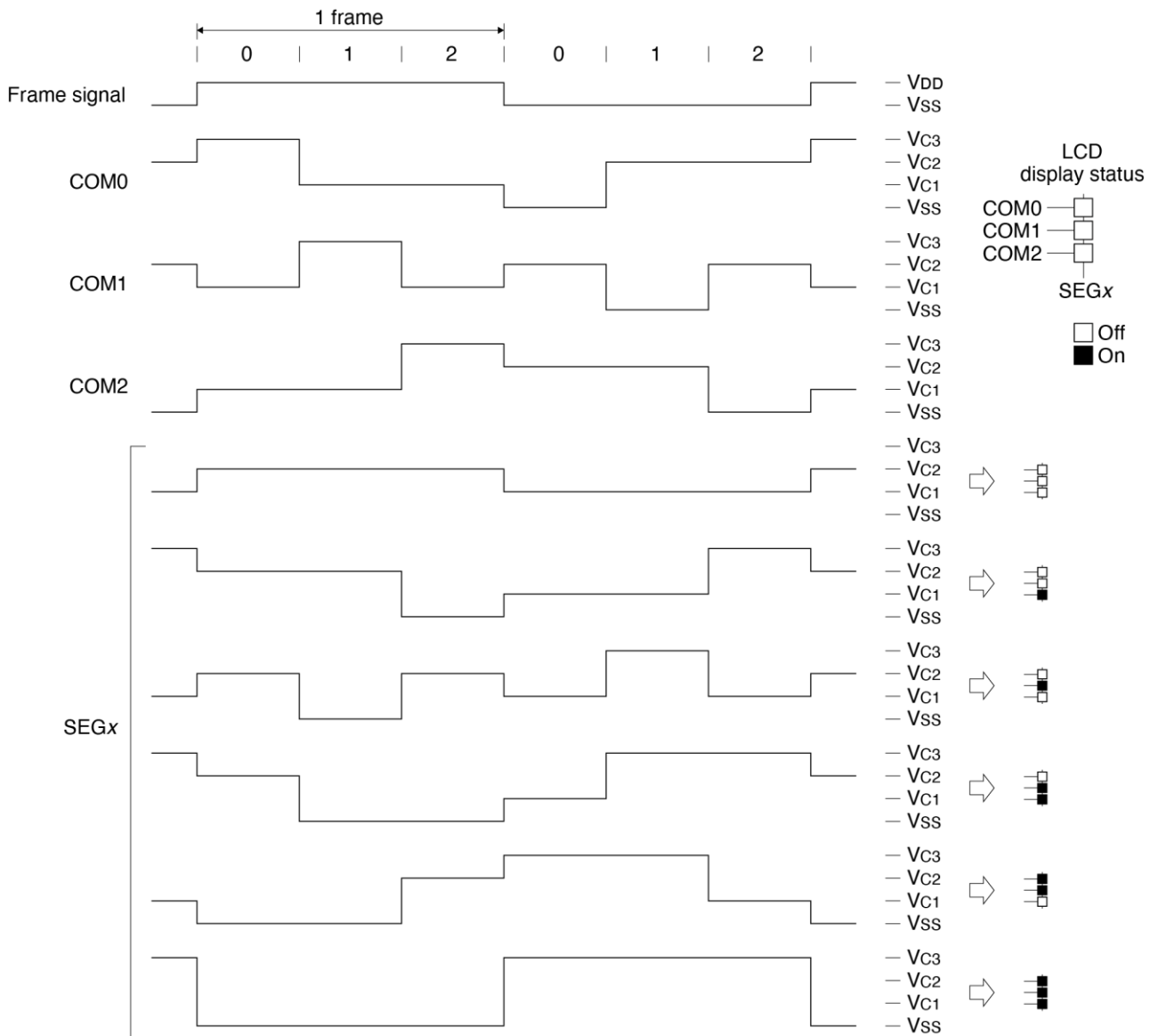


Figure 16.5.5.2 1/3 Duty Drive Waveform

16 LCD Driver (LCD4B)

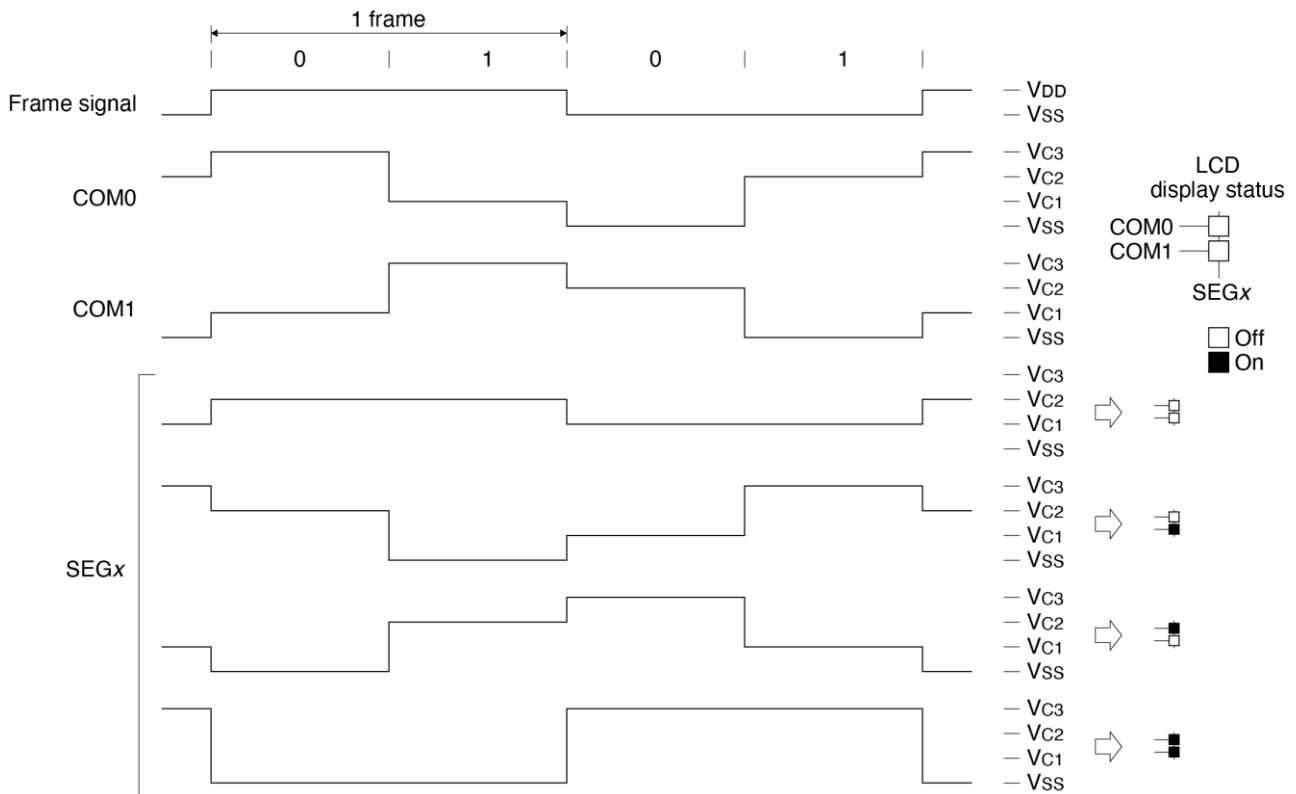


Figure 16.5.5.3 1/2 Duty Drive Waveform

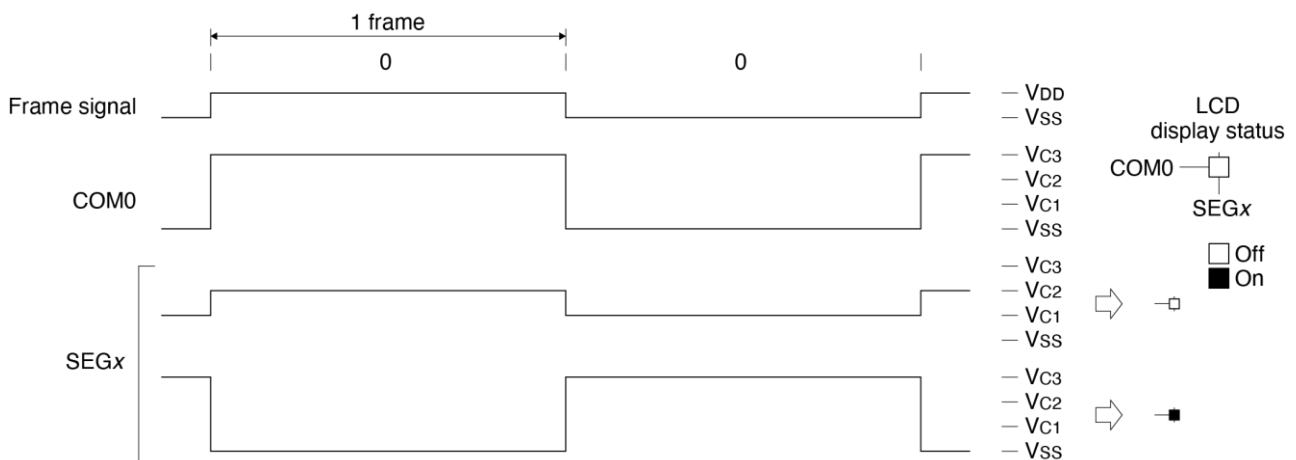


Figure 16.5.5.4 Static Drive Waveform

16.5.6 Partial Common Output Drive

By setting the LCD4COMC*.COMxDEN bit ($x = \text{COM No.}$) to 0, any common outputs can be set to off waveform regardless of the display data RAM contents. The partial common output drive function limits the display to the required area only to reduce power consumption.

16.5.7 n-Segment-Line Inverse AC Drive

The n-line inverse AC drive function may improve the display quality when being reduced such as when crosstalk occurs. To activate the n-line inverse AC drive function, select the number of lines to be inverted using the LCD4TIM2.NLINE[1:0] bits. The setting value should be determined after being evaluated using the actual circuit board. Note that using the n-line inverse AC drive function increases current consumption.

Table 16.5.7.1 Selecting Number of Inverse Lines

LCD4TIM2.NLINE[1:0] bits	Number of inverse lines
0x3	3 lines
0x2	2 lines
0x1	1 line
0x0	Normal drive

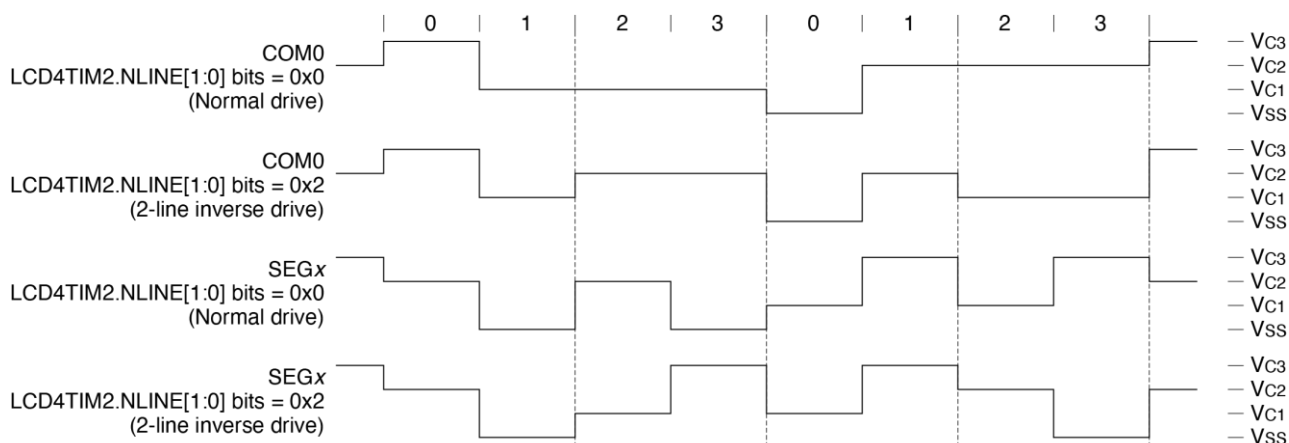


Figure 16.5.7.1 1/4 Duty Normal Drive Waveform and 2-line Inverse Drive Waveform

16.6 Display Data RAM

The display data RAM is located beginning with address 0x7000.

The correspondence between the memory bits of the display data RAM and the common/segment pins varies depending on the selected conditions below.

- Drive duty (1/4 to 1/2 or static drive)
- Segment pin assignment (normal or inverse)
- Common pin assignment (normal or inverse)

Figure 16.6.3.1 to Figure 16.6.3.4 show the correspondence between display data RAM and the common/segment pins by drive duty.

Writing 1 to the display data RAM bit corresponding to a segment on the LCD panel turns the segment on, while writing 0 turns the segment off. Since the display memory is a RAM allowing reading and writing, bits can be controlled individually using logic operation instructions (read-modify-write instructions).

The area unused for display can be used as general-purpose RAM.

16.6.1 Display Area Selection

In the display data RAM, two screen areas can be allocated and the LCD4DSP.DSPAR bit can be used to switch between the screens. Setting the LCD4DSP.DSPAR bit to 0 selects display area 0; setting to 1 selects display area 1.

16.6.2 Segment Pin Assignment

The display data RAM address assignment for the segment pins can be inverted using the LCD4DSP.SEGREV bit.

When the LCD4DSP.SEGREV bit is set to 1, memory addresses are assigned to segment pins in ascending order.

When the LCD4DSP.SEGREV bit is set to 0, memory addresses are assigned to segment pins in descending order.

16.6.3 Common Pin Assignment

The display data RAM bit assignment for the common pins can be inverted using the LCD4DSP.COMREV bit. When the LCD4DSP.COMREV bit is set to 1, memory bits are assigned to common pins in ascending order. When the LCD4DSP.COMREV bit is set to 0, memory bits are assigned to common pins in descending order.

16 LCD Driver (LCD4B)

Unused area (general-purpose RAM)
 Unimplemented area

1/4 duty

Bit	Address				LCD4DSP. COMREV bit = 1	LCD4DSP. COMREV bit = 0			
D0	0x7000	0x7001	0x7002	0x7003	Display area 0	0x7012	0x7013	COM0	COM3
D1								COM1	COM2
D2								COM2	COM1
D3								COM3	COM0
D4									
D5									
D6									
D7									
D0	0x7020	0x7021	0x7022	0x7023	Display area 1	0x7032	0x7033	COM0	COM3
D1								COM1	COM2
D2								COM2	COM1
D3								COM3	COM0
D4									
D5									
D6									
D7									
LCD4DSP.SEGREV bit = 1	SEG0	SEG1	SEG2	SEG3	...	SEG18	SEG19		
LCD4DSP.SEGREV bit = 0	SEG19	SEG18	SEG17	SEG16	...	SEG1	SEG0		

Figure 16.6.3.1 Display Data RAM Map (1/4 duty)

1/3 duty

Bit	Address				LCD4DSP. COMREV bit = 1	LCD4DSP. COMREV bit = 0			
D0	0x7000	0x7001	0x7002	0x7003	Display area 0	0x7012	0x7013	COM0	COM2
D1								COM1	COM1
D2								COM2	COM0
D3									
D4									
D5									
D6									
D7									
D0	0x7020	0x7021	0x7022	0x7023	Display area 1	0x7032	0x7033	COM0	COM2
D1								COM1	COM1
D2								COM2	COM0
D3									
D4									
D5									
D6									
D7									
LCD4DSP.SEGREV bit = 1	SEG0	SEG1	SEG2	SEG3	...	SEG18	SEG19		
LCD4DSP.SEGREV bit = 0	SEG19	SEG18	SEG17	SEG16	...	SEG1	SEG0		

Figure 16.6.3.2 Display Data RAM Map (1/3 duty)

1/2 duty

Bit	Address						LCD4DSP. COMREV bit = 1	LCD4DSP. COMREV bit = 0		
D0	Display area 0						COM0	COM1		
D1	Display area 0						COM1	COM0		
D2	0x7000	0x7001	0x7002	0x7003			X	X		
D3					0x7012	0x7013				
D4										
D5										
D6										
D7										
D0	Display area 0								COM0	COM1
D1	Display area 0								COM1	COM0
D2	0x7020	0x7021	0x7022	0x7023			X	X		
D3					0x7032	0x7033				
D4										
D5										
D6										
D7										
LCD4DSP.SEGREV bit = 1	SEG0	SEG1	SEG2	SEG3	...				SEG18	SEG19
LCD4DSP.SEGREV bit = 0	SEG19	SEG18	SEG17	SEG16	...				SEG1	SEG0

Figure 16.6.3.3 Display Data RAM Map (1/2 duty)

Static drive

Bit	Address						LCD4DSP. COMREV bit = 1	LCD4DSP. COMREV bit = 0
D0	Display area 0						COM0	COM0
D1	Display area 0						X	X
D2	0x7000	0x7001	0x7002	0x7003				
D3					0x7012	0x7013		
D4								
D5								
D6								
D7								
D0	Display area 1							
D1	Display area 1						X	X
D2	0x7020	0x7021	0x7022	0x7023				
D3					0x7032	0x7033		
D4								
D5								
D6								
D7								
LCD4DSP.SEGREV bit = 1	SEG0	SEG1	SEG2	SEG3	...			
LCD4DSP.SEGREV bit = 0	SEG19	SEG18	SEG17	SEG16	...		SEG1	SEG0

Figure 16.6.3.4 Display Data RAM Map (static drive)

16.7 Interrupt

The LCD4B has a function to generate the interrupt shown in Table 16.7.1.

Table 16.7.1 LCD4B Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Frame	LCD4INTF.FRMIIF	Frame switching	Writing 1

The LCD4B provides an interrupt enable bit corresponding to the interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

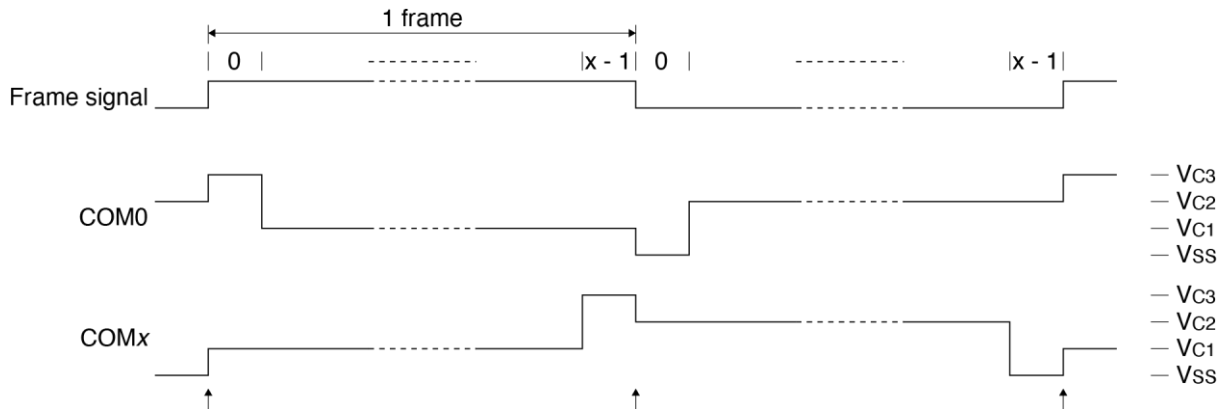


Figure 16.7.1 Frame Interrupt Timings (1/x duty)

16.8 Control Registers

LCD4B Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4CLK	15–9	–	0x00	–	R	–
	8	DBRUN	1	H0	R/W	
	7	–	0	–	R	
	6–4	CLKDIV[2:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 **DBRUN**

This bit sets whether the LCD4B operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bit 7 Reserved

Bits 6–4 **CLKDIV[2:0]**

These bits select the division ratio of the LCD4B operating clock.

Bits 3–2 Reserved

Bits 1–0 **CLKSRC[1:0]**

These bits select the clock source of the LCD4B.

Table 16.8.1 Clock Source and Division Ratio Settings

LCD4CLK. CLKDIV[2:0] bits	LCD4CLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x7	Reserved	1/1	Reserved	1/1
0x6				
0x5	1/128		1/128	
0x4	1/64		1/64	
0x3	1/32		1/32	
0x2	1/16		1/16	
0x1	1/8		1/8	
0x0	1/4		1/4	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The LCD4CLK register settings can be altered only when the LCD4CTL.MODEN bit = 0.

LCD4B Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4CTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	LCDDIS	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–2 Reserved

Bit 1 LCDDIS

This bit enables the SEG/COM-pin discharge operations when “Display off” is selected.

1 (R/W): Enable SEG/COM-pin discharge operations

0 (R/W): Disable SEG/COM-pin discharge operations

Setting this bit to 1 configures the SEG/COM pins to output a low level when “Display off” is selected.

Setting to 0 configures the SEG/COM pins to enter Hi-Z status when “Display off” is selected.

Bit 0 MODEN

This bit enables the LCD4B operations.

1 (R/W): Enable LCD4B operations

0 (R/W): Disable LCD4B operations

Setting this bit to 1 starts supplying the operating clock to LCD4B.

Note: If the LCD4CTL.MODEN bit is altered from 1 to 0 while the LCD panel is displaying, the LCD display is automatically turned off and the LCD4DSP.DSPC[1:0] bits are set to 0x0. Also the LCD voltage regulator is automatically turned off and the LCD4PWR.VCEN bit is set to 0.

LCD4B Timing Control Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4TIM1	15–12	–	0x0	–	R	–
	11–8	FRMCNT[3:0]	0x7	H0	R/W	
	7–2	–	0x00	–	R	
	1–0	LDUTY[1:0]	0x3	H0	R/W	

Bits 15–12 Reserved

Bits 11–8 FRMCNT[3:0]

These bits set the frame frequency. For more information, refer to “Frame Frequency.”

Bits 7–2 Reserved

Bits 1–0 LDUTY[1:0]

These bits set the drive duty. For more information, refer to “Drive Duty Switching.”

16 LCD Driver (LCD4B)

LCD4B Timing Control Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4TIM2	15–10	–	0x00	–	R	–
	9–8	BSTC[1:0]	0x1	H0	R/W	
	7–2	–	0x00	–	R	
	1–0	NLINE[1:0]	0x0	H0	R/W	

Bits 15–10 Reserved

Bits 9–8 BSTC[1:0]

These bits select the booster clock frequency for the LCD voltage booster.

Table 16.8.2 Booster Clock Frequency

LCD4TIM2.BSTC[1:0] bits	Booster clock frequency [Hz]
0x3	$f_{CLK_LCD4B}/64$
0x2	$f_{CLK_LCD4B}/32$
0x1	$f_{CLK_LCD4B}/16$
0x0	$f_{CLK_LCD4B}/4$

f_{CLK_LCD4B} : LCD4B operating clock frequency [Hz]

Bits 7–2 Reserved

Bits 1–0 NLINE[1:0]

These bits enable the n-line inverse AC drive function and set the number of inverse lines. For more information, refer to “n-Segment-Line Inverse AC Drive.”

LCD4B Power Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4PWR	15	EXVCSEL	1	H0	R/W	–
	14–13	RESISEL[1:0]	0x0	H0	R/W	
	12–8	LC[4:0]	0x00	H0	R/W	
	7–5	–	0x0	–	R	
	4	BSTEN	0	H0	R/W	
	3	–	0	–	R	
	2	HVLD	0	H0	R/W	
	1	VCSEL	0	H0	R/W	
	0	VCEN	0	H0	R/W	

Bit 15 EXVCSEL

This bit selects the LCD drive power supply mode (external voltage application mode or internal generation mode).

1 (R/W): External voltage application mode

0 (R/W): Internal generation mode

Note: Be sure to avoid applying voltages to the V_{C1} to V_{C3} pins when the LCD4PWR.EXVCSEL bit is set to 0, as the LCD power supply pins are short-circuited to GND.

Bits 14–13 RESISEL[1:0]

These bits select the internal LCD voltage dividing resistor value.

Table 16.8.3 Internal LCD Voltage Divider Resistor Value Adjustment

LCD4PWR.RESISEL[1:0] bits	Internal resistor value
0x3	Large
0x2	↓
0x1	Small
0x0	Internal voltage dividing resistors are not used.

Bits 12–8 LC[4:0]

These bits set the LCD panel contrast.

Table 16.8.4 LCD Contrast Adjustment

LCD4PWR.LC[4:0] bits	Contrast
0x1f	High (dark)
0x1e	↑
:	⋮
0x01	↓
0x00	Low (light)

Bits 7–5 **Reserved**

Bit 4 **BSTEN**

This bit turns the LCD voltage booster on and off.

1 (R/W): LCD voltage booster on

0 (R/W): LCD voltage booster off

For more information, refer to “LCD Power Supply.”

Bit 3 **Reserved**

Bit 2 **HVLD**

This bit sets the LCD voltage regulator into heavy load protection mode.

1 (R/W): Heavy load protection mode

0 (R/W): Normal mode

For more information, refer to “LCD Voltage Regulator Settings.”

Bit 1 **VCSEL**

This bit sets the LCD voltage regulator output (reference voltage for boosting).

1 (R/W): V_{C2}

0 (R/W): V_{C1}

For more information, refer to “LCD Voltage Regulator Settings.”

Note: The LCD4PWR.VCSEL bit must be set to 0 in an external voltage application mode.

Bit 0 **VCEN**

This bit turns the LCD voltage regulator on and off.

1 (R/W): LCD voltage regulator on

0 (R/W): LCD voltage regulator off

For more information, refer to “LCD Power Supply.”

Note: Before setting the LCD4PWR.VCEN bit to 1, set the LCD4PWR.EXVCSEL bit to 0. Setting the LCD4PWR.EXVCSEL bit to 1 automatically clears the LCD4PWR.VCEN bit to 0.

LCD4B Display Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4DSP	15–8	–	0x00	–	R	–
	7	–	0	–	R	–
	6	SEGREV	1	H0	R/W	–
	5	COMREV	1	H0	R/W	–
	4	DSPREV	1	H0	R/W	–
	3	–	0	–	R	–
	2	DSPAR	0	H0	R/W	–
	1–0	DSPC[1:0]	0x0	H0	R/W	–

Bits 15–7 **Reserved**

Bit 6 **SEGREV**

This bit selects the segment pin assignment direction.

1 (R/W): Normal assignment

0 (R/W): Inverse assignment

For more information, see Figure 16.6.3.1 to Figure 16.6.3.4.

16 LCD Driver (LCD4B)

Bit 5 COMREV

This bit selects the common pin assignment direction.

1 (R/W): Normal assignment

0 (R/W): Inverse assignment

For more information, see Figure 16.6.3.1 to Figure 16.6.3.4.

Bit 4 DSPREV

This bit controls black/white inversion on the LCD display.

1 (R/W): Normal display

0 (R/W): Inverted display

Bit 3 Reserved

Bit 2 DSPAR

This bit switches the display area in the display data RAM.

1 (R/W): Display area 1

0 (R/W): Display area 0

Bits 1–0 DSPC[1:0]

These bits control the LCD display on/off and select a display mode. For more information, refer to “Display On/Off.”

LCD4B COM Pin Control Register 0

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4COMC0	15–8	–	0x00	–	R	–
	7–4	–	0x0	–	R	
	3	COM3DEN	1	H0	R/W	
	2	COM2DEN	1	H0	R/W	
	1	COM1DEN	1	H0	R/W	
	0	COM0DEN	1	H0	R/W	

Bits 15–4 Reserved

Bits 3–0 COMxDEN

These bits configure the partial drive of the COMx pins.

1 (R/W): Normal output

0 (R/W): Off waveform output

LCD4B Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4INTF	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	FRMIF	0	H0	R/W	

Bits 15–1 Reserved

Bit 0 FRMIF

This bit indicates the frame interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

LCD4B Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD4INTE	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	FRMIE	0	H0	R/W	

Bits 15–1 Reserved

Bit 0

FRMIE

This bit enables the frame interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt

17 R/F Converter (RFC2)

17.1 Overview

The RFC2 is a CR oscillation type A/D converter (R/F converter).

The features of the RFC2 are listed below.

- Generates the RFC2 measurement power supply voltage (For details, refer to the “Power Supply, Reset, and Clocks” chapter.)
- Converts the sensor resistance into a digital value by performing CR oscillation and counting the oscillation clock.
- Achieves high-precision measurement system with low errors by oscillating the reference resistor and the sensor in the same conditions to obtain the difference between them.
- Includes a 24-bit measurement counter to count the oscillation clocks.
- Includes a 24-bit time base counter to count the internal clock for equalizing the measurement time between the reference resistor and the sensor.
- Supports DC bias resistive sensors and AC bias resistive sensors.
(A thermometer/hygrometer can be easily implemented by connecting a thermistor or a humidity sensor and a few passive elements (resistor and capacitor).)
- Allows measurement (counting) by inputting external clocks.
- Provides an output and continuous oscillation function for monitoring the oscillation frequency.
- Can generate reference oscillation completion, sensor (A and B) oscillation completion, measurement counter overflow error, and time base counter overflow error interrupts.

Figure 17.1.1 shows the RFC2 configuration.

Table 17.1.1 RFC2 Channel Configuration of S1C17W11

Item	S1C17W11
Number of channels	1 channel (Ch.0)

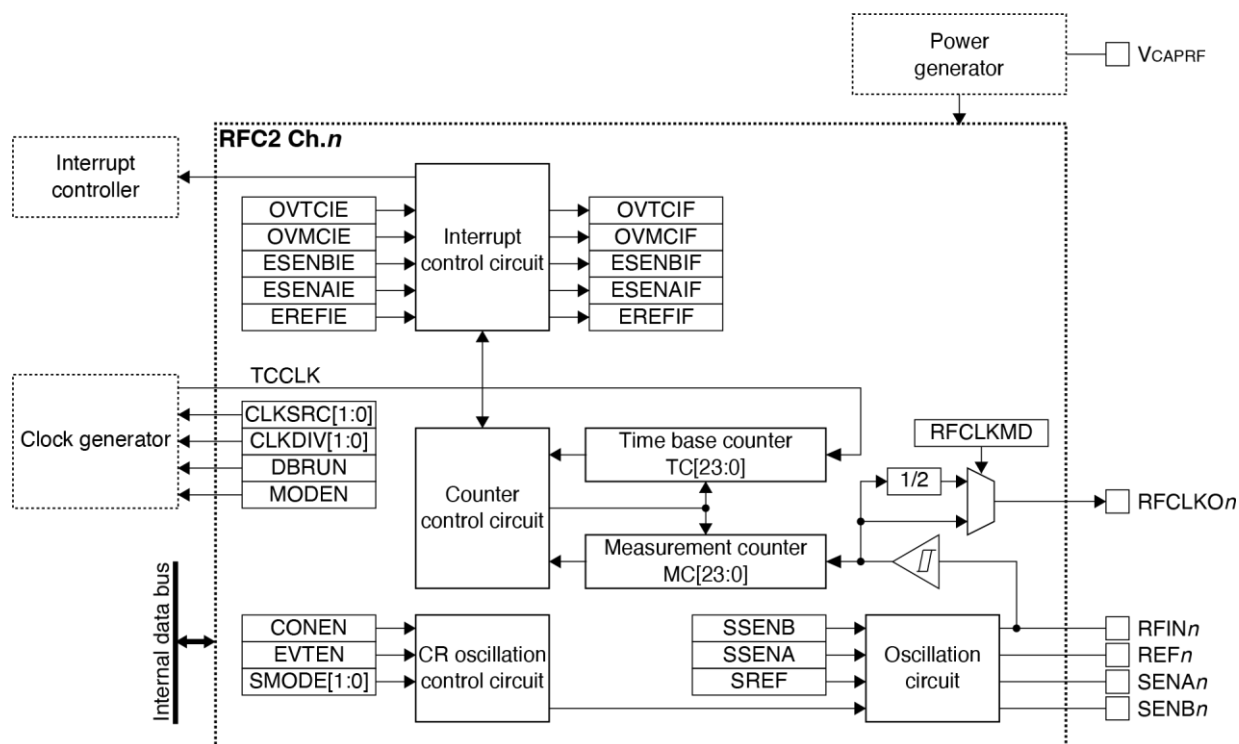


Figure 17.1.1 RFC2 Configuration

17.2 Input/Output Pins and External Connections

17.2.1 List of Input/Output Pins

Table 17.2.1.1 lists the RFC2 pins.

Table 17.2.1.1 List of RFC2 Pins

Pin name	I/O*	Initial status*	Function
V _{CAPRF}	A	Hi-Z	RFC2 measurement power supply voltage output pin
SEN _{Bn}	A	Hi-Z	Sensor B oscillation control pin
SEN _{An}	A	Hi-Z	Sensor A oscillation control pin
REF _n	A	Hi-Z	Reference oscillation control pin
RFIN _n	A	V _{SS}	RFCLK input or oscillation control pin
RFCLK _{On}	O	Hi-Z	RFCLK monitoring output pin RFCLK is output to monitor the oscillation frequency.

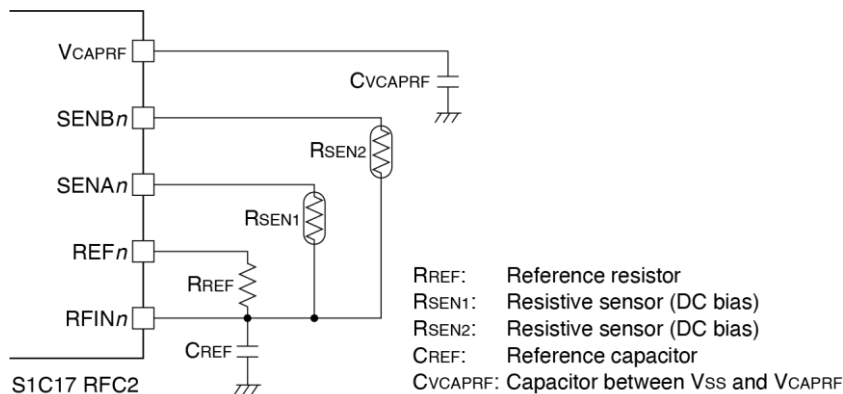
* Indicates the status when the pin is configured for the RFC2.

If the port is shared with the RFC2 pin and other functions, the RFC2 input/output function must be assigned to the port before activating the RFC2. For more information, refer to the “I/O Ports” chapter.

Note: The RFIN_n pin goes to V_{SS} level when the port is switched. Be aware that large current may flow if the pin is biased by an external circuit.

17.2.2 External Connections

The figures below show connection examples between the RFC2 and external sensors. For the oscillation mode and external clock input mode, refer to “Operating Mode.”



* Leave the unused pin (SEN_{An} or SEN_{Bn}) open if one resistive sensor only is used.

Figure 17.2.2.1 Connection Example in Resistive Sensor DC Oscillation Mode

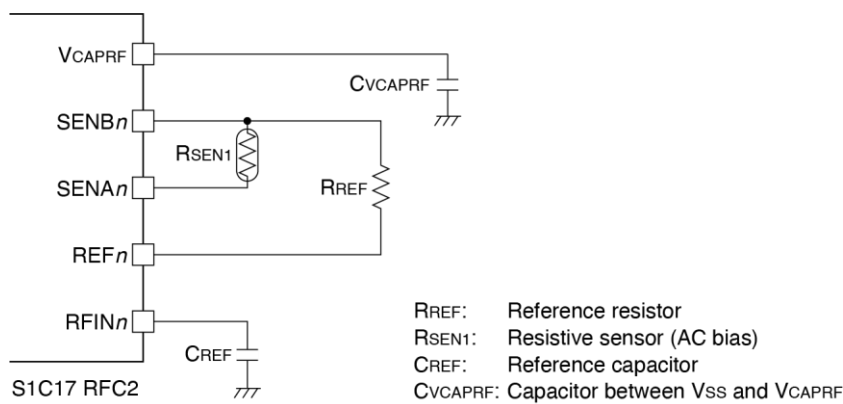


Figure 17.2.2.2 Connection Example in Resistive Sensor AC Oscillation Mode

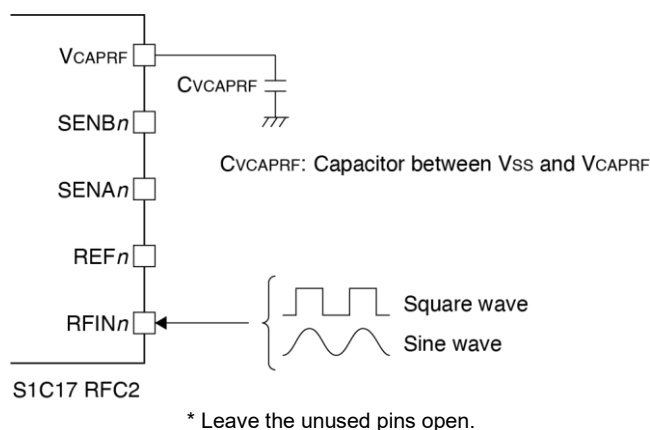


Figure 17.2.2.3 External Clock Input in External Clock Input Mode

17.3 Clock Settings

17.3.1 RFC2 Operating Clock

When using the RFC, the RFC2 operating clock TCCLK must be supplied to the RFC2 from the clock generator. The TCCLK supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).
2. Set the following RFC n CLK register bits:
 - RFC n CLK.CLKSRC[1:0] bits (Clock source selection)
 - RFC n CLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)

The time base counter performs counting with TCCLK set here. Selecting a higher clock results in higher conversion accuracy, note, however, that the frequency should be determined so that the time base counter will not overflow during reference oscillation. Also, set the frequency to be equal to or faster than the CR oscillation frequency using an external resistor and reference capacitor.

17.3.2 Clock Supply in SLEEP Mode

When using RFC2 during SLEEP mode, the RFC2 operating clock TCCLK must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the TCCLK clock source.

17.3.3 Clock Supply in DEBUG Mode

The TCCLK supply during DEBUG mode should be controlled using the RFC n CLK.DBRUN bit.

The TCCLK supply to the RFC2 is suspended when the CPU enters DEBUG mode if the RFC n CLK.DBRUN bit = 0. After the CPU returns to normal mode, the TCCLK supply resumes. Although the RFC2 stops operating when the TCCLK supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the RFC n CLK.DBRUN bit = 1, the TCCLK supply is not suspended and the RFC2 will keep operating in DEBUG mode.

17.4 Operations

17.4.1 Initialization

The RFC2 should be initialized with the procedure shown below.

1. Configure the PWGVD1CTL.REGSEL[1:0] bits. (Set RFC regulator)
(For details, refer to the “Power Supply, Reset, and Clock” chapter.)
2. Configure the RFC n CLK.CLKSRC[1:0] and RFC n CLK.CLKDIV[1:0] bits. (Configure operating clock)
3. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the RFC n INTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the RFC n INTE register to 1. (Enable interrupts)

17 R/F Converter (RFC2)

4. Assign the RFC2 input/output function to the ports. (Refer to the “I/O Ports” chapter.)
5. Configure the following RFC n CTL register bits:
 - RFC n CTL.EVTEN bit (Enable/disable external clock input mode)
 - RFC n CTL.SMODE[1:0] bits (Select oscillation mode)
 - Set the RFC n CTL.MODEN bit to 1. (Enable RFC2 operations)

Note: If the initial measurement results of the R/F converter are unstable, such as immediately after power-on, the measurement may become stable if you perform the discharge procedure for the V_{D1} and V_{CAPRF} terminals in the transition procedure to the low voltage side in the RFC regulator settings in step 1 before starting measurement.

17.4.2 Operating Modes

The RFC2 has two oscillation modes that use the RFC2 internal oscillation circuit and an external clock input mode for measurements using an external input clock. The channels may be configured to a different mode from others.

Oscillation mode

The oscillation mode is selected using the RFC n CTL.SMODE[1:0] bits.

DC oscillation mode for resistive sensor measurements

This mode performs measurements by DC driving the reference resistor and the resistive sensor to oscillate. Set the RFC2 into this mode when a DC bias resistive sensor is connected. This mode allows connection of two resistive sensors to a channel.

AC oscillation mode for resistive sensor measurements

This mode performs measurements by AC driving the reference resistor and the resistive sensor to oscillate. Set the RFC2 into this mode when an AC bias resistive sensor is connected. One resistive sensor only can be connected to a channel.

External clock input mode (event counter mode)

This mode enables input of external clock/pulses to perform counting similar to the internal oscillation clock. A sine wave may be input as well as a square wave (for the threshold value of the Schmitt input, refer to “R/F Converter Characteristics, High level Schmitt input threshold voltage V_{T+} and Low level Schmitt input threshold voltage V_{T-} ” in the “Electrical Characteristics” chapter). This function is enabled by setting the RFC n CTL.EVTEN bit to 1. The measurement procedure is the same as when the internal oscillation circuit is used.

17.4.3 RFC2 Counters

The RFC2 incorporates two counters shown below.

Measurement counter (MC)

The measurement counter is a 24-bit presettable up counter. Counting the reference oscillation clock and the sensor oscillation clock for the same duration of time using this counter minimizes errors caused by voltage, and unevenness of IC quality, as well as external parts and on-board parasitic elements. The counter values should be corrected via software after the reference and sensor oscillations are completed according to the sensor characteristics to determine the value being currently detected by the sensor.

Time base counter (TC)

The time base counter is a 24-bit presettable up/down counter. The time base counter counts up with TCCLK during reference oscillation to measure the reference oscillation time. During sensor oscillation, it counts down from the reference oscillation time and stops the sensor oscillation when it reaches 0x000000. This means that the sensor oscillation time becomes equal to the reference oscillation time. The value counted during reference oscillation should be saved in the memory. It can be reused at subsequent sensor oscillations omitting reference oscillations.

Counter initial value

To obtain the difference between the reference oscillation and sensor oscillation clock count values from the measurement counter simply, appropriate initial values must be set to the measurement counter before starting reference oscillation.

Connecting the reference element and sensor with the same resistance will result in $\langle \text{Initial value: } n \rangle = \langle \text{Counter value at the end of sensor oscillation: } m \rangle$ (if error = 0). Setting a large $\langle \text{Initial value: } n \rangle$ increases the resolution of measurement. However, the measurement counter may overflow during sensor oscillation when the sensor value decreases below the reference element value (the measurement will be canceled). The initial value for the measurement counter should be determined taking the range of sensor value into consideration. The time base counter should be set to 0x000000 before starting reference oscillation.

Counter value read

The measurement and time base counters operate on RFCCLK and TCCLK, respectively. Therefore, to read correctly by the CPU while the counter is running, read the counter value twice or more and check to see if the same value is read.

17.4.4 Converting Operations and Control Procedure

An R/F conversion procedure and the RFC2 operations are shown below. Although the following descriptions assume that the internal oscillation circuit is used, external clock input mode can be controlled with the same procedure.

R/F control procedure

1. Set the initial value (0x000000 - n) to the RFCnMCH and RFCnMCL registers (measurement counter).
2. Clear the RFCnTCH and RFCnTCL registers (time base counter) to 0x000000.
3. Clear both the RFCnINTF.EREFIF and RFCnINTF.OVTCIF bits by writing 1.
4. Set the RFCnTRG.SREF bit to 1 to start reference oscillation.
5. Wait for an RFC2 interrupt.
 - i. If the RFCnINTF.EREFIF bit = 1 (reference oscillation completion), clear the RFCnINTF.EREFIF bit and then go to Step 6.
 - ii. If the RFCnINTF.OVTCIF bit = 1 (time base counter overflow error), clear the RFCnINTF.OVTCIF bit and terminate measurement as an error or retry after altering the measurement counter initial value.
6. Clear the RFCnINTF.ESENAIF, RFCnINTF.ESENBIF, and RFCnINTF.OVMCIF bits by writing 1.
7. Set the RFCnTRG.SSENA bit (sensor A) or the RFCnTRG.SSENB bit (sensor B) corresponding to the sensor to be measured to 1 to start sensor oscillation (use the RFCnTRG.SSENA bit in AC oscillation mode).
8. Wait for an RFC2 interrupt.
 - i. If the RFCnINTF.ESENAIF bit = 1 (sensor A oscillation completion) or the RFCnINTF.ESENBIF bit = 1 (sensor B oscillation completion), clear the RFCnINTF.ESENAIF or RFCnINTF.ESENBIF bit and then go to Step 9.
 - ii. If the RFCnINTF.OVMCIF bit = 1 (measurement counter overflow error), clear the RFCnINTF.OVMCIF bit and terminate measurement as an error or retry after altering the measurement counter initial value.
9. Read the RFCnMCH and RFCnMCL registers (measurement counter) and correct the results depending on the sensor to obtain the detected value.

R/F converting operations

Reference oscillation

When the RFCnTRG.SREF bit is set to 1 in Step 4 of the conversion procedure above, the RFC2 Ch.n starts CR oscillation using the reference resistor. The measurement counter starts counting up using the CR oscillation clock from the initial value that has been set. The time base counter starts counting up using TCCLK from 0x000000.

When the measurement counter or the time base counter overflows (0xfffff → 0x000000), the RFCnTRG.SREF bit is cleared to 0 and the reference oscillation stops automatically.

The measurement counter overflow sets the RFCnINTF.EREFIF bit to 1 indicating that the reference oscillation has been terminated normally. If the RFCnINTE.EREFIE bit = 1, a reference oscillation completion interrupt request occurs at this point.

The time base counter overflow sets the RFCnINTF.OVTCIF bit to 1 indicating that the reference oscillation has been terminated abnormally. If the RFCnINTE.OVTCIE bit = 1, a time base counter overflow error interrupt request occurs at this point.

Sensor oscillation

When the RFCnTRG.SSENA bit (sensor A) or the RFCnTRG.SSENB bit (sensor B) is set to 1 in Step 7 of the conversion procedure above, the RFC2 Ch.n starts CR oscillation using the sensor. The measurement counter starts counting up using the CR oscillation clock from 0x000000. The time base counter starts counting down using TCCLK from the value at the end of reference oscillation.

When the time base counter reaches 0x000000 or the measurement counter overflows (0xfffff → 0x000000), the RFCnTRG.SSENA bit or the RFCnTRG.SSENB bit that started oscillation is cleared to 0 and the sensor oscillation stops automatically.

The time base counter reaching 0x000000 sets the RFCnINTF.ESENAIF bit (sensor A) or the RFCnINTF.ESENBIF bit (sensor B) to 1 indicating that the sensor oscillation has been terminated normally. If the RFCnINTE.ESENAIE bit = 1 or the RFCnINTE.ESENBIE bit = 1, a sensor A or sensor B oscillation completion interrupt request occurs at this point.

The measurement counter overflow sets the RFCnINTF.OVMCIF to 1 indicating that the sensor oscillation has been terminated abnormally. If the RFCnINTE.OVMCIE bit = 1, a measurement counter overflow error interrupt request occurs at this point.

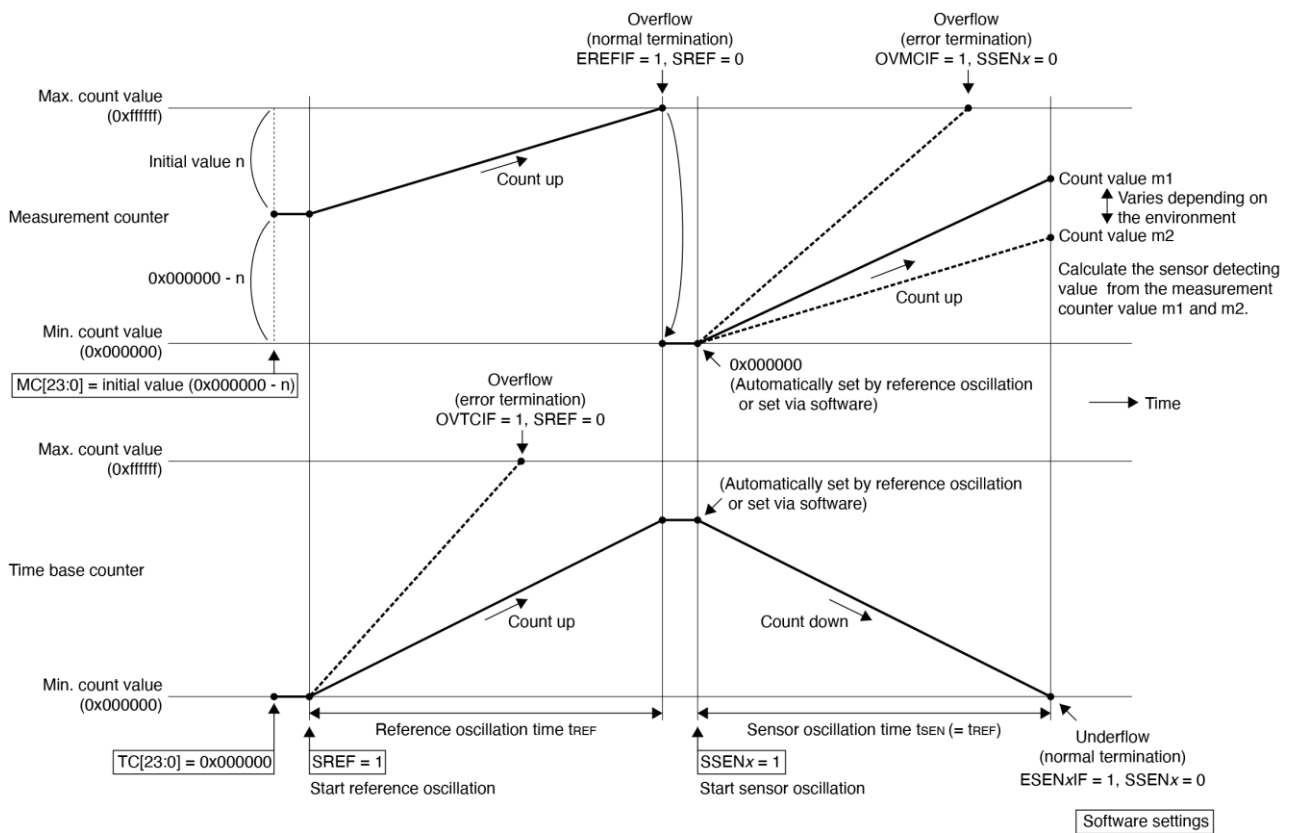


Figure 17.4.4.1 Counter Operations During Reference/Sensor Oscillation

Forced termination

To abort reference oscillation or sensor oscillation, write 0 to the RFCnTRG.SREF bit (reference oscillation), the RFCnTRG.SSENA bit (sensor A oscillation), or the RFCnTRG.SSENB bit (sensor B oscillation) used to start the oscillation. The counters maintain the value at the point they stopped, note, however, that the conversion results cannot be guaranteed if the oscillation is resumed. When resuming oscillation, execute from counter initialization again.

Conversion error

Performing reference oscillation and sensor oscillation with the same resistor and capacitor results $n \approx m$. The difference between n and m is a conversion error. Table 17.4.4.1 lists the error factors. (n: measurement counter initial value, m: measurement counter value at the end of sensor oscillation)

Table 17.4.4.1 Error Factors

Error factor	Influence
External part tolerances	Large
Parasitic capacitance and resistance of the board	Middle
Temperature	Small
Unevenness of IC quality	Small

17.4.5 CR Oscillation Frequency Monitoring Function

The CR oscillation clock (RFCLK) generated during converting operation can be output from the RFCLKOn pin for monitoring. By setting the RFCnCTL.CONEN bit to 1, the RFC2 Ch.n enters continuous oscillation mode that disables oscillation stop conditions to continue oscillating operations. In this case, set the the RFCnTRG.SREF bit (reference oscillation), the RFCnTRG.SSENA bit (sensor A oscillation), or the RFCnTRG.SSENB bit (sensor B oscillation) to 1 to start oscillation. Set the bit to 0 to stop oscillation. Using this function helps easily measure the CR oscillation clock frequency. Furthermore, setting the RFCnCTL.RFCLKMD bit to 1 changes the output clock to the divided-by-two RFCLK clock.

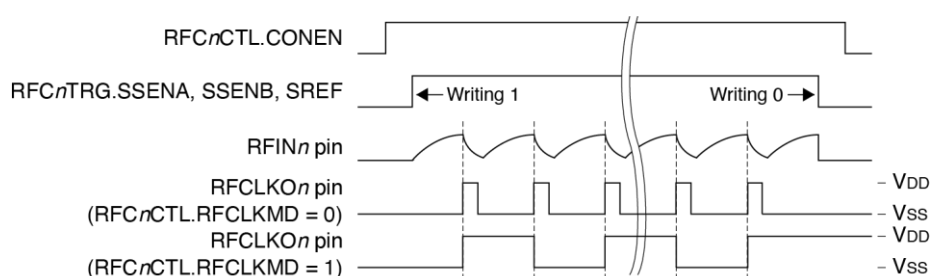


Figure 17.4.5.1 CR Oscillation Clock (RFCLK) Waveform

17.5 Interrupts

The RFC2 has a function to generate the interrupts shown in Table 17.5.1.

Table 17.5.1 RFC2 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Reference oscillation completion	RFCnINTF.EREFIF	When reference oscillation has been completed normally due to a measurement counter overflow	Writing 1
Sensor A oscillation completion	RFCnINTF.ESENAIF	When sensor A oscillation has been completed normally due to the time base counter reaching 0x000000	Writing 1
Sensor B oscillation completion	RFCnINTF.ESENBIF	When sensor B oscillation has been completed normally due to the time base counter reaching 0x000000	Writing 1
Measurement counter overflow error	RFCnINTF.OVMCIF	When sensor oscillation has been terminated abnormally due to a measurement counter overflow	Writing 1
Time base counter overflow error	RFCnINTF.OVTCIF	When reference oscillation has been terminated abnormally due to a time base counter overflow	Writing 1

The RFC2 provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

17.6 Control Registers

RFC2 Ch.n Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnCLK	15–9	–	0x00	–	R	–
	8	DBRUN	1	H0	R/W	–
	7–6	–	0x0	–	R	–
	5–4	CLKDIV[1:0]	0x0	H0	R/W	–
	3–2	–	0x0	–	R	–
	1–0	CLKSRC[1:0]	0x0	H0	R/W	–

17 R/F Converter (RFC2)

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the RFC2 operating clock is supplied in DEBUG mode or not.

1 (R/W): Clock supplied in DEBUG mode

0 (R/W): No clock supplied in DEBUG mode

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

These bits select the division ratio of the RFC2 operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the RFC2.

Table 17.6.1 Clock Source and Division Ratio Settings

RFC _n CLK. CLKDIV[1:0] bits	WDTCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x3	1/8	1/1	1/8	1/1
0x2	1/4		1/4	
0x1	1/2		1/2	
0x0	1/1		1/1	

(Note) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

Note: The RFC_nCLK register settings can be altered only when the RFC_nCTL.MODEN bit = 0.

RFC2 Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC _n CTL	15–9	–	0x00	–	R	–
	8	RFCLKMD	0	H0	R/W	
	7	CONEN	0	H0	R/W	
	6	EVTEN	0	H0	R/W	
	5–4	SMODE[1:0]	0x0	H0	R/W	
	3–1	–	0x0	–	R	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 RFCLKMD

This bit sets the RFCLK_{On} pin to output the divided-by-two oscillation clock.

1 (R/W): Divided-by-two clock output

0 (R/W): Oscillation clock output

For more information, refer to “CR Oscillation Frequency Monitoring Function.”

Bit 7 CONEN

This bit disables the automatic CR oscillation stop function to enable continuous oscillation function.

1 (R/W): Enable continuous oscillation

0 (R/W): Disable continuous oscillation

For more information, refer to “CR Oscillation Frequency Monitoring Function.”

Bit 6 EVTEN

This bit enables external clock input mode (event counter mode).

1 (R/W): External clock input mode

0 (R/W): Normal mode

For more information, refer to “Operating Modes.”

Note: Do not input an external clock before the RFC_nCTL.EVTEN bit is set to 1. The RFIN_n pin is pulled down to V_{SS} level when the port function is switched for the R/F converter.

Bits 5–4 SMODE[1:0]

These bits configure the oscillation mode. For more information, refer to “Operating Modes.”

Table 17.6.2 Oscillation Mode Selection

RFCnCTL.SMODE[1:0] bits	Oscillation mode
0x3, 0x2	Reserved (setting not allowed)
0x1	AC oscillation mode for resistive sensor measurements
0x0	DC oscillation mode for resistive sensor measurements

Bits 3–1 Reserved**Bit 0 MODEN**

This bit enables the RFC2 operations.

1 (R/W): Enable RFC2 operations (The operating clock is supplied.)

0 (R/W): Disable RFC2 operations (The operating clock is stopped.)

Note: If the RFCnCTL.MODEN bit is altered from 1 to 0 during R/F conversion, the counter value being converted cannot be guaranteed. R/F conversion cannot be resumed.

RFC2 Ch.n Oscillation Trigger Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnTRG	15–8	–	0x00	–	R	–
	7–3	–	0x00	–	R	
	2	SSENB	0	H0	R/W	
	1	SSENA	0	H0	R/W	
	0	SREF	0	H0	R/W	

Bits 15–3 Reserved**Bit 2 SSENB**

This bit controls CR oscillation for sensor B. This bit also indicates the CR oscillation status.

1 (W): Start oscillation

0 (W): Stop oscillation

1 (R): Being oscillated

0 (R): Stopped

Note: Writing 1 to the RFCnTRG.SSENB bit does not start oscillation when the RFCnCTL.SMODE[1:0] bits = 0x1 (AC oscillation mode for resistive sensor measurements).

Bit 1 SSENA

This bit controls CR oscillation for sensor A. This bit also indicates the CR oscillation status.

1 (W): Start oscillation

0 (W): Stop oscillation

1 (R): Being oscillated

0 (R): Stopped

Bit 0 SREF

This bit controls CR oscillation for the reference resistor. This bit also indicates the CR oscillation status.

1 (W): Start oscillation

0 (W): Stop oscillation

1 (R): Being oscillated

0 (R): Stopped

- Notes:
- Settings in this register are all ineffective when the RFCnCTL.MODEN bit = 0 (RFC2 operation disabled).
 - When writing 1 to the RFCnTRG.SREF bit, the RFCnTRG.SSENA bit, or the RFCnTRG.SSENB bit to start oscillation, be sure to avoid having more than one bit set to 1.
 - Be sure to clear the interrupt flags (RFCnINTF.EREFIF bit, RFCnINTF.ESENAIF bit, RFCnINTF.ESENBIF bit, RFCnINTF.OVMCIF bit, and RFCnINTF.OVTCIF bit) before starting oscillation using this register.

17 R/F Converter (RFC2)

RFC2 Ch.n Measurement Counter Low and High Registers

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC _n MCL	15–0	MC[15:0]	0x0000	H0	R/W	–
RFC _n MCH	15–8	–	0x00	–	R	–
	7–0	MC[23:16]	0x00	H0	R/W	–

Or

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC _n MCL	31–24	–	0x00	–	R	–
RFC _n MCH	23–0	MC[23:0]	0x000000	H0	R/W	–

Bits 31–24 Reserved

Bits 23–0 MC[23:0]

Measurement counter data can be read and written through these bits.

Note: The measurement counter must be set from the low-order value (RFC_nMCL.MC[15:0] bits) first when data is set using a 16-bit access instruction. The counter may not be set to the correct value if the high-order value (RFC_nMCH.MC[23:16] bits) is written first.

RFC2 Ch.n Time Base Counter Low and High Registers

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC _n TCL	15–0	TC[15:0]	0x0000	H0	R/W	–
RFC _n TCH	15–8	–	0x00	–	R	–
	7–0	TC[23:16]	0x00	H0	R/W	–

Or

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC _n TCL	31–24	–	0x00	–	R	–
RFC _n TCH	23–0	TC[23:0]	0x000000	H0	R/W	–

Bits 31–24 Reserved

Bits 23–0 TC[23:0]

Time base counter data can be read and written through these bits.

Note: The time base counter must be set from the low-order value (RFC_nTCL.TC[15:0] bits) first when data is set using a 16-bit access instruction. The counter may not be set to the correct value if the high-order value (RFC_nTCH.TC[23:16] bits) is written first.

RFC2 Ch.n Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC _n INTF	15–8	–	0x00	–	R	–
	7–5	–	0x0	–	R	–
	4	OVTCIF	0	H0	R/W	Cleared by writing 1.
	3	OVMCIF	0	H0	R/W	
	2	ESENBIF	0	H0	R/W	
	1	ESENAIF	0	H0	R/W	
0	EREFIF	0	H0	R/W		

Bits 15–5 Reserved

Bit 4 OVTCIF

Bit 3 OVMCIF

Bit 2 ESENBIF

Bit 1 ESENAIF

Bit 0 EREFIF

These bits indicate the RFC2 interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:

RFC n INTF.OVTCIF bit: Time base counter overflow error interrupt

RFC n INTF.OVMCIE bit: Measurement counter overflow error interrupt

RFC n INTF.ESENBIF bit: Sensor B oscillation completion interrupt

RFC n INTF.ESENAIF bit: Sensor A oscillation completion interrupt

RFC n INTF.EREFIF bit: Reference oscillation completion interrupt

RFC2 Ch. n Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC n INTE	15–8	–	0x00	–	R	–
	7–5	–	0x0	–	R	
	4	OVTCIE	0	H0	R/W	
	3	OVMCIE	0	H0	R/W	
	2	ESENBIE	0	H0	R/W	
	1	ESENAIE	0	H0	R/W	
	0	EREFIE	0	H0	R/W	

Bits 15–5 Reserved

Bit 4 OVTCIE

Bit 3 OVMCIE

Bit 2 ESENBIE

Bit 1 ESENAIE

Bit 0 EREFIE

These bits enable RFC2 interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

RFC n INTE.OVTCIE bit: Time base counter overflow error interrupt

RFC n INTE.OVMCIE bit: Measurement counter overflow error interrupt

RFC n INTE.ESENBIE bit: Sensor B oscillation completion interrupt

RFC n INTE.ESENAIE bit: Sensor A oscillation completion interrupt

RFC n INTE.EREFIE bit: Reference oscillation completion interrupt

18 Charge Pump Regulator (CHREG)

18.1 Overview

CHREG is a circuit that boosts the power supply voltage (V_{DD}) and provides the boosted voltage as output. The boosted voltage can be used as a power supply for external devices such as a buzzer and LEDs.

The main features of CHREG are as follows:

- Generates the power supply voltage for driving the buzzer and LEDs
- Automatically generates the timing required for operating the charge pump

Note: The buzzer volume or LED brightness may be affected when the buzzer and LED are driven simultaneously using this boosted voltage. Therefore, exclusive use is recommended.

Figure 18.1.1 shows the CHREG configuration.

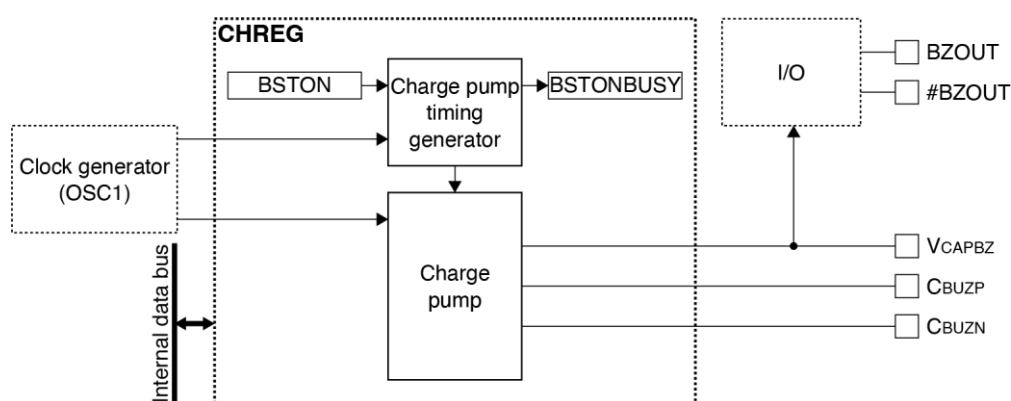


Figure 18.1.1 CHREG Configuration

18.2 Input/Output Pins and External Connections

18.2.1 List of Input/Output Pins

Table 18.2.1.1 lists the CHREG pins.

Table 18.2.1.1 List of CHREG Pins

Pin name	I/O	Initial status	Function
BZOUT	O	Hi-Z	Sound generator buzzer output pin
#BZOUT	O	Hi-Z	Sound generator inverted buzzer output pin
VCAPBZ	A	V_{DD}	Power supply output pin for buzzer/LED drive
CBUZP	A	V_{DD}	Boosting capacitor connection pin for buzzer/LED drive power
CBUZN	A	Hi-Z	Boosting capacitor connection pin for buzzer/LED drive power

18.2.2 External Connections

Figure 18.2.2.1 shows an example of external connections among CHREG, a buzzer, LEDs, and capacitors.

18 Charge Pump Regulator (CHREG)

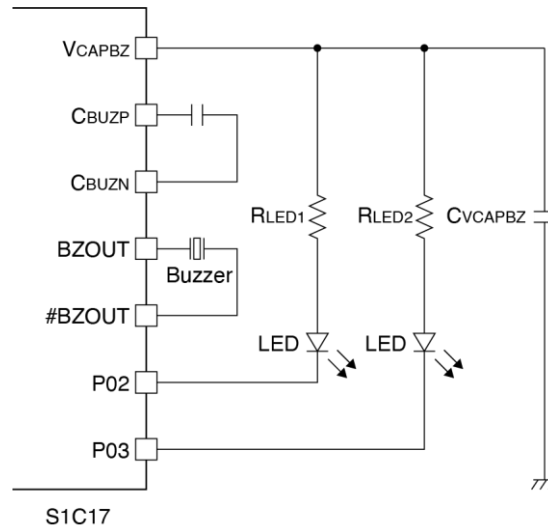


Figure 18.2.2.1 Example of External Connection for CHREG

18.3 Operations

18.3.1 Charge Pump ON/OFF Procedure

The charge pump can be turned ON or OFF using the following steps:

1. Enable OSC1 oscillation and wait for the oscillation to stabilize.
2. Write 1 to the CHREGCTL.BSTON bit.

The CHREGCTL.BSTONBSY bit is set to 1, and the voltage boosting operation starts.

3. When the boosting is completed, the CHREGCTL.BSTONBSY bit is cleared to 0, and the typical 5 V output ($V_{DD} \times 2$) is provided from the buzzer/LED power output pin.

(For details on the output voltage, refer to “Charge Pump Regulator (CHREG) Characteristics” in the “Electrical Characteristics” chapter.)

4. Write 0 to the CHREGCTL.BSTON bit.

The charge pump turns OFF, and the output returns to V_{DD} .

Note: Do not stop OSC1 oscillation while the charge pump is ON (CHREGCTL.BSTONBSY = 1).

18.3.2 Charge Pump Settings

The CHREGCONF register allows configuration of the charge pump, but the default settings should be used.

18.4 Control Registers

CHREG Configuration Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CHREGCONF	15–0	–	0x0030	–	R	Do not change the value.

Bits 15–0 Reserved

CHREG Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CHREGCTL	15–9	–	0x00	–	R	–
	8	BSTONBSY	0	H0	R	
	7–1	–	0x0	–	R	
	0	BSTON	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 BSTONBSY

This bit indicates the operating status of the charge pump.

1 (R): Boosting operation in progress

0 (R): Boosting completed or OFF

Bits 7–1 Reserved

Bit 0 BSTON

This bit turns the charge pump ON or OFF.

1 (W): Start boosting operation

0 (W): OFF

1 (R): ON (boosting completed) or in progress

0 (R): OFF

19 Multiplier/Divider (COPRO2)

19.1 Overview

COPRO2 is the coprocessor that provides multiplier/divider functions. The features of COPRO2 are listed below.

- Multiplication: Supports signed/unsigned multiplications.
(16 bits \times 16 bits = 32 bits)
Can be executed in 1 cycle.
- Multiplication and accumulation (MAC): Supports signed/unsigned MAC operations with overflow detection function. (16 bits \times 16 bits + 32 bits = 32 bits)
Can be executed in 1 cycle.
- Division: Supports signed/unsigned divisions.
(32 bits \div 32 bits = 32 bits with 32-bit remainder)
Can be executed in 17 to 20 cycles.
Overflow detection and division by zero processing are not supported.

Figure 19.1.1 shows the COPRO2 configuration.

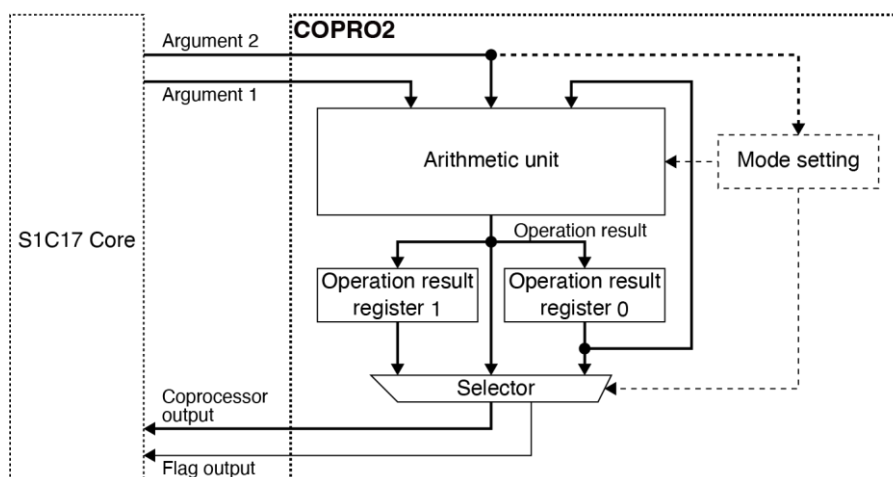


Figure 19.1.1 COPRO2 Configuration

19.2 Operation Mode and Output Mode

COPRO2 operates according to the operation mode specified by the application program. As listed in Table 19.2.1, COPRO2 supports 11 operations.

The multiplication, division and MAC results are 32-bit data, therefore, the S1C17 Core cannot read them in one access cycle. The output mode is provided to specify the high-order 16 bits or low-order 16 bits of the operation result register 0 or 1 to be read from COPRO2.

The operation and output modes can be specified with a 7-bit data by writing it to the mode setting register in COPRO2. Use a "ld.cw" instruction for this writing.

```
ld.cw %rd,%rs    %rs[6:0] is written to the mode setting register. (%rd: not used)
ld.cw %rd,imm7  imm7[6:0] is written to the mode setting register. (%rd: not used)
```

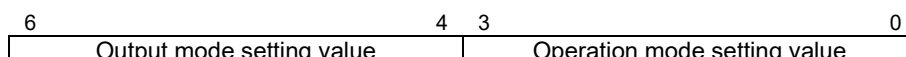


Figure 19.2.1 Mode Setting Register

19 Multiplier/Divider (COPRO2)

Table 19.2.1 Mode Settings

Setting value (D[6:4])	Output mode	Setting value (D[3:0])	Operation mode
0x0	16 low-order bits output mode 0 The low-order 16 bits of the operation result register 0 can be read as the coprocessor output.	0x0	Initialize mode 0 Clears the operation result registers 0 and 1 to 0x0.
0x1	16 high-order bits output mode 0 The high-order 16 bits of the operation result register 0 can be read as the coprocessor output.	0x1	Initialize mode 1 Loads the 16-bit augend into the low-order 16 bits of the operation result register 0.
0x2	16 low-order bits output mode 1 The low-order 16 bits of the operation result register 1 can be read as the coprocessor output.	0x2	Initialize mode 2 Loads the 32-bit data into the operation result register 0.
0x3	16 high-order bits output mode 1 The high-order 16 bits of the operation result register 1 can be read as the coprocessor output.	0x3	Operation result read mode Outputs the data in the operation result registers 0 and 1 without computation.
0x4–0x7	Reserved	0x4	Unsigned multiplication mode Performs unsigned multiplication.
		0x5	Signed multiplication mode Performs signed multiplication.
		0x6	Unsigned MAC mode Performs unsigned MAC operation.
		0x7	Signed MAC mode Performs signed MAC operation.
		0x8	Unsigned division mode Performs unsigned division.
		0x9	Signed division mode Performs signed division.
		0xa	Initialize mode 3 Loads the 32-bit data into the operation result register 1.
		0xb–0xf	Reserved

19.3 Multiplication

The multiplication function performs “A (32 bits) = B (16 bits) × C (16 bits).”

The following shows a procedure to perform a multiplication:

1. Set the mode to 0x04 (unsigned multiplication, 16 low-order bits output mode 0) or 0x05 (signed multiplication, 16 low-order bits output mode 0).
2. Send the 16-bit multiplicand (B) and 16-bit multiplier (C) to COPRO2 using a “ld.ca” instruction.
3. Read the one-half result (16 low-order bits = A[15:0]) and the flag status.
4. Set the mode to 0x13 (operation result read, 16 high-order bits output mode 0).
5. Read another one-half result (16 high-order bits = A[31:16]).

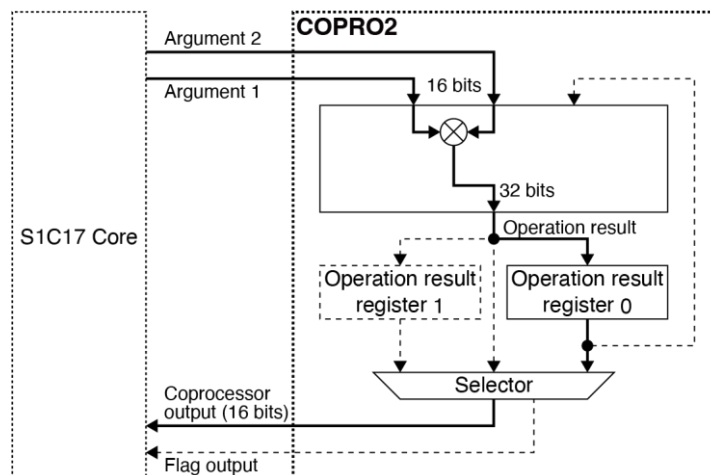


Figure 19.3.1 Data Path in Multiplication Mode

Table 19.3.1 Operation in Multiplication Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x04 or 0x05	ld.ca %rd,%rs	res0[31:0] ← %rd × %rs %rd ← res0[15:0]	psr (CVZN) ← 0b0000	The operation result register 0 keeps the operation result until it is rewritten by other operation.
	(ext imm9) ld.ca %rd,imm7	res0[31:0] ← %rd × imm7/16 %rd ← res0[15:0]		
0x14 or 0x15	ld.ca %rd,%rs	res0[31:0] ← %rd × %rs %rd ← res0[31:16]		
	(ext imm9) ld.ca %rd,imm7	res0[31:0] ← %rd × imm7/16 %rd ← res0[31:16]		

res0: operation result register 0

Example:

```
ld.cw %r0, 0x04 ; Sets the mode (unsigned multiplication mode and 16 low-order bits output mode 0).
ld.ca %r0, %r1 ; Performs “res0[31:0] = %r0[15:0] × %r1[15:0]” and loads the 16 low-order bits of the
                ; result to %r0.
ld.cw %r0, 0x13 ; Sets the mode (operation result read mode and 16 high-order bits output mode 0).
ld.ca %r1, %r0 ; Loads the 16 high-order bits of the result to %r1.
```

19.4 Division

The division function performs “A (32 bits) = B (32 bits) ÷ C (32 bits), D (32 bits) = remainder.”

The following shows a procedure to perform a division:

1. Set the mode to 0x02 (initialize mode 2).
2. Set the 32-bit dividend (B) to the operation result register 0 using a “ld.cf” instruction.
3. Set the mode to 0x08 (unsigned division, 16 low-order bits output mode 0) or 0x09 (signed division, 16 low-order bits output mode 0).
4. Send the 32-bit divisor (C) to COPRO2 using a “ld.ca” instruction.
5. Read the one-half result (16 low-order bits = A[15:0]) of the operation result register 0 (quotient) and the flag status.
6. Set the mode to 0x13 (operation result read, 16 high-order bits output mode 0).
7. Read another one-half result (16 high-order bits = A[31:16]) of the operation result register 0 (quotient).
8. Set the mode to 0x23 (operation result read, 16 low-order bits output mode 1).
9. Read the one-half result (16 low-order bits = D[15:0]) of the operation result register 1 (remainder).
10. Set the mode to 0x33 (operation result read, 16 high-order bits output mode 1).
11. Read another one-half result (16 high-order bits = D[31:16]) of the operation result register 1 (remainder).

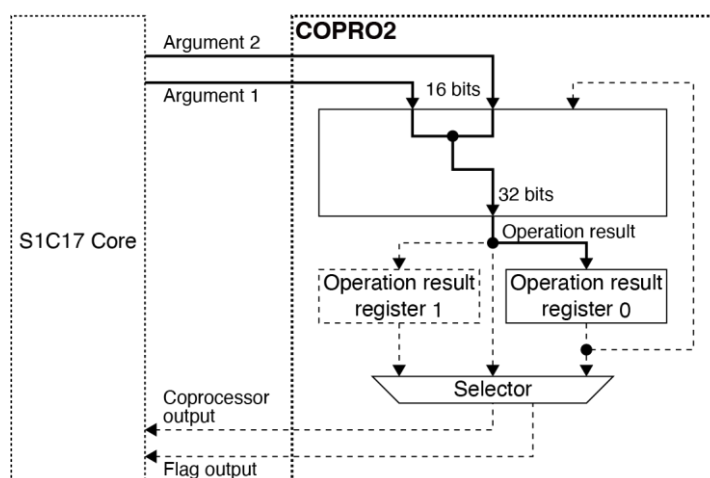


Figure 19.4.1 Data Path in Initialize Mode 2

19 Multiplier/Divider (COPRO2)

Table 19.4.1 Initializing the Operation Result Register 0 (32 bits)

Mode setting value	Instruction	Operations	Remarks
0x02	ld.cf %rd,%rs	res0[31:16] ← %rd res0[15:0] ← %rs	
	(ext imm9) ld.cf %rd,imm7	res0[31:16] ← %rd res0[15:0] ← imm7/16	

res0: operation result register 0

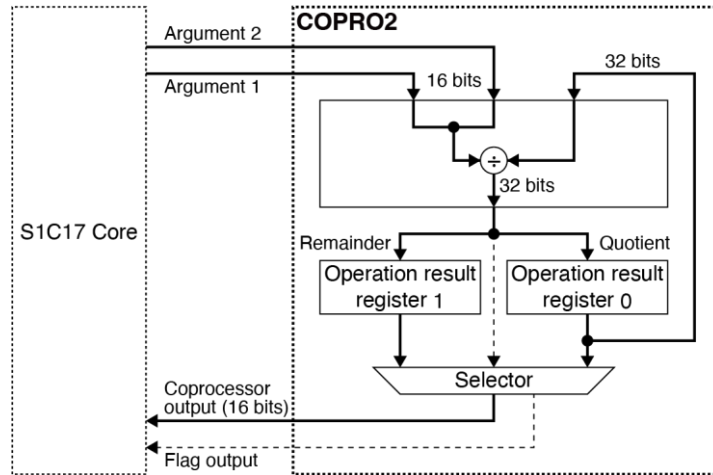


Figure 19.4.2 Data Path in Division Mode

Table 19.4.2 Operation in Division Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x08 or 0x09	ld.ca %rd,%rs	res0[31:0] + {%rd, %rs} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res0[15:0] (Quotient)	psr (CVZN) ← 0b0000	The operation result registers 0 and 1 keep the operation results until they are rewritten by other operation. COPRO2 does not support 0 ÷ 0 division.
	(ext imm9) ld.ca %rd,imm7	res0[31:0] + {%rd, imm7/16} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res0[15:0] (Quotient)		
0x18 or 0x19	ld.ca %rd,%rs	res0[31:0] + {%rd, %rs} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res0[31:16] (Quotient)		
	(ext imm9) ld.ca %rd,imm7	res0[31:0] + {%rd, imm7/16} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res0[31:16] (Quotient)		
0x28 or 0x29	ld.ca %rd,%rs	res0[31:0] + {%rd, %rs} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res1[15:0] (Remainder)		
	(ext imm9) ld.ca %rd,imm7	res0[31:0] + {%rd, imm7/16} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res1[15:0] (Remainder)		
0x38 or 0x39	ld.ca %rd,%rs	res0[31:0] + {%rd, %rs} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res1[31:16] (Remainder)		
	(ext imm9) ld.ca %rd,imm7	res0[31:0] + {%rd, imm7/16} res0[31:0] ← Quotient res1[31:0] ← Remainder %rd ← res1[31:16] (Remainder)		

res0: operation result register 0, res1: operation result register 1

Example:

```
ld.cw %r0, 0x02 ; Sets the mode (initialize mode 2).
ld.cf %r0, %r1 ; Set the dividend {%r0, %r1} to the operation result register 0.
ld.cw %r0, 0x08 ; Sets the mode (unsigned division mode and 16 low-order bits output mode 0).
ld.ca %r0, %r1 ; Performs "res0[31:0] (quotient), res1[31:0] (remainder) = res0[31:0] ÷
                {%r0[15:0], %r1[15:0]}" and loads the 16 low-order bits of the result (quotient) to %r0.
ld.ca %r1, %r0 ; Loads the 16 low-order bits of the result (quotient) to %r1.
ld.cw %r0, 0x13 ; Sets the mode (operation result read mode and 16 high-order bits output mode 0).
ld.ca %r2, %r0 ; Loads the 16 high-order bits of the result (quotient) to %r2.
ld.cw %r0, 0x23 ; Sets the mode (operation result read mode and 16 low-order bits output mode 1).
ld.ca %r3, %r0 ; Loads the 16 low-order bits of the result (remainder) to %r3.
ld.cw %r0, 0x33 ; Sets the mode (operation result read mode and 16 high-order bits output mode 1).
ld.ca %r4, %r0 ; Loads the 16 high-order bits of the result (remainder) to %r4.
```

19.5 MAC

The MAC (multiplication and accumulation) function performs “ A (32 bits) = B (16 bits) \times C (16 bits) + A (32 bits).” The following shows a procedure to perform a MAC operation:

- Set the initial value (A) to the operation result register 0.
 - To clear the operation result registers ($A = 0$):
 - Set the mode to 0x00 (initialize mode 0). (It is not necessary to send 0x00 to COPRO2 with another instruction.)
 - To load a 16-bit value to the operation result register 0:
 - Set the operation mode to 0x01 (initialize mode 1) and then send the initial value (16 bits) to COPRO2 using a “ld.cf” instruction.
 - To load a 32-bit value to the operation result register 0:
 - Set the operation mode to 0x02 (initialize mode 2) and then send the initial value (32 bits) to COPRO2 using a “ld.cf” instruction.
- Set the mode to 0x06 (unsigned MAC, 16 low-order bits output mode 0) or 0x07 (signed MAC, 16 low-order bits output mode 0).
- Repeat sending the 16-bit multiplicand (B) and 16-bit multiplier (C) to COPRO2 the number of times required using a “ld.ca” instruction.
- Read the one-half result (16 low-order bits = $A[15:0]$) and the flag status.
- Set the mode to 0x13 (operation result read, 16 high-order bits output mode).
- Read another one-half result (16 high-order bits = $A[31:16]$).

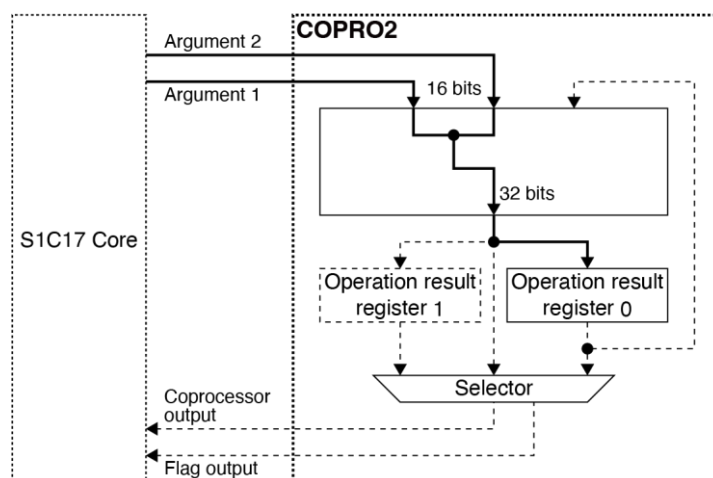


Figure 19.5.1 Data Path in Initialize Mode

19 Multiplier/Divider (COPRO2)

Table 19.5.1 Initializing the Operation Result Register 0

Mode setting value	Instruction	Operations	Remarks
0x00	-	res0[31:0] ← 0x0 res1[31:0] ← 0x0	Setting the operating mode executes the initialization without sending data.
0x01	ld.cf %rd,%rs	res0[31:16] ← 0x0 res0[15:0] ← %rs	
	(ext imm9) ld.cf %rd,imm7	res0[31:16] ← 0x0 res0[15:0] ← imm7/16	
0x02	ld.cf %rd,%rs	res0[31:16] ← %rd res0[15:0] ← %rs	
	(ext imm9) ld.cf %rd,imm7	res0[31:16] ← %rd res0[15:0] ← imm7/16	

res0: operation result register 0, res1: operation result register 1

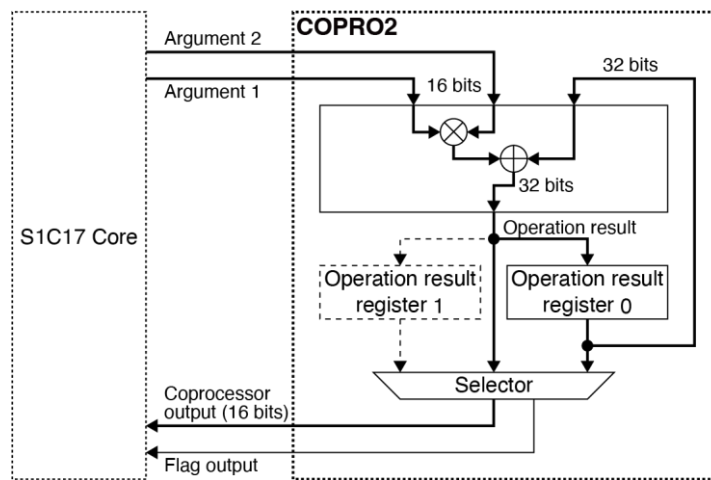


Figure 19.5.2 Data Path in MAC Mode

Table 19.5.2 Operation in MAC Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x06 or 0x07	ld.ca %rd,%rs	res0[31:0] ← %rd × %rs + res0[31:0] %rd ← res0[15:0]	psr (CVZN) ← 0b0100 if an overflow has occurred	The operation result register 0 keeps the operation result until it is rewritten by other operation. Overflow can be detected only in signed MAC mode (it does not occur in unsigned MAC mode).
	(ext imm9) ld.ca %rd,imm7	res0[31:0] ← %rd × imm7/16 + res0[31:0] %rd ← res0[15:0]		
0x16 or 0x17	ld.ca %rd,%rs	res0[31:0] ← %rd × %rs + res0[31:0] %rd ← res0[31:16]	Otherwise psr (CVZN) ← 0b0000	
	(ext imm9) ld.ca %rd,imm7	res0[31:0] ← %rd × imm7/16 + res0[31:0] %rd ← res0[31:16]		

res0: operation result register 0

Example:

```
ld.cw %r0, 0x00 ; Sets the mode (initialize mode 0) to clear the operation result register 0 to 0x0000.
ld.cw %r0, 0x07 ; Sets the mode (signed MAC mode and 16 low-order bits output mode 0).
ld.ca %r0, %r1 ; Performs "res0[31:0] = %r0[15:0] × %r1[15:0] + res0[31:0]" and loads the 16 low-order
                ; bits of the result to %r0.
ld.cw %r0, 0x13 ; Sets the mode (operation result read mode and 16 high-order bits output mode 0).
ld.ca %r1, %r0 ; Loads the 16 high-order bits of the result to %r1.
```

Conditions to set the overflow (V) flag

An overflow occurs in a signed MAC operation and the overflow (V) flag is set to 1 when the signs of the multiplication result, operation result register value, and multiplication & accumulation result match the following conditions:

Table 19.5.3 Conditions to Set the Overflow (V) Flag

Mode setting value	Sign of multiplication result	Sign of operation result register value	Sign of multiplication & accumulation result
0x07	0 (positive)	0 (positive)	1 (negative)
0x07	1 (negative)	1 (negative)	0 (positive)

An overflow occurs when a MAC operation performs addition of positive values and a negative value results, or it performs addition of negative values and a positive value results. The coprocessor holds the operation result until the overflow (V) flag is cleared.

Conditions to clear the overflow (V) flag

The overflow (V) flag that has been set will be cleared when an overflow has not been occurred during execution of the “ld.ca” instruction for MAC operation or when the “ld.ca” or “ld.cf” instruction is executed in an operation mode other than operation result read mode.

19.6 Reading Operation Results

The “ld.ca” instruction cannot load a 32-bit operation result to a CPU register, so a multiplication, division or MAC operation returns the one-half (16 bits according to the output mode) result (A[15:0] or A[31:16]) and the flag status to the CPU registers. Another one-half should be read by setting COPRO2 into operation result read mode. The operation result register keeps the loaded operation result until it is rewritten by other operation.

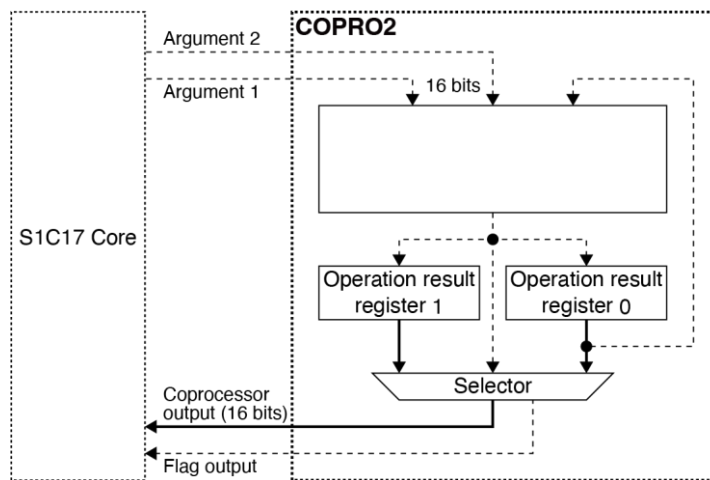


Figure 19.6.1 Data Path in Operation Result Read Mode

Table 19.6.1 Operation in Operation Result Read Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x03	ld.ca %rd,%rs	%rd ← res[15:0]	psr (CVZN) ← 0b0000	This operation mode does not affect the operation result registers 0 and 1.
	ld.ca %rd,imm7	%rd ← res[15:0]		
0x13	ld.ca %rd,%rs	%rd ← res[31:16]		
	ld.ca %rd,imm7	%rd ← res[31:16]		
0x23	ld.ca %rd,%rs	%rd ← res1[15:0]		
	ld.ca %rd,imm7	%rd ← res1[15:0]		
0x33	ld.ca %rd,%rs	%rd ← res1[31:16]		
	ld.ca %rd,imm7	%rd ← res1[31:16]		

res0: operation result register 0, res1: operation result register 1

20 Electrical Characteristics

20.1 Absolute Maximum Ratings

(V_{SS} = 0 V)

Item	Symbol	Condition	Rated value	Unit	
Power supply voltage	V _{DD}		-0.3 to 4.0	V	
Flash programming voltage	V _{PP}		-0.3 to 8.0	V	
LCD power supply voltage	V _{C1}		-0.3 to 7.0	V	
	V _{C2}		-0.3 to 7.0	V	
	V _{C3}		-0.3 to 7.0	V	
Input voltage	V _I	P00-07, P10-17, P20-25, PD0-D1, #RESET	-0.3 to V _{DD} + 0.5	V	
Output voltage	V _O	P00-07, P10-17, P20-25, PD0-D2	-0.3 to V _{DD} + 0.5	V	
High level output current	I _{OH}	1 pin	P00-07, P10-17, P20-25, PD0-D2	-10	V
		Total of all pins		-20	mA
Low level output current	I _{OL}	1 pin	P00-07, P10-17, P20-25, PD0-D2	10	mA
		Total of all pins		20	mA
Operating temperature	T _a		-40 to 85	°C	
Storage temperature	T _{stg}		-65 to 125	°C	

20.2 Recommended Operating Conditions

(V_{SS} = 0 V) **

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	
Power supply voltage	V _{DD}	For normal operation	1.2	–	3.6	V	
		For Flash programming	When V _{PP} is supplied externally	2.2	–	3.6	V
			When V _{PP} is generated internally	2.2	–	3.6	V
		For EEPROM programming	V _{PP} is generated internally	2.2	–	3.6	V
Flash programming voltage	V _{PP}		7.3	7.5	7.7	V	
LCD power supply voltage	V _{C1}	When an external voltage is applied V _{C1} ≤ V _{C2} ≤ V _{C3} , V _{C1} ≤ V _{DD}	–	1.0	1.9	V	
			–	2.0	3.8	V	
			–	3.0	5.7	V	
OSC1 oscillator oscillation frequency	f _{OSC1}	Internal oscillator	–	32	–	kHz	
OSC3 oscillator oscillation frequency	f _{OSC3}	Internal oscillator	V _{DD} = 1.2 to 3.6 V	0.5	–	1.1	MHz
			V _{DD} = 1.6 to 3.6 V	0.5	–	4.2	MHz
			V _{DD} = 1.2 to 3.6 V	0.016	–	4.2	MHz
EXOSC external clock frequency	f _{EXOSC}	When supplied from an external oscillator	V _{DD} = 1.6 to 3.6 V	0.016	–	4.2	MHz
Bypass capacitor between V _{SS} and V _{DD}	C _{PW1}		–	3.3	–	μF	
Capacitor between V _{SS} and V _{D1}	C _{PW2}		–	1	–	μF	
Capacitors between V _{SS} and V _{C1-3}	C _{LCD1-3}	*2	–	1	–	μF	
Capacitor between C _{P1} and C _{P2}	C _{LCD4}	*2	–	1	–	μF	
Capacitor between V _{SS} and V _{CAPRF}	C _{VCAPRF}		–	0.1	–	μF	
Capacitor between V _{SS} and V _{CAPBZ}	C _{VCAPBZ}		–	2.2	–	μF	
Capacitor between C _{BUZP} and C _{BUZN}	C _{BUZ}		–	0.22	–	μF	
DSIO pull-up resistor	R _{DBG}	*3	–	10	–	kΩ	
Capacitor between V _{SS} and V _{PP}	C _{VPP}		–	0.1	–	μF	

*1 The potential variation of the V_{SS} voltage should be suppressed to within ±0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count).

*2 The V_{C1}–V_{C3} pins can be left open when the LCD driver is not used. Furthermore, the C_{P1}–C_{P2} pins are not used.

*3 R_{DBG} is not required when using the DSIO pin as a general-purpose I/O port.

20 Electrical Characteristics

20.3 Current Consumption

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = 25^\circ\text{C}$, EXOSC = OFF, PWGVD1CTL.REGMODE[1:0] bits = 0x0 (automatic mode), FLASHCWAIT.RDWAIT[1:0] bits = 0x1 (2 cycles), PWGVD1CTL.REGSEL[1:0]bits = 0x1 ($V_{D1} = 1.4$ V)

Item	Symbol	Condition	Ta	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I _{SLEEP1}	IOSC = OFF, OSC1 = OFF, OSC3 = OFF, VDD = 1.5V PWGVD1CTL.REGSEL[1:0] bits = 0x0 ($V_{D1} = 1.2$ V)	25°C	–	109	–	nA
			85°C	–	530	9,000	nA
	I _{SLEEP2}	IOSC = OFF, OSC1 = OFF, OSC3 = OFF PWGVD1CTL.REGSEL[1:0] bits = 0x0 ($V_{D1} = 1.2$ V)	25°C	–	116	–	nA
			85°C	–	580	9,000	nA
	I _{SLEEP3}	IOSC = OFF, OSC1 = OFF, OSC3 = OFF	25°C	–	118	–	nA
			85°C	–	639	9,000	nA
Current consumption in HALT mode	I _{HALT10}	IOSC = ON, OSC1 = 32 kHz, OSC3 = OFF		–	26	55	μA
	I _{HALT20}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = OFF PWGVD1CTL.PWGSEL[1:0] bits = 0x0 ($V_{D1} = 1.2$ V)		–	1.28	4.5	μA
	I _{HALT21}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = OFF		–	1.30	4.5	μA
	I _{HALT30}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = 1 MHz		–	65	135	μA
Current consumption in Run mode	I _{RUN10} ^{*1}	IOSC = ON, OSC1 = 32 kHz, OSC3 = OFF, SYSCLK = IOSC PWGVD1CTL.PWGSEL[1:0] bits = 0x0 ($V_{D1} = 1.2$ V)		–	77	140	μA
	I _{RUN11} ^{*1}	IOSC = ON, OSC1 = 32 kHz, OSC3 = OFF, SYSCLK = IOSC		–	90	170	μA
	I _{RUN12} ^{*1}	IOSC = ON, OSC1 = 32 kHz, OSC3 = OFF, SYSCLK = IOSC, FLASHCWAIT.RDWAIT[1:0] bits = 0x0 (1 cycle)		–	108	250	μA
	I _{RUN20} ^{*1}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = OFF, SYSCLK = OSC1		–	3.82	9.0	μA
	I _{RUN21} ^{*1}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = OFF, SYSCLK = OSC1		–	4.3	10.0	μA
	I _{RUN22} ^{*1}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = OFF, SYSCLK = OSC1, PWGVD1CTL.REGMODE[1:0] bits = 0x2 (normal mode)		–	13.5	25.0	μA
	I _{RUN30} ^{*1}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = 1 MHz, SYSCLK = OSC3		–	160	350	μA
	I _{RUN31} ^{*1}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = 1 MHz, SYSCLK = OSC3, PWGVD1CTL.PWGSEL[1:0] bits = 0x0 ($V_{D1} = 1.2$ V)		–	156	340	μA
	I _{RUN32} ^{*1}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = 1 MHz, SYSCLK = OSC3 FLASHCWAIT.RDWAIT[1:0] bits = 0x0 (1cycle)		–	180	400	μA
	I _{RUN33} ^{*1}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = 1 MHz, SYSCLK = OSC3, running in the RAM		–	160	350	μA
I _{RUN34} ^{*1}	IOSC = OFF, OSC1 = 32 kHz, OSC3 = 4 MHz, SYSCLK = OSC3, $V_{DD} = 1.6$ to 3.6 V		–	440	900	μA	

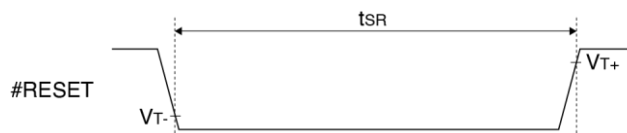
*1 The current consumption values were measured when a test program consisting of 60.5 % ALU instructions, 17 % branch instructions, 12 % RAM read instructions, and 10.5 % RAM write instructions was executed continuously in the Flash memory.

20.4 System Reset Controller (SRC) Characteristics

#RESET pin characteristics

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

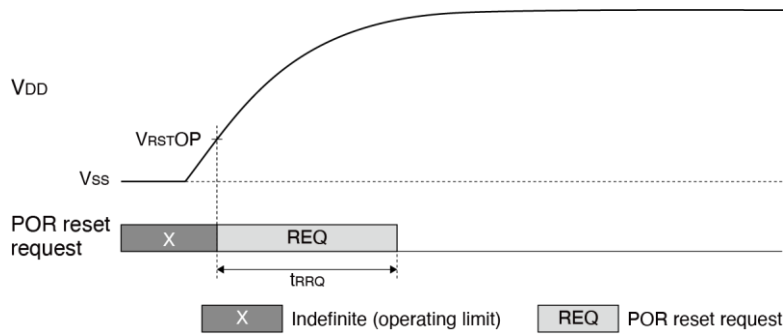
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}		$0.5 \times V_{DD}$	–	$0.8 \times V_{DD}$	V
Low level Schmitt input threshold voltage	V_{T-}		$0.2 \times V_{DD}$	–	$0.5 \times V_{DD}$	V
Schmitt input hysteresis voltage	ΔV_T		20	–	–	mV
Input pull-up resistance	R_{IN}		100	270	500	kΩ
Pin capacitance	C_{IN}		–	–	15	pF
Reset Low pulse width	t_{SR}		5	–	–	μs



POR characteristics

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
POR operating limit voltage	V_{RSTOP}		–	0.5	0.95	V
POR reset request hold time	t_{RRQ}		0.01	–	4	ms



Note: When performing a power-on-reset again after the power is turned off, decrease the V_{DD} voltage to V_{RSTOP} or less.

Reset hold circuit characteristics

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Reset hold time*1	t_{RSTR}		0.5	–	0.9	ms

*1 Time until the internal reset signal is negated after the reset request is canceled.

20.5 Clock Generator (CLG) Characteristics

Oscillator circuit characteristics including resonators change depending on conditions (board pattern, components used, etc.). Use these characteristic values as a reference and perform matching evaluation using the actual printed circuit board.

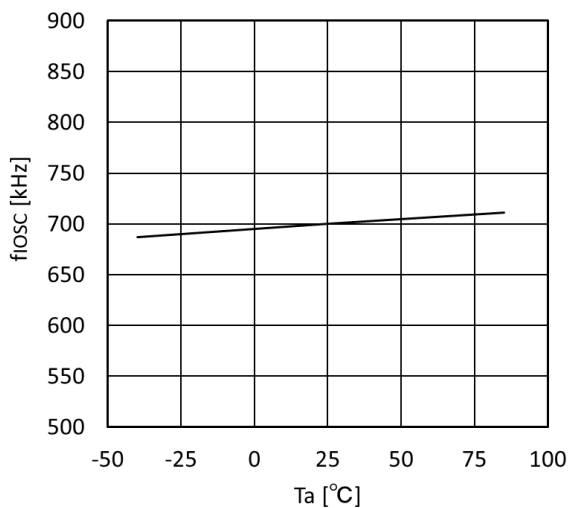
IOSC oscillator circuit characteristics

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

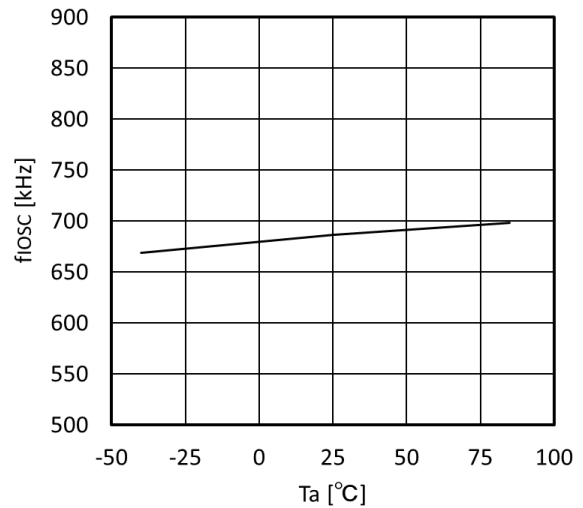
Item	Symbol	Condition	V_{DD}	V_{D1}	T_a	Min.	Typ.	Max.	Unit
Oscillation start time	t_{stal}					–	–	3	μs
Oscillation frequency	f_{IOSC}		1.6 to 3.6 V	1.4V	25°C	685	700	715	kHz
			1.2 to 1.6 V	–		665	700	715	
			1.6 to 3.6 V	1.4V	-40 to 85°C	664	700	735	kHz
			1.2 to 1.6 V	–		630	700	735	

IOSC oscillation frequency-temperature characteristic

$V_{DD} = 1.6$ to 3.6 V, $V_{D1} = 1.4$ V, Typ. Value



$V_{DD} = 1.2$ to 1.6 V, $V_{D1} = 1.2$ V, Typ. Value



20 Electrical Characteristics

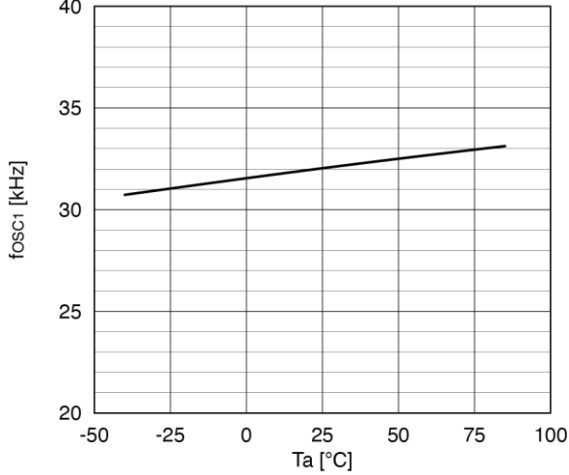
OSC1 oscillator circuit characteristics

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = 25^\circ\text{C}$

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Internal oscillator oscillation frequency	f_{OSC1}		31.04	32	32.96	kHz

OSC1 internal oscillation frequency-temperature characteristic

Typ. value



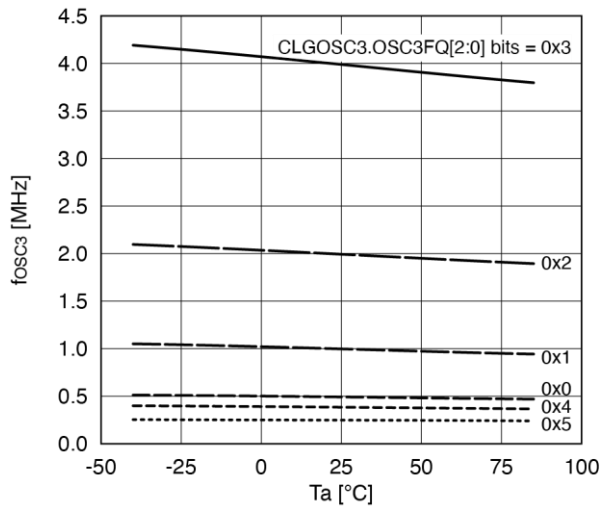
OSC3 oscillator circuit characteristics

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = 25^\circ\text{C}$

Item	Symbol	Condition	V_{DD}	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta3}			–	–	3	μs
Oscillation frequency	f_{OSC3}	CLGOSC3.OSC3FQ[2:0] bits = 0x3	1.6 to 3.6 V	3.80	4.00	4.20	MHz
		CLGOSC3.OSC3FQ[2:0] bits = 0x2	1.6 to 3.6 V	1.90	2.00	2.10	MHz
		CLGOSC3.OSC3FQ[2:0] bits = 0x1	1.6 to 3.6 V	0.95	1.00	1.05	MHz
			1.2 to 1.6 V	–	1.00	–	MHz
		CLGOSC3.OSC3FQ[2:0] bits = 0x0	1.6 to 3.6 V	0.475	0.50	0.525	MHz
			1.2 to 1.6 V	–	0.50	–	MHz
		CLGOSC3.OSC3FQ[2:0] bits = 0x4	1.6 to 3.6 V	0.365	0.384	0.403	MHz
			1.2 to 1.6 V	–	0.384	–	MHz
CLGOSC3.OSC3FQ[2:0] bits = 0x5	1.6 to 3.6 V	0.238	0.25	0.263	MHz		
	1.2 to 1.6 V	–	0.25	–	MHz		

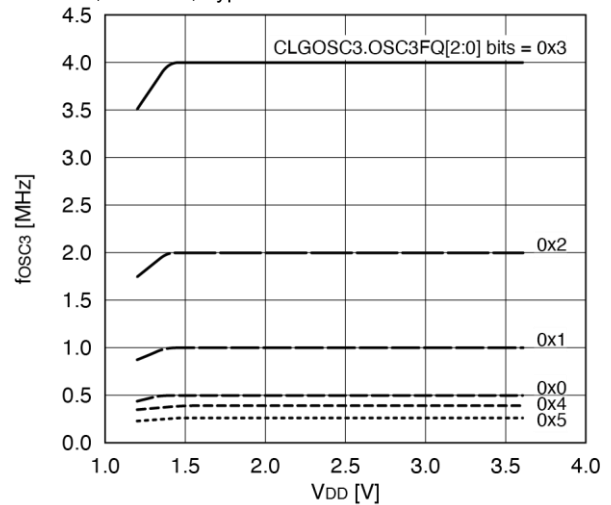
OSC3 internal oscillation frequency-temperature characteristic

$V_{DD} = 1.6$ to 3.6 V, $VD1=1.4$, Typ. value



OSC3 internal oscillation frequency-power supply voltage characteristic

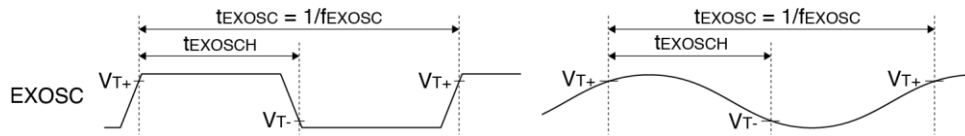
$T_a = 25^\circ\text{C}$, $VD1=1.4$, Typ. value



EXOSC external clock input characteristics

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EXOSC external clock duty ratio	t_{EXOSCD}	$t_{EXOSCD} = t_{EXOSCH}/f_{EXOSC}$	46	—	54	%
High level Schmitt input threshold voltage	V_{T+}		$0.5 \times V_{DD}$	—	$0.8 \times V_{DD}$	V
Low level Schmitt input threshold voltage	V_{T-}		$0.2 \times V_{DD}$	—	$0.5 \times V_{DD}$	V
Schmitt input hysteresis voltage	ΔV_T		120	—	—	mV



20.6 Flash Memory Characteristics

Unless otherwise specified: $V_{DD} = 2.2$ to 3.6 V, $V_{SS} = 0$ V^{*1}, $T_a = -40$ to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Programming count ^{*2}	C_{FEP}	Programmed data is guaranteed to be retained for 10 years.	1,000	—	—	times

*1 The potential variation of the V_{SS} voltage should be suppressed to within ± 0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count).

*2 Assumed that Erasing + Programming as count of 1. The count includes programming in the factory for shipment with ROM data programmed.

20.7 EEPROM Characteristics

Unless otherwise specified: $V_{DD} = 2.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Programming count	C_{EEP}	Programmed data is guaranteed to be retained for 20 years.	100,000	—	—	times
Programming time	t_{PRG}		—	0.2	15	ms
Programming current ^{*1}	I_{EEPROM}		—	3.2	10	mA
Programming power supply start-up time	t_{CPST}		^{*2}	—	—	ms
Effective EEPROM reset pulse width	t_{XPOR}		500	—	—	ns

*1 The value is added to the current consumption in RUN mode.

*2 Determine the value referencing the equation below.

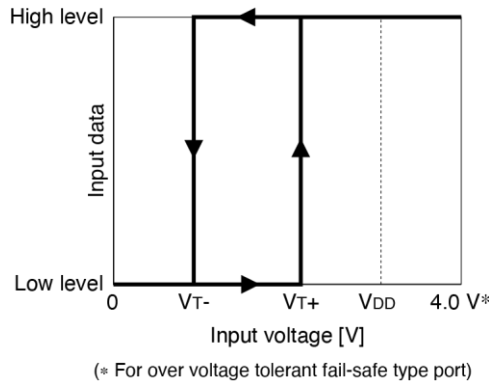
$$t_{CPST} = 37,500 \times C_{VPP} + 15$$

t_{CPST} : Programming power supply start-up time [μs], C_{VPP} : External smoothing capacitance [μF]

20.8 Input/Output Port (PPORT) Characteristics

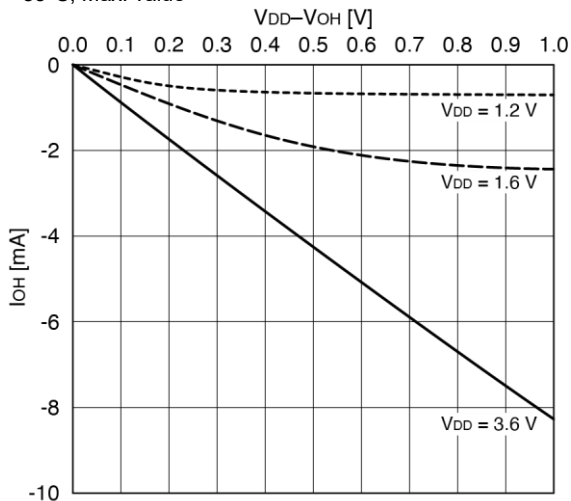
Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit		
High level Schmitt input threshold voltage	V_{T+}	P00-07, P10-17, P20-25, PD0-D1	$0.5 \times V_{DD}$	–	$0.8 \times V_{DD}$	V		
Low level Schmitt input threshold voltage	V_{T-}	P00-07, P10-17, P20-25, PD0-D1	$0.2 \times V_{DD}$	–	$0.5 \times V_{DD}$	V		
Schmitt input hysteresis voltage	ΔV_T	P00-07, P10-17, P20-25, PD0-D1	120	–	–	mV		
High level output current	I_{OH}	P00-07, P10-17, P20-25, PD0-D2, $V_{OH} = 0.9 \times V_{DD}$	1.2 to 1.6V	–	–	-0.2	mA	
			1.6 to 3.6V	–	–	-0.5	mA	
Low level output current	I_{OL}	P00-01, P04-07, P10-17, P20-25, PD0-D2, $V_{OL} = 0.1 \times V_{DD}$	1.2 to 1.6V	0.2	–	–	mA	
			1.6 to 3.6V	0.5	–	–	mA	
			P02-P03, $V_{OL} = 0.1 \times V_{DD}$	1.2 to 1.6V	4	–	–	mA
				1.6 to 2.5V	7	–	–	mA
2.5 to 3.6V	10	–	–	–	mA			
Leakage current	I_{LEAK}	P00-07, P10-17, P20-25, PD0-D2	-150	–	150	nA		
Input pull-up resistance	R_{INU}	P00-01, P04-07, P10-17, P20-25, PD0-D1	100	200	500	k Ω		
Input pull-down resistance	R_{IND}	P00-01, P04-07, P10-17, P20-25, PD0-D1	100	200	500	k Ω		
Pin capacitance	C_{IN}	P00-07, P10-17, P20-25, PD0-D1	–	–	15	pF		



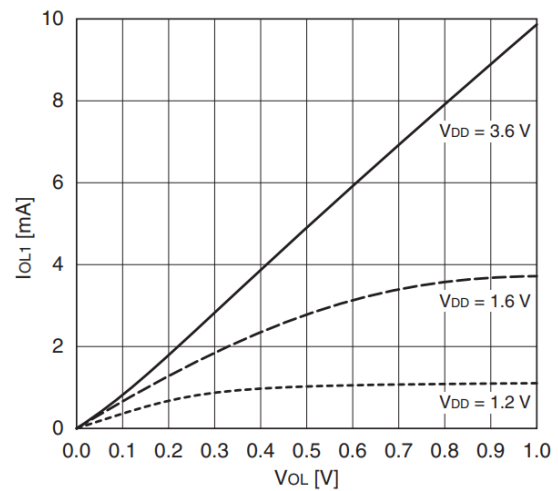
High-level output current characteristic

$T_a = 85^\circ\text{C}$, Max. value



Low-level output current characteristic

$T_a = 85^\circ\text{C}$, Min. value



20.9 Supply Voltage Detector (SVD5) Characteristics

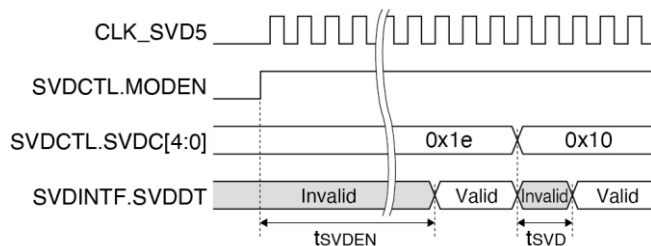
Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	V_{DD}	Min.	Typ.	Max.	Unit
EXSVD pin input voltage range	V_{EXSVD}			0	–	5.5	V
EXSVD input impedance	R_{EXSVD}	SVDCTL.SVDC[4:0] bits = 0x00		366	407	448	k Ω
		SVDCTL.SVDC[4:0] bits = 0x01		388	431	474	k Ω
		SVDCTL.SVDC[4:0] bits = 0x02		409	455	500	k Ω
		SVDCTL.SVDC[4:0] bits = 0x03		431	479	527	k Ω
		SVDCTL.SVDC[4:0] bits = 0x04		452	503	553	k Ω
		SVDCTL.SVDC[4:0] bits = 0x05		474	527	579	k Ω
		SVDCTL.SVDC[4:0] bits = 0x06		495	550	606	k Ω
		SVDCTL.SVDC[4:0] bits = 0x07		517	574	632	k Ω
		SVDCTL.SVDC[4:0] bits = 0x08		539	598	658	k Ω
		SVDCTL.SVDC[4:0] bits = 0x09		560	622	685	k Ω
		SVDCTL.SVDC[4:0] bits = 0x0a		582	646	711	k Ω
		SVDCTL.SVDC[4:0] bits = 0x0b		603	670	737	k Ω
		SVDCTL.SVDC[4:0] bits = 0x0c		625	694	763	k Ω
		SVDCTL.SVDC[4:0] bits = 0x0d		646	718	790	k Ω
		SVDCTL.SVDC[4:0] bits = 0x0e		668	742	816	k Ω
		SVDCTL.SVDC[4:0] bits = 0x0f		689	766	842	k Ω
		SVDCTL.SVDC[4:0] bits = 0x10		711	790	869	k Ω
		SVDCTL.SVDC[4:0] bits = 0x11		754	838	921	k Ω
		SVDCTL.SVDC[4:0] bits = 0x12		775	862	948	k Ω
		SVDCTL.SVDC[4:0] bits = 0x13		797	886	974	k Ω
		SVDCTL.SVDC[4:0] bits = 0x14		819	909	1,000	k Ω
		SVDCTL.SVDC[4:0] bits = 0x15		840	933	1,027	k Ω
		SVDCTL.SVDC[4:0] bits = 0x16		862	957	1,053	k Ω
		SVDCTL.SVDC[4:0] bits = 0x17		883	981	1,079	k Ω
		SVDCTL.SVDC[4:0] bits = 0x18		905	1,005	1,106	k Ω
		SVDCTL.SVDC[4:0] bits = 0x19		926	1,029	1,132	k Ω
		SVDCTL.SVDC[4:0] bits = 0x1a		969	1,077	1,185	k Ω
		SVDCTL.SVDC[4:0] bits = 0x1b		991	1,101	1,211	k Ω
		SVDCTL.SVDC[4:0] bits = 0x1c		1,012	1,125	1,237	k Ω
		SVDCTL.SVDC[4:0] bits = 0x1d		1,034	1,149	1,264	k Ω
		SVDCTL.SVDC[4:0] bits = 0x1e		1,055	1,173	1,290	k Ω
		SVDCTL.SVDC[4:0] bits = 0x1f		1,077	1,197	1,316	k Ω
EXSVD detection voltage	V_{SVD_EXT}	SVDCTL.SVDC[4:0] bits = 0x00		1.121	1.15	1.179	V
		SVDCTL.SVDC[4:0] bits = 0x01		1.170	1.20	1.230	V
		SVDCTL.SVDC[4:0] bits = 0x02		1.219	1.25	1.281	V
		SVDCTL.SVDC[4:0] bits = 0x03		1.268	1.30	1.333	V
		SVDCTL.SVDC[4:0] bits = 0x04		1.316	1.35	1.384	V
		SVDCTL.SVDC[4:0] bits = 0x05		1.365	1.40	1.435	V
		SVDCTL.SVDC[4:0] bits = 0x06		1.414	1.45	1.486	V
		SVDCTL.SVDC[4:0] bits = 0x07		1.463	1.50	1.538	V
		SVDCTL.SVDC[4:0] bits = 0x08		1.511	1.55	1.589	V
		SVDCTL.SVDC[4:0] bits = 0x09		1.560	1.60	1.640	V
		SVDCTL.SVDC[4:0] bits = 0x0a		1.609	1.65	1.691	V
		SVDCTL.SVDC[4:0] bits = 0x0b		1.658	1.70	1.743	V
		SVDCTL.SVDC[4:0] bits = 0x0c		1.755	1.80	1.845	V
		SVDCTL.SVDC[4:0] bits = 0x0d		1.853	1.90	1.948	V
		SVDCTL.SVDC[4:0] bits = 0x0e		1.950	2.00	2.050	V
		SVDCTL.SVDC[4:0] bits = 0x0f		2.048	2.10	2.153	V
		SVDCTL.SVDC[4:0] bits = 0x10		2.145	2.20	2.255	V
		SVDCTL.SVDC[4:0] bits = 0x11		2.194	2.25	2.306	V
		SVDCTL.SVDC[4:0] bits = 0x12		2.243	2.30	2.358	V
		SVDCTL.SVDC[4:0] bits = 0x13		2.291	2.35	2.409	V
		SVDCTL.SVDC[4:0] bits = 0x14		2.340	2.40	2.460	V
		SVDCTL.SVDC[4:0] bits = 0x15		2.389	2.45	2.511	V
		SVDCTL.SVDC[4:0] bits = 0x16		2.438	2.50	2.563	V
		SVDCTL.SVDC[4:0] bits = 0x17		2.486	2.55	2.614	V
		SVDCTL.SVDC[4:0] bits = 0x18		2.535	2.60	2.665	V
		SVDCTL.SVDC[4:0] bits = 0x19		2.584	2.65	2.716	V
		SVDCTL.SVDC[4:0] bits = 0x1a		2.633	2.70	2.768	V
		SVDCTL.SVDC[4:0] bits = 0x1b		2.730	2.80	2.870	V

20 Electrical Characteristics

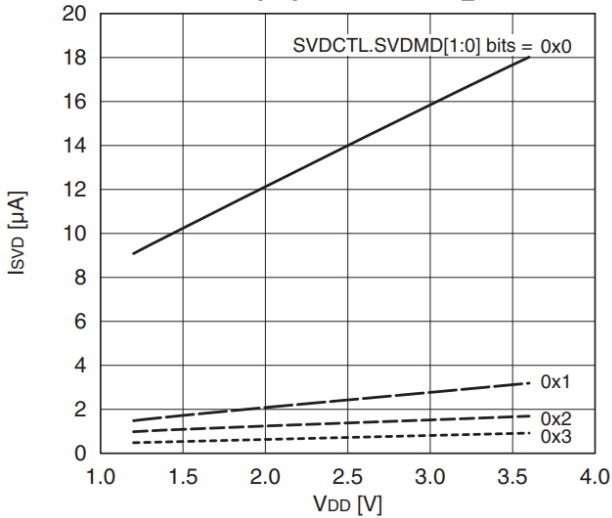
Item	Symbol	Condition	V _{DD}	Min.	Typ.	Max.	Unit
EXSVD input impedance	R _{EXSVD}	SVDCTL.SVDC[4:0] bits = 0x1c		2.828	2.90	2.973	V
		SVDCTL.SVDC[4:0] bits = 0x1d		2.925	3.00	3.075	V
		SVDCTL.SVDC[4:0] bits = 0x1e		3.023	3.10	3.178	V
		SVDCTL.SVDC[4:0] bits = 0x1f		3.120	3.20	3.280	V
SVD detection voltage	V _{SVD}	SVDCTL.SVDC[4:0] bits = 0x00	1.1 to 3.6V	1.121	1.15	1.179	V
		SVDCTL.SVDC[4:0] bits = 0x01		1.170	1.20	1.230	V
		SVDCTL.SVDC[4:0] bits = 0x02		1.219	1.25	1.281	V
		SVDCTL.SVDC[4:0] bits = 0x03		1.268	1.30	1.333	V
		SVDCTL.SVDC[4:0] bits = 0x04		1.316	1.35	1.384	V
		SVDCTL.SVDC[4:0] bits = 0x05		1.365	1.40	1.435	V
		SVDCTL.SVDC[4:0] bits = 0x06		1.414	1.45	1.486	V
		SVDCTL.SVDC[4:0] bits = 0x07		1.463	1.50	1.538	V
		SVDCTL.SVDC[4:0] bits = 0x08		1.511	1.55	1.589	V
		SVDCTL.SVDC[4:0] bits = 0x09		1.560	1.60	1.640	V
		SVDCTL.SVDC[4:0] bits = 0x0a		1.609	1.65	1.691	V
		SVDCTL.SVDC[4:0] bits = 0x0b		1.658	1.70	1.743	V
		SVDCTL.SVDC[4:0] bits = 0x0c		1.755	1.80	1.845	V
		SVDCTL.SVDC[4:0] bits = 0x0d		1.853	1.90	1.948	V
		SVDCTL.SVDC[4:0] bits = 0x0e		1.950	2.00	2.050	V
		SVDCTL.SVDC[4:0] bits = 0x0f		2.048	2.10	2.153	V
		SVDCTL.SVDC[4:0] bits = 0x10		2.145	2.20	2.255	V
		SVDCTL.SVDC[4:0] bits = 0x11		2.194	2.25	2.306	V
		SVDCTL.SVDC[4:0] bits = 0x12		2.243	2.30	2.358	V
		SVDCTL.SVDC[4:0] bits = 0x13		2.291	2.35	2.409	V
		SVDCTL.SVDC[4:0] bits = 0x14		2.340	2.40	2.460	V
		SVDCTL.SVDC[4:0] bits = 0x15		2.389	2.45	2.511	V
		SVDCTL.SVDC[4:0] bits = 0x16		2.438	2.50	2.563	V
		SVDCTL.SVDC[4:0] bits = 0x17		2.486	2.55	2.614	V
		SVDCTL.SVDC[4:0] bits = 0x18		2.535	2.60	2.665	V
		SVDCTL.SVDC[4:0] bits = 0x19		2.584	2.65	2.716	V
		SVDCTL.SVDC[4:0] bits = 0x1a		2.633	2.70	2.768	V
		SVDCTL.SVDC[4:0] bits = 0x1b		2.730	2.80	2.870	V
SVDCTL.SVDC[4:0] bits = 0x1c		2.828	2.90	2.973	V		
SVDCTL.SVDC[4:0] bits = 0x1d		2.925	3.00	3.075	V		
SVDCTL.SVDC[4:0] bits = 0x1e		3.023	3.10	3.178	V		
SVDCTL.SVDC[4:0] bits = 0x1f		3.120	3.20	3.280	V		
SVD circuit enable response time	t _{SVDEN}	*1		–	–	500	μs
SVD circuit response time	t _{SVD}			–	–	60	μs
SVD circuit current	I _{SVD}	SVDCTL.SVDMD[1:0] bits = 0x0, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD5 = 32 kHz, Ta = 25°C		–	18	31	μA
		SVDCTL.SVDMD[1:0] bits = 0x1, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD5 = 32 kHz, Ta = 25°C		–	3.2	5.3	μA
		SVDCTL.SVDMD[1:0] bits = 0x2, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD5 = 32 kHz, Ta = 25°C		–	1.7	2.8	μA
		SVDCTL.SVDMD[1:0] bits = 0x3, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD5 = 32 kHz, Ta = 25°C		–	0.9	1.5	μA

*1 If CLK_SVD5 is configured in the neighborhood of 32 kHz, the SVDINTF.SVDDT bit is masked during the t_{SVDEN} period and it retains the previous value.



SVD circuit current - power supply voltage characteristic

Ta = 25°C, SVDCTL.SVDC[4:0] bits = 0x00, CLK_SVD5 = 32 kHz, Typ. value



20.10 UART (UART3) Characteristics

Unless otherwise specified: V_{DD} = 1.2 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	V _{DD}	Min.	Typ.	Max.	Unit
Transfer baud rate	U _{BRT1}	Normal mode	1.6 to 3.6 V	150	–	460,800	bps
			1.2 to 1.6 V	150	–	57,600	bps
	U _{BRT2}	IrDA mode	1.6 to 3.6 V	150	–	115,200	bps
			1.2 to 1.6 V	150	–	57,600	bps

20.11 Synchronous Serial Interface (SPIA) Characteristics

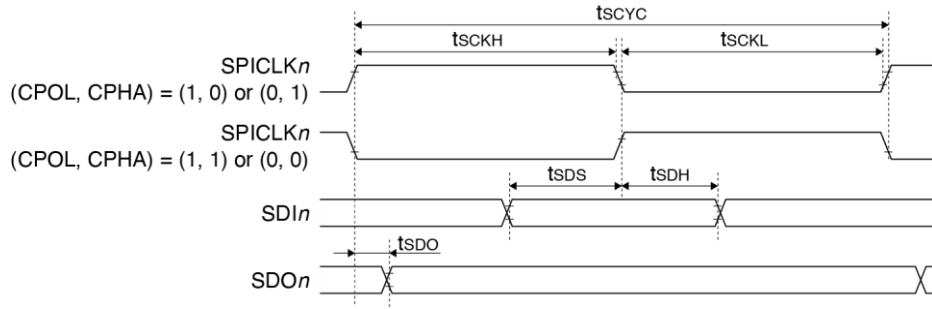
Unless otherwise specified: V_{DD} = 1.2 to 3.6 V, V_{SS} = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	V _{DD}	V _{D1}	Min.	Typ.	Max.	Unit
SPICLK _n cycle time	t _{SCYC}		1.6 to 3.6 V	1.4V	500	–	–	ns
			1.2 to 1.6 V	–	1,000	–	–	ns
SPICLK _n High pulse width	t _{SCKH}		1.6 to 3.6 V	1.4V	200	–	–	ns
			1.2 to 1.6 V	–	400	–	–	ns
SPICLK _n Low pulse width	t _{SCKL}		1.6 to 3.6 V	1.4V	200	–	–	ns
			1.2 to 1.6 V	–	400	–	–	ns
SDIn setup time	t _{SDS}		1.6 to 3.6 V	1.4V	125	–	–	ns
			1.2 to 1.6 V	–	250	–	–	ns
SDIn hold time	t _{SDH}		1.6 to 3.6 V	1.4V	70	–	–	ns
			1.2 to 1.6 V	–	140	–	–	ns
SDOn output delay time	t _{SDO}	C _L = 30 pF *1	1.6 to 3.6 V	1.4V	–	–	120	ns
			1.2 to 1.6 V	–	–	–	320	ns
#SPISS _n setup time	t _{SSS}		1.6 to 3.6 V	1.4V	125	–	–	ns
			1.2 to 1.6 V	–	250	–	–	ns
#SPISS _n High pulse width	t _{SSH}		1.6 to 3.6 V	1.4V	80	–	–	ns
			1.2 to 1.6 V	–	160	–	–	ns
SDOn output start time	t _{SDD}	C _L = 30 pF *1	1.6 to 3.6 V	1.4V	–	–	120	ns
			1.2 to 1.6 V	–	–	–	320	ns
SDOn output stop time	t _{SDZ}	C _L = 30 pF *1	1.6 to 3.6 V	1.4V	–	–	80	ns
			1.2 to 1.6 V	–	–	–	160	ns

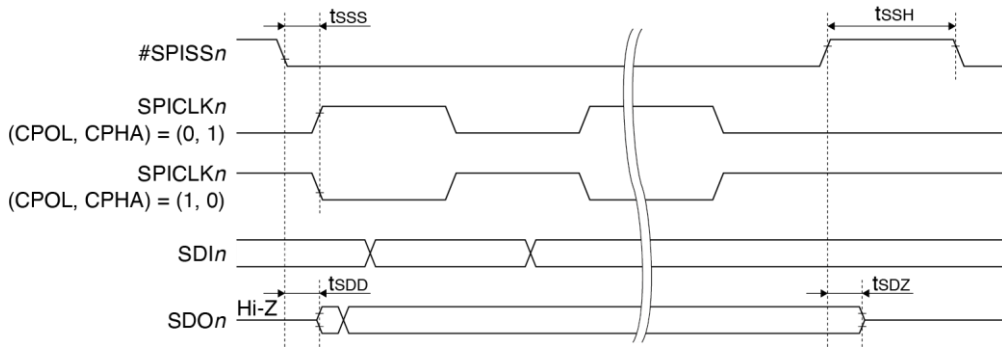
*1 C_L = Pin load

20 Electrical Characteristics

Master and slave modes



Slave mode

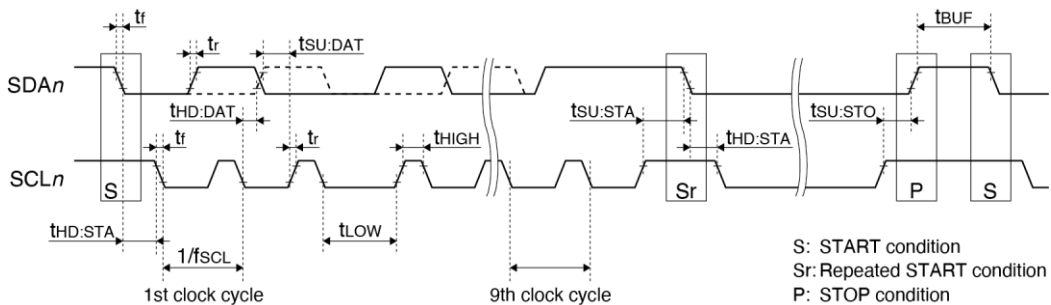


20.12 I²C (I2C) Characteristics

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	Standard mode $V_{DD} = 1.2$ to 3.6 V			Fast mode $V_{DD} = 1.6$ to 3.6 V $V_{D1} = 1.4$ V			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
SCLn frequency	f_{SCL}		0	–	100	0	–	400	kHz
Hold time (repeated) START condition *	$t_{HD:STA}$		4.0	–	–	0.6	–	–	μs
SCLn Low pulse width	t_{LOW}		4.7	–	–	1.3	–	–	μs
SCLn High pulse width	t_{HIGH}		4.0	–	–	0.6	–	–	μs
Repeated START condition setup time	$t_{SU:STA}$		4.7	–	–	0.6	–	–	μs
Data hold time	$t_{HD:DAT}$		0	–	–	0	–	–	μs
Data setup time	$t_{SU:DAT}$		250	–	–	100	–	–	ns
SDAn, SCLn rise time	t_r		–	–	1,000	–	–	300	ns
SDAn, SCLn fall time	t_f		–	–	300	–	–	300	ns
STOP condition setup time	$t_{SU:STO}$		4.0	–	–	0.6	–	–	μs
Bus free time	t_{BUF}		4.7	–	–	1.3	–	–	μs

* After this period, the first clock pulse is generated.



S: START condition
Sr: Repeated START condition
P: STOP condition

20.13 LCD Driver (LCD4B) Characteristics

The LCD driver characteristics varies depending on the panel load (panel size, drive duty, number of display pixels and display contents), so evaluate them by connecting to the actually used LCD panel.

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = 25^\circ\text{C}$, LCD4TIM2.BSTC[1:0] bits = 0x1 (Voltage booster clock = 2 kHz), No panel load

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	
LCD drive voltage (V_{C2} reference voltage) $V_{DD} = 2.7$ to 3.6 V ³ LCD4PWR.VCSEL bit = 1	V_{C1}	Connect 1 M Ω load resistor between V_{SS} and V_{C1}	$0.323 \times V_{C3}$ (Typ.)	–	$0.344 \times V_{C3}$ (Typ.)	V	
	V_{C2}	Connect 1 M Ω load resistor between V_{SS} and V_{C2}	$0.646 \times V_{C3}$ (Typ.)	–	$0.687 \times V_{C3}$ (Typ.)	V	
	V_{C3}	Connect 1 M Ω load resistor between V_{SS} and V_{C3}	LCD4PWR.LC[4:0] bits = 0x00	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x01	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x02	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x03	2.61	2.69	2.77	V
			LCD4PWR.LC[4:0] bits = 0x04	2.68	2.76	2.84	V
			LCD4PWR.LC[4:0] bits = 0x05	2.75	2.84	2.92	V
			LCD4PWR.LC[4:0] bits = 0x06	2.82	2.91	3.00	V
			LCD4PWR.LC[4:0] bits = 0x07	2.90	2.98	3.07	V
			LCD4PWR.LC[4:0] bits = 0x08	2.97	3.06	3.15	V
			LCD4PWR.LC[4:0] bits = 0x09	3.04	3.13	3.23	V
			LCD4PWR.LC[4:0] bits = 0x0a	3.11	3.21	3.30	V
			LCD4PWR.LC[4:0] bits = 0x0b	3.26	3.36	3.46	V
			LCD4PWR.LC[4:0] bits = 0x0c	3.40	3.51	3.61	V
			LCD4PWR.LC[4:0] bits = 0x0d	3.55	3.66	3.77	V
			LCD4PWR.LC[4:0] bits = 0x0e	3.69	3.81	3.92	V
			LCD4PWR.LC[4:0] bits = 0x0f	3.84	3.95	4.07	V
			LCD4PWR.LC[4:0] bits = 0x10	3.98	4.10	4.23	V
			LCD4PWR.LC[4:0] bits = 0x11	4.13	4.25	4.38	V
			LCD4PWR.LC[4:0] bits = 0x12	4.27	4.40	4.53	V
			LCD4PWR.LC[4:0] bits = 0x13	4.41	4.55	4.69	V
	LCD4PWR.LC[4:0] bits = 0x14	4.56	4.70	4.84	V		
	LCD4PWR.LC[4:0] bits = 0x15	4.70	4.85	5.00	V		
	LCD4PWR.LC[4:0] bits = 0x16	4.85	5.00	5.15	V		
	LCD4PWR.LC[4:0] bits = 0x17	4.92	5.07	5.23	V		
	LCD4PWR.LC[4:0] bits = 0x18	4.99	5.15	5.30	V		
LCD4PWR.LC[4:0] bits = 0x19	5.07	5.22	5.38	V			
LCD4PWR.LC[4:0] bits = 0x1a	5.14	5.30	5.46	V			
LCD4PWR.LC[4:0] bits = 0x1b	5.21	5.37	5.53	V			
LCD4PWR.LC[4:0] bits = 0x1c	5.28	5.45	5.61	V			
LCD4PWR.LC[4:0] bits = 0x1d	5.36	5.52	5.69	V			
LCD4PWR.LC[4:0] bits = 0x1e	5.43	5.60	5.76	V			
LCD4PWR.LC[4:0] bits = 0x1f	5.50	5.67	5.84	V			
LCD drive voltage (V_{C1} reference voltage) $V_{DD} = 1.2$ to 3.6 V ³ LCD4PWR.VCSEL bit = 0	V_{C1}	Connect 1 M Ω load resistor between V_{SS} and V_{C1}	$0.323 \times V_{C3}$ (Typ.)	–	$0.344 \times V_{C3}$ (Typ.)	V	
	V_{C2}	Connect 1 M Ω load resistor between V_{SS} and V_{C2}	$0.646 \times V_{C3}$ (Typ.)	–	$0.687 \times V_{C3}$ (Typ.)	V	
	V_{C3}	Connect 1 M Ω load resistor between V_{SS} and V_{C3}	LCD4PWR.LC[4:0] bits = 0x00	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x01	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x02	–	–	–	V
			LCD4PWR.LC[4:0] bits = 0x03	2.61	2.69	2.77	V
			LCD4PWR.LC[4:0] bits = 0x04	2.68	2.76	2.84	V
			LCD4PWR.LC[4:0] bits = 0x05	2.75	2.84	2.92	V
			LCD4PWR.LC[4:0] bits = 0x06	2.82	2.91	3.00	V
			LCD4PWR.LC[4:0] bits = 0x07	2.90	2.98	3.07	V
			LCD4PWR.LC[4:0] bits = 0x08	2.97	3.06	3.15	V
			LCD4PWR.LC[4:0] bits = 0x09	3.04	3.13	3.23	V
			LCD4PWR.LC[4:0] bits = 0x0a	3.11	3.21	3.30	V
			LCD4PWR.LC[4:0] bits = 0x0b	3.26	3.36	3.46	V
			LCD4PWR.LC[4:0] bits = 0x0c	3.40	3.51	3.61	V
			LCD4PWR.LC[4:0] bits = 0x0d	3.55	3.66	3.77	V
			LCD4PWR.LC[4:0] bits = 0x0e	3.69	3.81	3.92	V
			LCD4PWR.LC[4:0] bits = 0x0f	3.84	3.95	4.07	V

20 Electrical Characteristics

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	
LCD drive voltage (V _{C1} reference voltage) LCD4PWR.VCSEL bit = 0	V _{C3}	Connect 1 MΩ load resistor between V _{SS} and V _{C3}	LCD4PWR.LC[4:0] bits = 0x10	3.98	4.10	4.23	V
			LCD4PWR.LC[4:0] bits = 0x11	4.13	4.25	4.38	V
			LCD4PWR.LC[4:0] bits = 0x12	4.27	4.40	4.53	V
			LCD4PWR.LC[4:0] bits = 0x13	4.41	4.55	4.69	V
			LCD4PWR.LC[4:0] bits = 0x14	4.56	4.70	4.84	V
			LCD4PWR.LC[4:0] bits = 0x15	4.70	4.85	5.00	V
			LCD4PWR.LC[4:0] bits = 0x16	4.85	5.00	5.15	V
			LCD4PWR.LC[4:0] bits = 0x17	4.92	5.07	5.23	V
			LCD4PWR.LC[4:0] bits = 0x18	4.99	5.15	5.30	V
			LCD4PWR.LC[4:0] bits = 0x19	5.07	5.22	5.38	V
			LCD4PWR.LC[4:0] bits = 0x1a	5.14	5.30	5.46	V
			LCD4PWR.LC[4:0] bits = 0x1b	5.21	5.37	5.53	V
			LCD4PWR.LC[4:0] bits = 0x1c	5.28	5.45	5.61	V
			LCD4PWR.LC[4:0] bits = 0x1d	5.36	5.52	5.69	V
			LCD4PWR.LC[4:0] bits = 0x1e	5.43	5.60	5.76	V
LCD4PWR.LC[4:0] bits = 0x1f	5.50	5.67	5.84	V			
LCD built-in dividing resistor value	R ₁ ,	LCD4PWR.RESISEL[1:0] bits = 0x1	–	50	–	kΩ	
	R ₂ ,	LCD4PWR.RESISEL[1:0] bits = 0x2	–	150	–	kΩ	
	R ₃	LCD4PWR.RESISEL[1:0] bits = 0x3	–	300	–	kΩ	
Segment/Common output current	I _{SEGH}	SEGxx, COMy VSEGH = V _{C3} /V _{C2} /V _{C1} - 0.1 V, Ta = -40 to 85°C	–	–	-10	μA	
	I _{SEGL}	SEGxx, COMy VSEGL = V _{SS} /V _{C2} /V _{C1} + 0.1 V, Ta = -40 to 85°C	10	–	–	μA	
LCD circuit current (V _{C2} reference voltage)	I _{LCD2}	LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.VCSEL bit = 1 *1 *2	–	2.2	5.3	μA	
		LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.VCSEL bit = 1 *1 *2	–	1.0	2.3	μA	
LCD circuit current (V _{C1} reference voltage)	I _{LCD1}	LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.VCSEL bit = 0 *1 *2	–	4.2	10	μA	
		LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.VCSEL bit = 0 *1 *2	–	1.8	4.1	μA	
LCD circuit current in heavy load protection mode (V _{C2} reference voltage)	I _{LCD2H}	LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.VCSEL bit = 1, LCD4PWR.HVLD bit = 1 *1 *2	–	16	33	μA	
LCD circuit current in heavy load protection mode (V _{C1} reference voltage)	I _{LCD1H}	LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.VCSEL bit = 0, LCD4PWR.HVLD bit = 1 *1 *2	–	9	19	μA	
LCD circuit current (when LCD built-in voltage dividing resistors are used)	I _{LCDR1}	LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.RESISEL[1:0] bits = 0x1 *2	–	23	48	μA	
		LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.RESISEL[1:0] bits = 0x1 *2	–	22	46	μA	
		LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.RESISEL[1:0] bits = 0x2 *2	–	9	20	μA	
	I _{LCDR2}	LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.RESISEL[1:0] bits = 0x2 *2	–	8	18	μA	
		LCD4DSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD4PWR.RESISEL[1:0] bits = 0x3 *2	–	5	12	μA	
	I _{LCDR3}	LCD4DSP.DSPC[1:0] bits = 0x2 (all on), LCD4PWR.RESISEL[1:0] bits = 0x3 *2	–	4	10	μA	

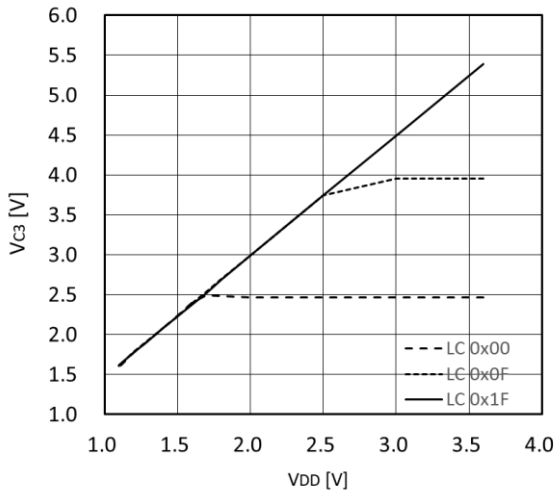
*1 Other LCD driver settings: LCD4PWR.LC[4:0] bits = 0x1f, CLK_LCD4B = 32 kHz, LCD4TIM1.FRMCNT[4:0] bits = 0x03 (frame frequency = 64 Hz)

*2 The value is added to the current consumption in HALT/RUN mode. Current consumption increases according to the display contents and panel load.

*3 The LCD drive voltage is lower than the LCD4PWR.LC[4:0] bit settings when V_{DD} = 1.2 to 1.6V. See the LCD drive voltage- supply voltage characteristics graph shown below.

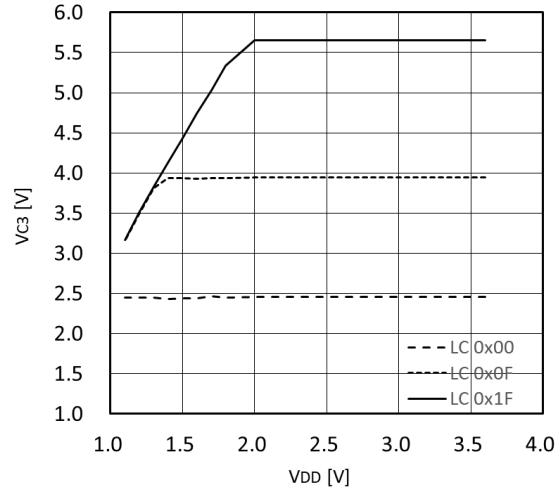
LCD drive voltage-supply voltage characteristic (V_{C2} reference voltage)

T_a = 25°C, Typ. value, when a 1 MΩ load resistor is connected between V_{SS} and V_{C3} (no panel load)



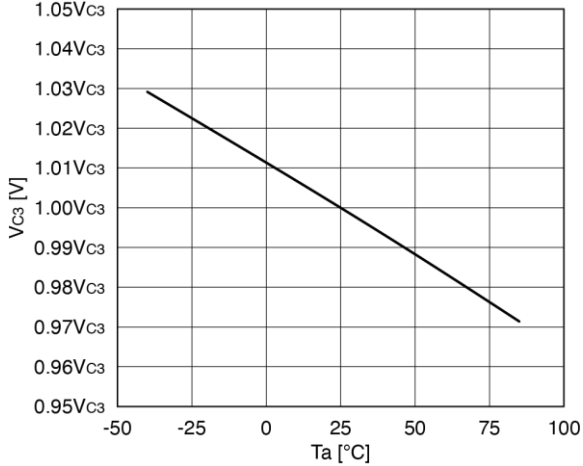
LCD drive voltage-supply voltage characteristic (V_{C1} reference voltage)

T_a = 25°C, Typ. value, when a 1 MΩ load resistor is connected between V_{SS} and V_{C3} (no panel load)



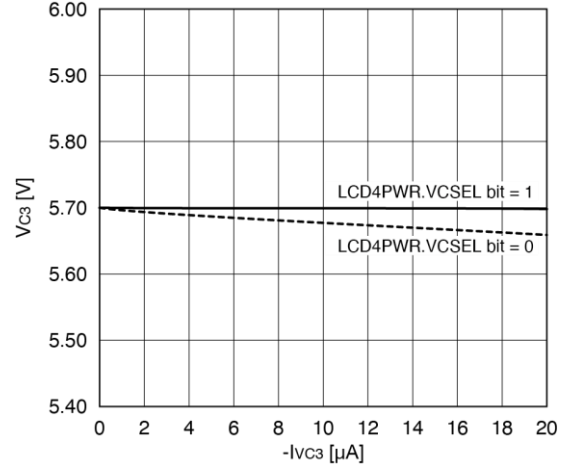
LCD drive voltage-temperature characteristic (V_{C1}/V_{C2} reference voltage)

Typ. value



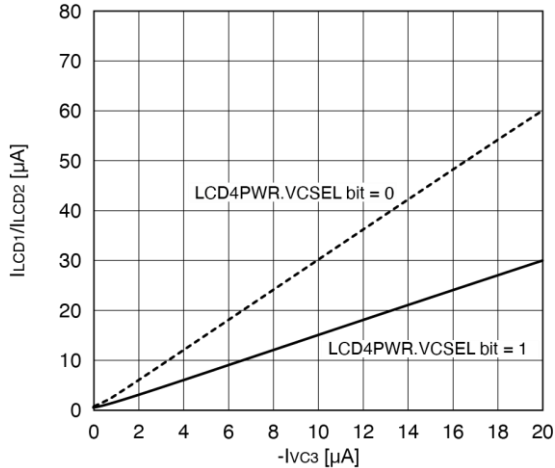
LCD drive voltage-load characteristic

V_{DD} = 3.6 V, T_a = 25°C, Typ. value, LCD4PWR.LC[4:0] bits = 0x1f, when a load is connected to the V_{C3} pin only



LCD circuit current-load characteristic

V_{DD} = 3.6 V, T_a = 25°C, Typ. value, LCD4PWR.LC[4:0] bits = 0x1f, when a load is connected to the V_{C3} pin only



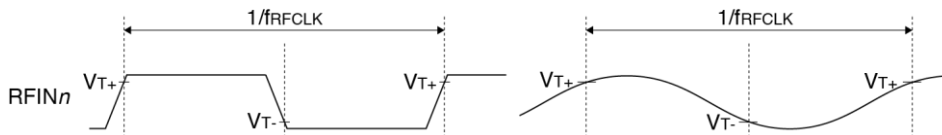
20.14 R/F Converter (RFC2) Characteristics

R/F converter characteristics change depending on conditions (board pattern, components used, etc.). Use these characteristic values as a reference and perform evaluation using the actual printed circuit board.

Unless otherwise specified: $V_{DD} = 1.2$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85 °C

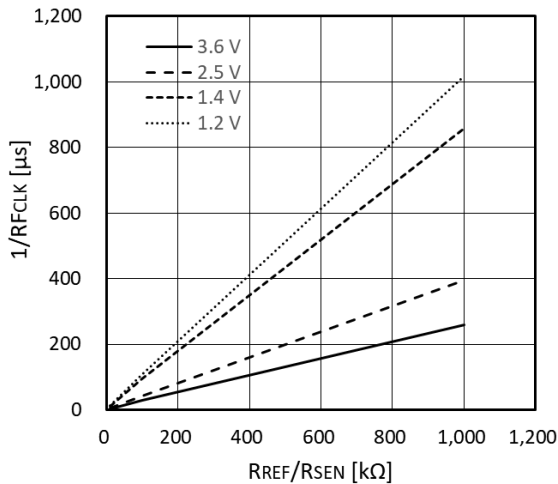
Item	Symbol	Condition	V_{DD}	Min.	Typ.	Max.	Unit
Reference/sensor oscillation frequency	f_{RFCLK}			1	–	1,000	kHz
Reference resistor/resistive sensor resistance	R_{REF}, R_{SEN}			10	–	–	k Ω
Reference capacitance	C_{REF}			100	–	–	pF
Time base counter clock frequency	f_{TCCLK}		1.6 to 3.6 V	–	–	4.2	MHz
			1.2 to 1.6 V	–	–	1.1	MHz
High level Schmitt input threshold voltage	V_{T+}			$0.5 \times V_{D1}$	–	$0.8 \times V_{D1}$	V
Low level Schmitt input threshold voltage	V_{T-}			$0.2 \times V_{D1}$	–	$0.5 \times V_{D1}$	V
Schmitt input hysteresis voltage	ΔV_T			120	–	–	mV
R/F converter operating current	I_{RFC}	$C_{REF} = 1,000$ pF, $R_{REF}/R_{SEN} = 100$ k Ω , $T_a = 25$ °C	3.6V	–	32	48	μ A
			2.4V	–	20	30	μ A
			2.0V	–	16	23	μ A

Waveforms for external clock input mode



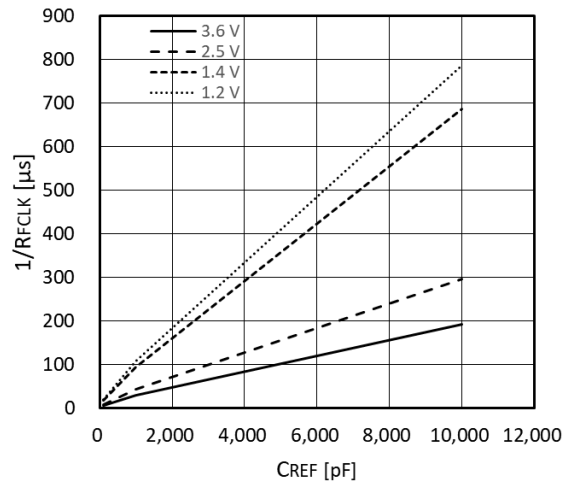
RFC reference/sensor oscillation period-resistance characteristic

$C_{REF} = 1,000$ pF, $T_a = 25$ °C, $V_{D1} = V_{CAPRF} = 1.4$ V, Typ. Value



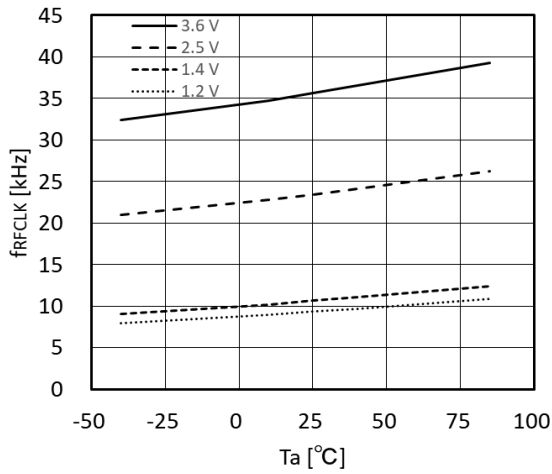
RFC reference/sensor oscillation period-capacitance characteristic

$R_{REF}/R_{SEN} = 100$ k Ω , $T_a = 25$ °C, Typ. Value



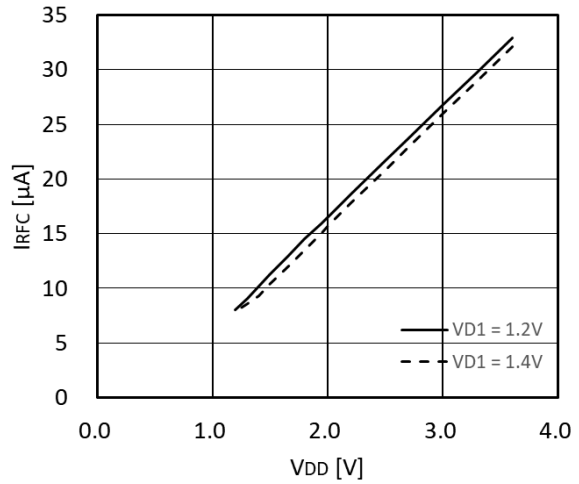
RFC reference/sensor oscillation frequency - temperature characteristic

$C_{REF} = 1,000 \text{ pF}$, $V_{D1} = V_{CAPRF} = 1.4\text{V}$, Typ. Value



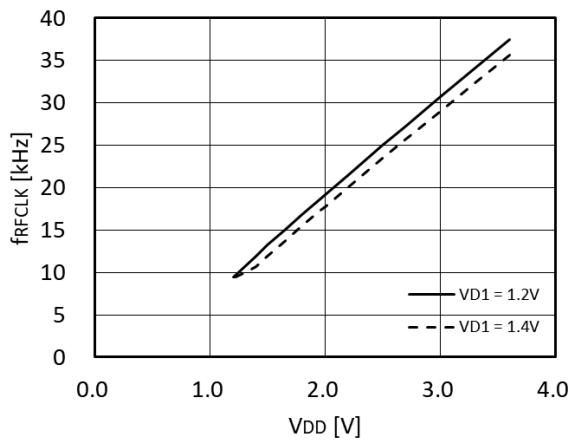
RFC reference/sensor oscillation current consumption-power supply voltage characteristic

$R_{REF}/R_{SEN} = 100 \text{ k}\Omega$, $C_{REF} = 1,000 \text{ pF}$, $T_a = 25 \text{ }^\circ\text{C}$, Typ. Value



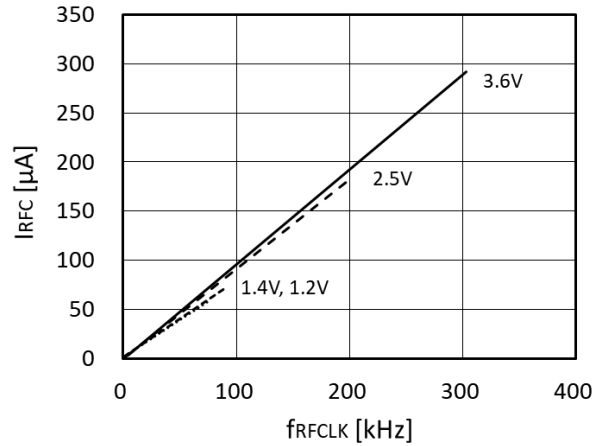
RFC reference/sensor oscillation frequency - power supply voltage characteristic

$R_{REF}/R_{SEN} = 100 \text{ k}\Omega$, $C_{REF} = 1,000 \text{ pF}$, $T_a = 25 \text{ }^\circ\text{C}$, Typ. Value



RFC reference/sensor oscillation current consumption-frequency characteristic

$C_{REF} = 1,000 \text{ pF}$, $T_a = 25 \text{ }^\circ\text{C}$, $V_{D1} = V_{CAPRF} = 1.4\text{V}$, Typ. Value



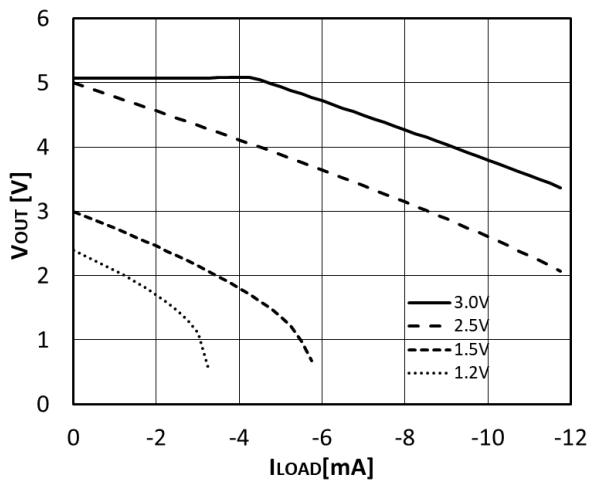
20.15 Charge Pump Regulator (CHREG) Characteristics

Unless otherwise specified: $V_{DD} = 2.5$ to 3.6 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

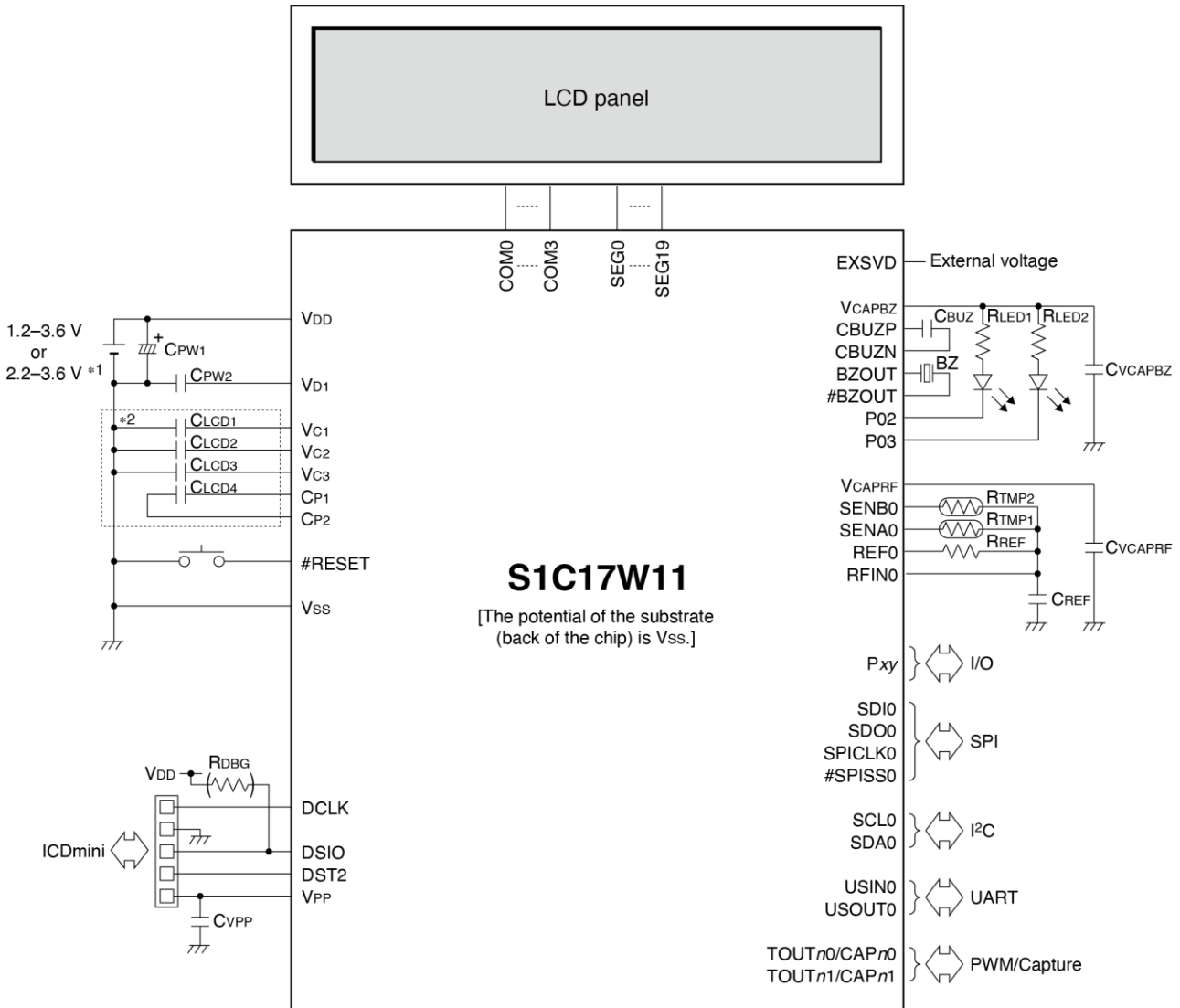
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Power supply voltage	V_{out}		3.8	5.0	5.5	V
Output current	I_{out}		10	–	–	mA
CHREG operating current	I_{op}	Clock frequency 100kHz, no load resistance	–	5	30	μA
Ripple voltage	V_{rip}	$T_a = 25^\circ\text{C}$	–	100	–	mV _{p-p}
Rise time	t_{ST}		–	–	5	ms

Load Current-Output Voltage Characteristics

$T_a = 25^\circ\text{C}$, Typ. Values



21 Basic External Connection Diagram



- *1: For Flash/EEPROM programming
- *2: When the internal LCD power supply is used
- () : Do not mount components if unnecessary.

Sample external components

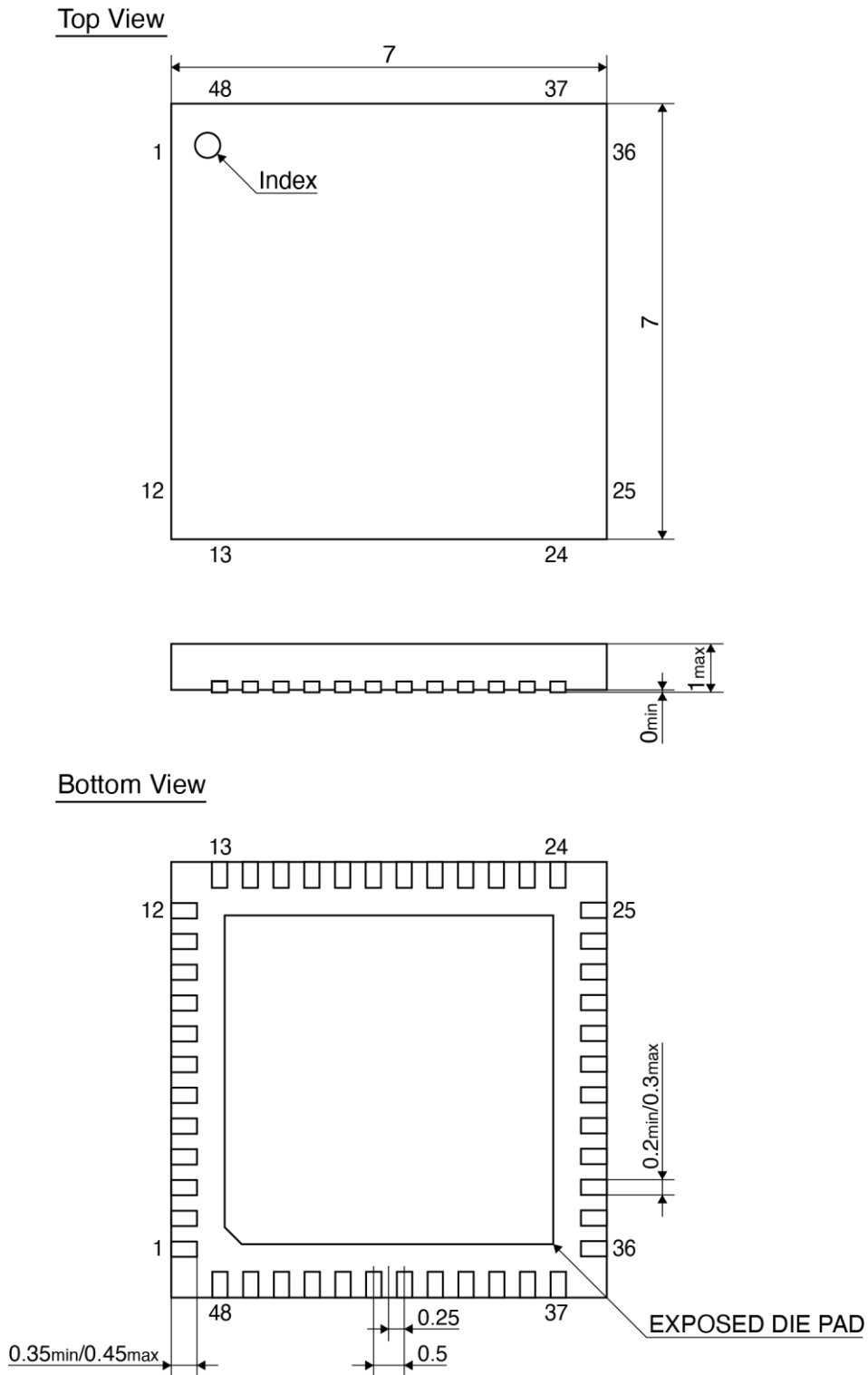
Symbol	Name	Recommended components
CPW1	Bypass capacitor between V _{SS} and V _{DD}	Ceramic capacitor or electrolytic capacitor
CPW2	Capacitor between V _{SS} and V _{D1}	Ceramic capacitor
CLCD1-3	Capacitors between V _{SS} and V _{C1-3}	Ceramic capacitors
CLCD4	Capacitor between C _{P1} and C _{P2}	Ceramic capacitor
BZ	Piezoelectric buzzer	PS1240P02 manufactured by TDK Corporation
RDBG	DSIO pull-up resistor	Thick film chip resistor
CVPP	Capacitor between V _{SS} and V _{PP}	Ceramic capacitor
RREF	RFC reference resistor	Thick film chip resistor
RTMP1,2	Resistive sensors	Temperature sensor 103AP-2 manufactured by SEMITEC Corporation Humidity sensor C15-M53R manufactured by SHINYEI Technology Co.,Ltd. (* In AC oscillation mode for resistive sensor measurements)
CREF	RFC reference capacitor	Ceramic capacitor
CVCAPRF	Capacitor between V _{SS} and V _{CAPRF}	Ceramic capacitor
RLED1,2	LED current-limiting resistors	Thick film chip resistors
CVCAPBZ	Capacitor between V _{SS} and V _{CAPBZ}	Ceramic capacitor
CBUZ	Capacitor between C _{BUZP} and C _{BUZN}	Ceramic capacitor

* For recommended component values, refer to "Recommended Operating Conditions" in the "Electrical Characteristics" chapter.

22 Package

SQFN7-48pin (P-VQFN048-0707-0.50)

(Unit: mm)



* The potential of the EXPOSED DIE PAD is the same as that of the substrate potential (V_{SS}) on the back of the IC.

Figure 22.1 SQFN7-48pin Package Dimensions

Appendix A List of Peripheral Circuit Control Registers

0x4000–0x4008			Misc Registers (MISC)				
Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000	MSCPROT (MISC System Protect Register)	15–0	PROT[15:0]	0x0000	H0	R/W	–
0x4002	MSCIRAMSZ (MISC IRAM Size Register)	15–9	–	0x00	–	R	–
		8	(reserved)	0	H0	R/WP	Always set to 0.
		7–3	–	0x04	–	R	–
		2–0	IRAMSZ[2:0]	0x2	H0	R/WP	–
0x4004	MSCTTBRL (MISC Vector Table Address Low Register)	15–8	TTBR[15:8]	0x80	H0	R/WP	–
		7–0	TTBR[7:0]	0x00	H0	R	–
0x4006	MSCTTBRH (MISC Vector Table Address High Register)	15–8	–	0x00	–	R	–
		7–0	TTBR[23:16]	0x00	H0	R/WP	–
0x4008	MSCPSR (MISC PSR Register)	15–8	–	0x00	–	R	–
		7–5	PSRIL[2:0]	0x0	H0	R	–
		4	PSRIE	0	H0	R	–
		3	PSRC	0	H0	R	–
		2	PSRV	0	H0	R	–
		1	PSRZ	0	H0	R	–
		0	PSRN	0	H0	R	–

0x4020–0x4022			Power Generator (PWG2)				
Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4020	PWGVD1CTL (PWG2 VD1 Regulator Control Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	–
		6	REGDIS	0	H0	R/WP	–
		5–4	REGSEL[1:0]	0x1	H0	R/WP	–
		3–2	–	0x0	–	R	–
		1–0	REGMODE[1:0]	0x0	H0	R/WP	–
0x4022	PWGRFCCTL (PWG2 RFC Regulator Control Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	–
		6	RFCREGDIS	0	H0	R/WP	–
		5–1	–	0x00	–	R	–
		0	(reserved)	0	H0	R/WP	Always set to 0.

0x4040–0x4054			Clock Generator (CLG)				
Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4040	CLGSCLK (CLG System Clock Control Register)	15	WUPMD	0	H0	R/WP	–
		14	–	0	–	R	–
		13–12	WUPDIV[1:0]	0x0	H0	R/WP	–
		11–10	–	0x0	–	R	–
		9–8	WUPSRC[1:0]	0x0	H0	R/WP	–
		7–6	–	0x0	–	R	–
		5–4	CLKDIV[1:0]	0x0	H0	R/WP	–
		3–2	–	0x0	–	R	–
		1–0	CLKSRC[1:0]	0x0	H0	R/WP	–
0x4042	CLGOSC (CLG Oscillation Control Register)	15–12	–	0x0	–	R	–
		11	EXOSCSLPC	1	H0	R/W	–
		10	OSC3SLPC	1	H0	R/W	–
		9	OSC1SLPC	1	H0	R/W	–
		8	IOSCSLPC	1	H0	R/W	–
		7–4	–	0x0	–	R	–
		3	EXOSCEN	0	H0	R/W	–
		2	OSC3EN	0	H0	R/W	–
		1	OSC1EN	0	H0	R/W	–
0	IOSCEN	1	H0	R/W	–		

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4046	CLGOSC1 (CLG OSC1 Control Register)	15-8	-	0x00	-	R	-
		7-2	-	0x00	-	R	-
		1-0	OSC1WT[1:0]	0x1	H0	R/WP	-
0x4048	CLGOSC3 (CLG OSC3 Control Register)	15-13	-	0x00	-	R	-
		12-10	OSC3FQ[2:0]	0x3	-	R/WP	-
		9-8	-	0x0	-	R	-
		7-3	-	0x00	-	R	-
		2-0	OSC3WT[2:0]	0x2	H0	R/WP	-
0x404c	CLGINTF (CLG Interrupt Flag Register)	15-8	-	0x00	-	R	-
		7-3	-	0x0	H0	R	-
		2	OSC3STAIF	0	H0	R/W	Cleared by writing 1.
		1	OSC1STAIF	0	H0	R/W	-
0x404e	CLGINTE (CLG Interrupt Enable Register)	15-8	-	0x00	-	R	-
		7-3	-	0x0	-	R	-
		2	OSC3STAIE	0	H0	R/W	-
		1	OSC1STAIE	0	H0	R/W	-
0x4050	CLGFOUT (CLG FOUT Control Register)	15-8	-	0x00	-	R	-
		7	-	0	-	R	-
		6-4	FOUTDIV[2:0]	0x0	H0	R/W	-
		3-2	FOUTSRC[1:0]	0x0	H0	R/W	-
		1	-	0	-	R	-
0x4052	CLGTRIM1 (CLG Oscillation Frequency Trimming Register 1)	15-14	-	0x0	-	R	-
		13-8	OSC1AJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.
		7-6	-	0x0	-	R	-
		5-0	IOSCAJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.
0x4054	CLGTRIM2 (CLG Oscillation Frequency Trimming Register 2)	15-13	-	0x0	-	R	-
		12-8	OSC3AJ384[4:0]	*	H0	R/WP	* Determined by factory adjustment.
		7-5	-	0x0	-	R	-
		4-0	OSC3AJ4M[4:0]	*	H0	R/WP	* Determined by factory adjustment.

0x4080-0x4090

Interrupt Controller (ITC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4080	ITCLV0 (ITC Interrupt Level Setup Register 0)	15-11	-	0x00	-	R	-
		10-8	ILV1[2:0]	0x0	H0	R/W	Port interrupt (ILVPPORT)
		7-3	-	0x00	-	R	-
		2-0	ILV0[2:0]	0x0	H0	R/W	Supply voltage detector interrupt (ILVSVD5)
0x4082	ITCLV1 (ITC Interrupt Level Setup Register 1)	15-11	-	0x00	-	R	-
		10-8	ILV3[2:0]	0x0	H0	R/W	Clock generator interrupt (ILVCLG)
		7-0	-	0x00	-	R	-
0x4084	ITCLV2 (ITC Interrupt Level Setup Register 2)	15-11	-	0x00	-	R	-
		10-8	ILV5[2:0]	0x0	H0	R/W	16-bit timer Ch.0 interrupt (ILVT16_0)
		7-0	-	0x00	-	R	-
0x4086	ITCLV3 (ITC Interrupt Level Setup Register 3)	15-11	-	0x00	-	R	-
		10-8	ILV7[2:0]	0x0	H0	R/W	16-bit timer Ch.1 interrupt (ILVT16_1)
		7-3	-	0x00	-	R	-
0x4088	ITCLV4 (ITC Interrupt Level Setup Register 4)	2-0	ILV6[2:0]	0x0	H0	R/W	UART Ch.0 interrupt (ILVUART3_0)
		15-11	-	0x00	-	R	-
		10-8	ILV9[2:0]	0x0	H0	R/W	I ² C interrupt (ILVI2C_0)
		7-3	-	0x00	-	R	-
		2-0	ILV8[2:0]	0x0	H0	R/W	Synchronous serial interface Ch.0 interrupt (ILVSPIA_0)

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x408a	ITCLV5 (ITC Interrupt Level Setup Register 5)	15-11	–	0x00	–	R	–
		10-8	ILV11[2:0]	0x0	H0	R/W	16-bit PWM timer Ch.1 interrupt (ILVT16B_1)
		7-3	–	0x00	–	R	–
		2-0	ILV10[2:0]	0x0	H0	R/W	16-bit PWM timer Ch.0 interrupt (ILVT16B_0)
0x408c	ITCLV6 (ITC Interrupt Level Setup Register 6)	15-11	–	0x00	–	R	–
		10-8	ILV13[2:0]	0x0	H0	R/W	16-bit timer Ch.2 interrupt (ILVT16_2)
		7-0	–	0x00	–	R	–
0x408e	ITCLV7 (ITC Interrupt Level Setup Register 7)	15-11	–	0x00	–	R	–
		10-8	ILV15[2:0]	0x0	H0	R/W	LCD driver interrupt (ILVLCD4B)
		7-3	–	0x00	–	R	–
		2-0	ILV14[2:0]	0x0	H0	R/W	Sound generator interrupt (ILVSNDA2)
0x4090	ITCLV8 (ITC Interrupt Level Setup Register 8)	15-11	–	0x00	–	R	–
		10-8	ILV17[2:0]	0x0	H0	R/W	EEPROM controller interrupt (ILVEPRC)
		7-3	–	0x00	–	R	–
		2-0	ILV16[2:0]	0x0	H0	R/W	R/F converter Ch.0 interrupt (ILVRFC2_0)

0x40a0–0x40a4

Watchdog Timer (WDT2)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x40a0	WDTCLK (WDT2 Clock Control Register)	15-9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/WP	–
		7-6	–	0x0	–	R	–
		5-4	CLKDIV[1:0]	0x0	H0	R/WP	–
		3-2	–	0x0	–	R	–
		1-0	CLKSRC[1:0]	0x0	H0	R/WP	–
0x40a2	WDTCTL (WDT2 Control Register)	15-11	–	0x00	–	R	–
		10-9	MOD[1:0]	0x0	H0	R/WP	–
		8	STATNMI	0	H0	R	–
		7-5	–	0x0	–	R	–
		4	WDTCTRST	0	H0	WP	Always read as 0.
		3-0	WDTRUN[3:0]	0xa	H0	R/WP	–
0x40a4	WDTCMP (WDT2 Counter Compare Match Register)	15-10	–	0x00	–	R	–
		9-0	CMP[9:0]	0x3ff	H0	R/WP	–

0x4100–0x4106

Supply Voltage Detector (SVD5)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4100	SVDCLK (SVD5 Clock Control Register)	15-9	–	0x00	–	R	–
		8	DBRUN	1	H0	R/WP	–
		7	–	0	–	R	–
		6-4	CLKDIV[2:0]	0x0	H0	R/WP	–
		3-2	–	0x0	–	R	–
		1-0	CLKSRC[1:0]	0x0	H0	R/WP	–
0x4102	SVDCTL (SVD5 Control Register)	15	VSEL	0	H1	R/WP	–
		14-13	SVDSC[1:0]	0x0	H0	R/WP	Writing takes effect when the SVDCTL.SVDMD[1:0] bits are not 0x0.
		12-8	SVDC[4:0]	0x1e	H1	R/WP	–
		7-4	SVDRE[3:0]	0x0	H1	R/WP	–
		3	–	0	–	R	–
		2-1	SVDMD[1:0]	0x0	H0	R/WP	–
		0	MODEN	0	H1	R/WP	–
0x4104	SVDINTF (SVD5 Status and Interrupt Flag Register)	15-9	–	0x00	–	R	–
		8	SVDDT	x	–	R	–
		7-1	–	0x00	–	R	–
		0	SVDIF	0	H1	R/W	Cleared by writing 1.

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4106	SVDINTE (SVD5 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	SVDIE	0	H0	R/W	

0x4160–0x416c

16-bit Timer (T16) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4160	T16_0CLK (T16 Ch.0 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x4162	T16_0MOD (T16 Ch.0 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4164	T16_0CTL (T16 Ch.0 Control Register)	15–9	–	0x00	–	R	–
		8	PRUN	0	H0	R/W	
		7–2	–	0x00	–	R	
		1	PRESET	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x4166	T16_OTR (T16 Ch.0 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4168	T16_0TC (T16 Ch.0 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x416a	T16_0INTF (T16 Ch.0 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	
0x416c	T16_0INTE (T16 Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x41b0

Flash Controller (FLASHC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x41b0	FLASHCWAIT (FLASHC Flash Read Cycle Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1–0	RDWAIT[1:0]	0x1	H0	R/WP	

0x41c0–0x41ca

EEPROM Controller (EEPROMC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	
0x41c0	EPRCCTL0 (EEPROMC Control Register 0)	15–9	–	0x00	–	R	–	
		8	EP_XPOR	1	H0	R/WP		
		7–2	–	0x00	–	R		
		1	EP_PWRSET	0	H0	R/WP		
		0	EP_WMODE	0	H0	R/WP		
0x41c2	EPRCCTL1 (EEPROMC Control Register 1)	15–8	–	0x00	–	R	–	
		7–1	–	0x00	–	R		
		0	EP_CK	0	H0	WP		
0x41c4	EPRCADR (EEPROMC Address Register)	15–8	–	0x00	–	R	–	
		7–0	EP_ADDR[7:0]	0x00	H0	R/WP		
0x41c6	EPRCWDAT (EEPROMC Write Data Register)	15–8	–	0x00	–	R	–	
		7–0	EP_WDAT[7:0]	0x00	H0	R/WP		
0x41c8	EPRCINTF (EEPROMC Interrupt Flag Register)	15–8	–	0x00	–	R	–	
		7–2	–	0x00	–	R		
		1	ECCERIF	0	H0	R/WP		Cleared by writing 1.
		0	RXBIF	0	H0	R/WP		

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x41ca	EPRCINTE (EEPROMC Interrupt Enable Register)	15-8	-	0x00	-	R	
		7-2	-	0x00	-	R	
		1	ECCERIE	0	H0	R/WP	
		0	RXBIE	0	H0	R/WP	

0x4200-0x42e2

I/O Ports (PPORT)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4200	PODAT (P0 Port Data Register)	15-8	P0OUT[7:0]	0x00	H0	R/W	
		7-0	P0IN[7:0]	0x00	H0	R	
0x4202	P0IOEN (P0 Port Enable Register)	15-8	P0IEN[7:0]	0x00	H0	R/W	
		7-0	P0OEN[7:0]	0x00	H0	R/W	
0x4204	PORCTL (P0 Port Pull-up/down Control Register)	15-12	P0PDPU[7:4]	0x0	H0	R/W	
		11-10	-	0x0	-	R	
		9-8	P0PDPU[1:0]	0x0	H0	R/W	
		7-4	POREN[7:4]	0x0	H0	R/W	
		3-2	-	0x0	-	R	
		1-0	POREN[1:0]	0x0	H0	R/W	
0x4206	POINTF (P0 Port Interrupt Flag Register)	15-8	-	0x00	-	R	
		7-0	P0IF[7:0]	0x00	H0	R/W	
0x4208	POINTCTL (P0 Port Interrupt Control Register)	15-8	P0EDGE[7:0]	0x00	H0	R/W	
		7-0	P0IE[7:0]	0x00	H0	R/W	
0x420a	POCHATEN (P0 Port Chattering Filter Enable Register)	15-8	-	0x00	-	R	
		7-4	POCHATEN[7:4]	0x0	H0	R/W	
		3-2	-	0x0	-	R	
		1-0	POCHATEN[1:0]	0x0	H0	R/W	
0x420c	P0MODESEL (P0 Port Mode Select Register)	15-8	-	0x00	-	R	
		7-0	P0SEL[7:0]	0x00	H0	R/W	
0x420e	POFNCSEL (P0 Port Function Select Register)	15-14	P07MUX[1:0]	0x0	H0	R/W	
		13-12	P06MUX[1:0]	0x0	H0	R/W	
		11-10	P05MUX[1:0]	0x0	H0	R/W	
		9-8	P04MUX[1:0]	0x0	H0	R/W	
		7-6	P03MUX[1:0]	0x0	H0	R/W	
		5-4	P02MUX[1:0]	0x0	H0	R/W	
		3-2	P01MUX[1:0]	0x0	H0	R/W	
		1-0	P00MUX[1:0]	0x0	H0	R/W	
0x4210	P1DAT (P1 Port Data Register)	15-8	P1OUT[7:0]	0x00	H0	R/W	
		7-0	P1IN[7:0]	0x00	H0	R	
0x4212	P1IOEN (P1 Port Enable Register)	15-8	P1IEN[7:0]	0x00	H0	R/W	
		7-0	P1OEN[7:0]	0x00	H0	R/W	
0x4214	P1RCTL (P1 Port Pull-up/down Control Register)	15-8	P1PDPU[7:0]	0x00	H0	R/W	
		7-0	P1REN[7:0]	0x00	H0	R/W	
0x4216	P1INTF (P1 Port Interrupt Flag Register)	15-8	-	0x00	-	R	
		7-0	P1IF[7:0]	0x00	H0	R/W	
0x4218	P1INTCTL (P1 Port Interrupt Control Register)	15-8	P1EDGE[7:0]	0x00	H0	R/W	
		7-0	P1IE[7:0]	0x00	H0	R/W	
0x421a	P1CHATEN (P1 Port Chattering Filter Enable Register)	15-8	-	0x00	-	R	
		7-0	P1CHATEN[7:0]	0x00	H0	R/W	
0x421c	P1MODESEL (P1 Port Mode Select Register)	15-8	-	0x00	-	R	
		7-0	P1SEL[7:0]	0x00	H0	R/W	

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x421e	P1FNCSEL (P1 Port Function Select Register)	15-14	P17MUX[1:0]	0x0	H0	R/W	-
		13-12	P16MUX[1:0]	0x0	H0	R/W	
		11-10	P15MUX[1:0]	0x0	H0	R/W	
		9-8	P14MUX[1:0]	0x0	H0	R/W	
		7-6	P13MUX[1:0]	0x0	H0	R/W	
		5-4	P12MUX[1:0]	0x0	H0	R/W	
		3-2	P11MUX[1:0]	0x0	H0	R/W	
1-0	P10MUX[1:0]	0x0	H0	R/W			
0x4220	P2DAT (P2 Port Data Register)	15-14	-	0x0	-	R	-
		13-8	P2OUT[5:0]	0x00	H0	R/W	
		7-6	-	0x0	-	R	
0x4222	P2IOEN (P2 Port Enable Register)	5-0	P2IN[5:0]	0x00	H0	R	-
		15-14	-	0x0	-	R	
		13-8	P2IEN[5:0]	0x00	H0	R/W	
0x4224	P2RCTL (P2 Port Pull-up/down Control Register)	7-6	-	0x0	-	R	-
		5-0	P2REN[5:0]	0x00	H0	R/W	
		15-14	-	0x0	-	R	
0x4226	P2INTF (P2 Port Interrupt Flag Register)	7-6	-	0x0	-	R	-
		5-0	P2IF[5:0]	0x00	H0	R/W	
		15-8	-	0x00	-	R	
0x4228	P2INTCTL (P2 Port Interrupt Control Register)	7-6	-	0x0	-	R	-
		5-0	P2IE[5:0]	0x00	H0	R/W	
		13-8	P2EDGE[5:0]	0x00	H0	R/W	
0x422a	P2CHATEN (P2 Port Chattering Filter Enable Register)	7-6	-	0x0	-	R	-
		5-0	P2CHATEN[5:0]	0x00	H0	R/W	
		15-8	-	0x00	-	R	
0x422c	P2MODESEL (P2 Port Mode Select Register)	7-6	-	0x0	-	R	-
		5-0	P2SEL[5:0]	0x00	H0	R/W	
		15-12	-	0x0	-	R	
0x422e	P2FNCSEL (P2 Port Function Select Register)	11-10	P25MUX[1:0]	0x0	H0	R/W	-
		9-8	P24MUX[1:0]	0x0	H0	R/W	
		7-6	P23MUX[1:0]	0x0	H0	R/W	
		5-4	P22MUX[1:0]	0x0	H0	R/W	
		3-2	P21MUX[1:0]	0x0	H0	R/W	
		1-0	P20MUX[1:0]	0x0	H0	R/W	
		15-11	-	0x00	-	R	
0x42d0	PDDAT (Pd Port Data Register)	10	PDOUT2	0	H0	R/W	-
		9	PDOUT1	0	H0	R/W	
		8	PDOUT0	0	H0	R/W	
		7-3	-	0x00	-	R	
		2	-	0	-	R	
		1	PDIN1	X	H0	R	
		0	PDIN0	X	H0	R	
		15-11	-	0x00	-	R	
0x42d2	PDIOEN (Pd Port Enable Register)	10	(reserved)	0	H0	R/W	-
		9	PDIEN1	0	H0	R/W	
		8	PDIEN0	0	H0	R/W	
		7-3	-	0x00	-	R	
		2	PDOEN2	0	H0	R/W	
		1	PDOEN1	0	H0	R/W	
		0	PDOEN0	0	H0	R/W	
0x42d4	PDRCTL (Pd Port Pull-up/down Control Register)	10	(reserved)	0	H0	R/W	-
		9	PDPDPU1	0	H0	R/W	
		8	PDPDPU0	0	H0	R/W	
		7-5	-	0x00	-	R	
		2	(reserved)	0	H0	R/W	
		1	PDREN1	0	H0	R/W	
		0	PDREN0	0	H0	R/W	

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x42dc	PDMODESEL (Pd Port Mode Select Register)	15-8	–	0x00	–	R	–
		7-3	–	0	–	R	
		2	PDSEL2	1	H0	R/W	
		1	PDSEL1	1	H0	R/W	
		0	PDSEL0	1	H0	R/W	
0x42de	PDFNCSEL (Pd Port Function Select Register)	15-8	–	0x00	–	R	–
		7-6	–	0x0	–	R	
		5-4	PD2MUX[1:0]	0x0	H0	R/W	
		3-2	PD1MUX[1:0]	0x0	H0	R/W	
		1-0	PD0MUX[1:0]	0x0	H0	R/W	
0x42e0	PCLK (P Port Clock Control Register)	15-9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/WP	
		7-4	CLKDIV[3:0]	0x0	H0	R/WP	
		3-2	KRSTCFG[1:0]	0x0	H0	R/WP	
		1-0	CLKSRC[1:0]	0x0	H0	R/WP	
0x42e2	PINTFRGP (P Port Interrupt Flag Group Register)	15-8	–	0x00	–	R	–
		7-3	–	0x00	–	R	
		2	P2INT	0	H0	R	
		1	P1INT	0	H0	R	
		0	POINT	0	H0	R	

0x4300–0x430e

Universal Port Multiplexer (UPMUX)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4300	P0UPMUX0 (P00–01 Universal Port Multiplexer Setting Register)	15-13	P01PPFNC[2:0]	0x0	H0	R/W	–
		12-11	P01PERICH[1:0]	0x0	H0	R/W	
		10-8	P01PERISEL[2:0]	0x0	H0	R/W	
		7-5	P00PPFNC[2:0]	0x0	H0	R/W	
		4-3	P00PERICH[1:0]	0x0	H0	R/W	
		2-0	P00PERISEL[2:0]	0x0	H0	R/W	
0x4302	P0UPMUX1 (P02–03 Universal Port Multiplexer Setting Register)	15-13	P03PPFNC[2:0]	0x0	H0	R/W	–
		12-11	P03PERICH[1:0]	0x0	H0	R/W	
		10-8	P03PERISEL[2:0]	0x0	H0	R/W	
		7-5	P02PPFNC[2:0]	0x0	H0	R/W	
		4-3	P02PERICH[1:0]	0x0	H0	R/W	
		2-0	P02PERISEL[2:0]	0x0	H0	R/W	
0x4304	P0UPMUX2 (P04–05 Universal Port Multiplexer Setting Register)	15-13	P05PPFNC[2:0]	0x0	H0	R/W	–
		12-11	P05PERICH[1:0]	0x0	H0	R/W	
		10-8	P05PERISEL[2:0]	0x0	H0	R/W	
		7-5	P04PPFNC[2:0]	0x0	H0	R/W	
		4-3	P04PERICH[1:0]	0x0	H0	R/W	
		2-0	P04PERISEL[2:0]	0x0	H0	R/W	
0x4306	P0UPMUX3 (P06–07 Universal Port Multiplexer Setting Register)	15-13	P07PPFNC[2:0]	0x0	H0	R/W	–
		12-11	P07PERICH[1:0]	0x0	H0	R/W	
		10-8	P07PERISEL[2:0]	0x0	H0	R/W	
		7-5	P06PPFNC[2:0]	0x0	H0	R/W	
		4-3	P06PERICH[1:0]	0x0	H0	R/W	
		2-0	P06PERISEL[2:0]	0x0	H0	R/W	
0x4308	P1UPMUX0 (P10–11 Universal Port Multiplexer Setting Register)	15-13	P11PPFNC[2:0]	0x0	H0	R/W	–
		12-11	P11PERICH[1:0]	0x0	H0	R/W	
		10-8	P11PERISEL[2:0]	0x0	H0	R/W	
		7-5	P10PPFNC[2:0]	0x0	H0	R/W	
		4-3	P10PERICH[1:0]	0x0	H0	R/W	
		2-0	P10PERISEL[2:0]	0x0	H0	R/W	
0x430a	P1UPMUX1 (P12–13 Universal Port Multiplexer Setting Register)	15-13	P13PPFNC[2:0]	0x0	H0	R/W	–
		12-11	P13PERICH[1:0]	0x0	H0	R/W	
		10-8	P13PERISEL[2:0]	0x0	H0	R/W	
		7-5	P12PPFNC[2:0]	0x0	H0	R/W	
		4-3	P12PERICH[1:0]	0x0	H0	R/W	
		2-0	P12PERISEL[2:0]	0x0	H0	R/W	

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x430c	P1UPMUX2 (P14–15 Universal Port Multiplexer Setting Register)	15–13	P15PPFNC[2:0]	0x0	H0	R/W	–
		12–11	P15PERICH[1:0]	0x0	H0	R/W	
		10–8	P15PERISEL[2:0]	0x0	H0	R/W	
		7–5	P14PPFNC[2:0]	0x0	H0	R/W	
		4–3	P14PERICH[1:0]	0x0	H0	R/W	
		2–0	P14PERISEL[2:0]	0x0	H0	R/W	
0x430e	P1UPMUX3 (P16–17 Universal Port Multiplexer Setting Register)	15–13	P17PPFNC[2:0]	0x0	H0	R/W	–
		12–11	P17PERICH[1:0]	0x0	H0	R/W	
		10–8	P17PERISEL[2:0]	0x0	H0	R/W	
		7–5	P16PPFNC[2:0]	0x0	H0	R/W	
		4–3	P16PERICH[1:0]	0x0	H0	R/W	
		2–0	P16PERISEL[2:0]	0x0	H0	R/W	

0x4380–0x4390

UART (UART3) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	
0x4380	UA0CLK (UART3 Ch.0 Clock Control Register)	15–9	–	0x00	–	R	–	
		8	DBRUN	0	H0	R/W		
		7–6	–	0x0	–	R		
		5–4	CLKDIV[1:0]	0x0	H0	R/W		
		3–2	–	0x0	–	R		
		1–0	CLKSRC[1:0]	0x0	H0	R/W		
0x4382	UA0MOD (UART3 Ch.0 Mode Register)	15–13	–	0x0	–	R	–	
		12	PECAR	0	H0	R/W		
		11	CAREN	0	H0	R/W		
		10	BRDIV	0	H0	R/W		
		9	INVRX	0	H0	R/W		
		8	INVTX	0	H0	R/W		
		7	–	0	–	R		
		6	PUEN	0	H0	R/W		
		5	OUTMD	0	H0	R/W		
		4	IRMD	0	H0	R/W		
		3	CHLN	0	H0	R/W		
		2	PREN	0	H0	R/W		
		1	PRMD	0	H0	R/W		
		0	STPB	0	H0	R/W		
0x4384	UA0BR (UART3 Ch.0 Baud-Rate Register)	15–12	–	0x0	–	R	–	
		11–8	FMD[3:0]	0x0	H0	R/W		
		7–0	BRT[7:0]	0x00	H0	R/W		
0x4386	UA0CTL (UART3 Ch.0 Control Register)	15–8	–	0x00	–	R	–	
		7–2	–	0x00	–	R		
		1	SFTRST	0	H0	R/W		
0x4388	UA0TXD (UART3 Ch.0 Transmit Data Register)	0	MODEN	0	H0	R/W	–	
		15–8	–	0x00	–	R		
		7–0	TXD[7:0]	0x00	H0	R/W		
0x438a	UA0RXD (UART3 Ch.0 Receive Data Register)	15–8	–	0x00	–	R	–	
		7–0	RXD[7:0]	0x00	H0	R		
0x438c	UA0INTF (UART3 Ch.0 Status and Interrupt Flag Register)	15–10	–	0x00	–	R	–	
		9	RBSY	0	H0/S0	R		
		8	TBSY	0	H0/S0	R		
		7	–	0	–	R		
		6	TENDIF	0	H0/S0	R/W		Cleared by writing 1.
		5	FEIF	0	H0/S0	R/W		Cleared by writing 1 or reading the UA0RXD register.
		4	PEIF	0	H0/S0	R/W		
		3	OEIF	0	H0/S0	R/W		Cleared by writing 1.
		2	RB2FIF	0	H0/S0	R		Cleared by reading the UA0RXD register.
		1	RB1FIF	0	H0/S0	R		
0	TBEIF	1	H0/S0	R	Cleared by writing to the UA0TXD register.			

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x438e	UA0INTE (UART3 Ch.0 Interrupt Enable Register)	15-8	-	0x00	-	R	
		7	-	0	-	R	
		6	TENDIE	0	H0	R/W	
		5	FEIE	0	H0	R/W	
		4	PEIE	0	H0	R/W	
		3	OEIE	0	H0	R/W	
		2	RB2FIE	0	H0	R/W	
		1	RB1FIE	0	H0	R/W	
0x4390	UA0CAWF (UART3 Ch.0 Carrier Waveform Register)	15-8	-	0x00	-	R	
		7-0	CRPER[7:0]	0x00	H0	R/W	

0x43a0-0x43ac

16-bit Timer (T16) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43a0	T16_1CLK (T16 Ch.1 Clock Control Register)	15-9	-	0x00	-	R	
		8	DBRUN	0	H0	R/W	
		7-4	CLKDIV[3:0]	0x0	H0	R/W	
		3-2	-	0x0	-	R	
		1-0	CLKSRC[1:0]	0x0	H0	R/W	
0x43a2	T16_1MOD (T16 Ch.1 Mode Register)	15-8	-	0x00	-	R	
		7-1	-	0x00	-	R	
		0	TRMD	0	H0	R/W	
0x43a4	T16_1CTL (T16 Ch.1 Control Register)	15-9	-	0x00	-	R	
		8	PRUN	0	H0	R/W	
		7-2	-	0x00	-	R	
		1	PRESET	0	H0	R/W	
0x43a6	T16_1TR (T16 Ch.1 Reload Data Register)	15-0	TR[15:0]	0xffff	H0	R/W	
		15-0	TC[15:0]	0xffff	H0	R	
0x43aa	T16_1INTF (T16 Ch.1 Interrupt Flag Register)	15-8	-	0x00	-	R	Cleared by writing 1.
7-1	-	0x00	-	R			
0	UFIF	0	H0	R/W			
0x43ac	T16_1INTE (T16 Ch.1 Interrupt Enable Register)	15-8	-	0x00	-	R	
		7-1	-	0x00	-	R	
		0	UFIE	0	H0	R/W	

0x43b0-0x43ba

Synchronous Serial Interface (SPIA) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43b0	SPI0MOD (SPIA Ch.0 Mode Register)	15-12	-	0x0	-	R	
		11-8	CHLN[3:0]	0x7	H0	R/W	
		7-6	-	0x0	-	R	
		5	PUEN	0	H0	R/W	
		4	NOCLKDIV	0	H0	R/W	
		3	LSBFST	0	H0	R/W	
		2	CPHA	0	H0	R/W	
		1	CPOL	0	H0	R/W	
		0	MST	0	H0	R/W	
0x43b2	SPI0CTL (SPIA Ch.0 Control Register)	15-8	-	0x00	-	R	
		7-2	-	0x00	-	R	
		1	SFTRST	0	H0	R/W	
0x43b4	SPI0TXD (SPIA Ch.0 Transmit Data Register)	15-0	TXD[15:0]	0x0000	H0	R/W	
		15-0	RXD[15:0]	0x0000	H0	R	

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43b8	SPI0INTF (SPIA Ch.0 Interrupt Flag Register)	15-8	–	0x00	–	R	–
		7	BSY	0	H0	R	
		6-4	–	0x0	–	R	
		3	OEIF	0	H0/S0	R/W	Cleared by writing 1.
		2	TENDIF	0	H0/S0	R/W	
		1	RBFIF	0	H0/S0	R	Cleared by reading the SPI0RXD register.
		0	TBEIF	1	H0/S0	R	Cleared by writing to the SPI0TXD register.
0x43ba	SPI0INTE (SPIA Ch.0 Interrupt Enable Register)	15-8	–	0x00	–	R	–
		7-4	–	0x0	–	R	
		3	OEIE	0	H0	R/W	
		2	TENDIE	0	H0	R/W	
		1	RBFIE	0	H0	R/W	
		0	TBEIE	0	H0	R/W	

0x43c0–0x43d2

I²C (I2C) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43c0	I2C0CLK (I2C Ch.0 Clock Control Register)	15-9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7-6	–	0x0	–	R	
		5-4	CLKDIV[1:0]	0x0	H0	R/W	
		3-2	–	0x0	–	R	
		1-0	CLKSRC[1:0]	0x0	H0	R/W	
0x43c2	I2C0MOD (I2C Ch.0 Mode Register)	15-8	–	0x00	–	R	–
		7-3	–	0x00	–	R	
		2	OADR10	0	H0	R/W	
		1	GCEN	0	H0	R/W	
		0	–	0	–	R	
0x43c4	I2C0BR (I2C Ch.0 Baud-Rate Register)	15-8	–	0x00	–	R	–
		7	–	0	–	R	
		6-0	BRT[6:0]	0x7f	H0	R/W	
0x43c8	I2C0OADR (I2C Ch.0 Own Address Register)	15-10	–	0x00	–	R	–
		9-0	OADR[9:0]	0x000	H0	R/W	
0x43ca	I2C0CTL (I2C Ch.0 Control Register)	15-8	–	0x00	–	R	–
		7-6	–	0x0	–	R	
		5	MST	0	H0	R/W	
		4	TXNACK	0	H0/S0	R/W	
		3	TXSTOP	0	H0/S0	R/W	
		2	TXSTART	0	H0/S0	R/W	
		1	SFTRST	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x43cc	I2C0TXD (I2C Ch.0 Transmit Data Register)	15-8	–	0x00	–	R	–
		7-0	TXD[7:0]	0x00	H0	R/W	
0x43ce	I2C0RXD (I2C Ch.0 Receive Data Register)	15-8	–	0x00	–	R	–
		7-0	RXD[7:0]	0x00	H0	R	

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x43d0	I2C0INTF (I2C Ch.0 Status and Interrupt Flag Register)	15-13	–	0x0	–	R	Cleared by writing 1. Cleared by reading the I2C0RXD register. Cleared by writing to the I2C0TXD register.
		12	SDALLOW	0	H0	R	
		11	SCLLOW	0	H0	R	
		10	BSY	0	H0/S0	R	
		9	TR	0	H0	R	
		8	–	0	–	R	
		7	BYTEENDIF	0	H0/S0	R/W	
		6	GCIF	0	H0/S0	R/W	
		5	NACKIF	0	H0/S0	R/W	
		4	STOPIF	0	H0/S0	R/W	
		3	STARTIF	0	H0/S0	R/W	
		2	ERRIF	0	H0/S0	R/W	
		1	RBFIF	0	H0/S0	R	
0	TBEIF	0	H0/S0	R			
0x43d2	I2C0INTE (I2C Ch.0 Interrupt Enable Register)	15-8	–	0x00	–	R	–
		7	BYTEENDIE	0	H0	R/W	
		6	GCIE	0	H0	R/W	
		5	NACKIE	0	H0	R/W	
		4	STOPIE	0	H0	R/W	
		3	STARTIE	0	H0	R/W	
		2	ERRIE	0	H0	R/W	
		1	RBFIE	0	H0	R/W	
		0	TBEIE	0	H0	R/W	

0x5000–0x501a

16-bit PWM Timer (T16B) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5000	T16B0CLK (T16B Ch.0 Clock Control Register)	15-9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7-4	CLKDIV[3:0]	0x0	H0	R/W	
		3	–	0	–	R	
		2-0	CLKSRC[2:0]	0x0	H0	R/W	
0x5002	T16B0CTL (T16B Ch.0 Counter Control Register)	15-9	–	0x00	–	R	–
		8	MAXBSY	0	H0	R	
		7-6	–	0x0	–	R	
		5-4	CNTMD[1:0]	0x0	H0	R/W	
		3	ONEST	0	H0	R/W	
		2	RUN	0	H0	R/W	
		1	PRESET	0	H0	R/W	
0	MODEN	0	H0	R/W			
0x5004	T16B0MC (T16B Ch.0 Max Counter Data Register)	15-0	MC[15:0]	0xffff	H0	R/W	–
0x5006	T16B0TC (T16B Ch.0 Timer Counter Data Register)	15-0	TC[15:0]	0x0000	H0	R	–
0x5008	T16B0CS (T16B Ch.0 Counter Status Register)	15-8	–	0x00	–	R	–
		7-4	–	0x0	–	R	
		3	CAPI1	0	H0	R	
		2	CAPI0	0	H0	R	
		1	UP_DOWN	1	H0	R	
0	BSY	0	H0	R			
0x500a	T16B0INTF (T16B Ch.0 Interrupt Flag Register)	15-8	–	0x00	–	R	Cleared by writing 1.
		7-6	–	0x0	–	R	
		5	CAPOW1IF	0	H0	R/W	
		4	CMPCAP1IF	0	H0	R/W	
		3	CAPOW0IF	0	H0	R/W	
		2	CMPCAP0IF	0	H0	R/W	
		1	CNTMAXIF	0	H0	R/W	
0	CNTZEROIF	0	H0	R/W			

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x500c	T16B0INTE (T16B Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPOW1IE	0	H0	R/W	
		4	CMPCAP1IE	0	H0	R/W	
		3	CAPOW0IE	0	H0	R/W	
		2	CMPCAP0IE	0	H0	R/W	
		1	CNTMAXIE	0	H0	R/W	
		0	CNTZEROIE	0	H0	R/W	
0x5010	T16B0CCCTL0 (T16B Ch.0 Compare/Capture 0 Control Register)	15	SCS	0	H0	R/W	–
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
		4–2	TOUTMD[2:0]	0x0	H0	R/W	
0x5012	T16B0CCR0 (T16B Ch.0 Compare/Capture 0 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
		15	SCS	0	H0	R/W	
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
0x5018	T16B0CCCTL1 (T16B Ch.0 Compare/Capture 1 Control Register)	15	SCS	0	H0	R/W	–
		14–12	CBUFMD[2:0]	0x0	H0	R/W	
		11–10	CAPIS[1:0]	0x0	H0	R/W	
		9–8	CAPTRG[1:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6	TOUTMT	0	H0	R/W	
		5	TOUTO	0	H0	R/W	
		4–2	TOUTMD[2:0]	0x0	H0	R/W	
0x501a	T16B0CCR1 (T16B Ch.0 Compare/Capture 1 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–

0x5040–0x505a

16-bit PWM Timer (T16B) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5040	T16B1CLK (T16B Ch.1 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3	–	0	–	R	
		2–0	CLKSRC[2:0]	0x0	H0	R/W	
0x5042	T16B1CTL (T16B Ch.1 Counter Control Register)	15–9	–	0x00	–	R	–
		8	MAXBSY	0	H0	R	
		7–6	–	0x0	–	R	
		5–4	CNTMD[1:0]	0x0	H0	R/W	
		3	ONEST	0	H0	R/W	
		2	RUN	0	H0	R/W	
		1	PRESET	0	H0	R/W	
0	MODEN	0	H0	R/W			
0x5044	T16B1MC (T16B Ch.1 Max Counter Data Register)	15–0	MC[15:0]	0xffff	H0	R/W	–
0x5046	T16B1TC (T16B Ch.1 Timer Counter Data Register)	15–0	TC[15:0]	0x0000	H0	R	–
0x5048	T16B1CS (T16B Ch.1 Counter Status Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3	CAP11	0	H0	R	
		2	CAP10	0	H0	R	
		1	UP_DOWN	1	H0	R	
		0	BSY	0	H0	R	

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks			
0x504a	T16B1INTF (T16B Ch.1 Interrupt Flag Register)	15-8	–	0x00	–	R	Cleared by writing 1.			
		7-6	–	0x0	–	R				
		5	CAPOW1IF	0	H0	R/W				
		4	CMPCAP1IF	0	H0	R/W				
		3	CAPOW0IF	0	H0	R/W				
		2	CMPCAP0IF	0	H0	R/W				
		1	CNTMAXIF	0	H0	R/W				
0x504c	T16B1INTE (T16B Ch.1 Interrupt Enable Register)	15-8	–	0x00	–	R				
		7-6	–	0x0	–	R				
		5	CAPOW1IE	0	H0	R/W				
		4	CMPCAP1IE	0	H0	R/W				
		3	CAPOW0IE	0	H0	R/W				
		2	CMPCAP0IE	0	H0	R/W				
		1	CNTMAXIE	0	H0	R/W				
0x5050	T16B1CCCTL0 (T16B Ch.1 Compare/ Capture 0 Control Register)	15	SCS	0	H0	R/W				
		14-12	CBUFMD[2:0]	0x0	H0	R/W				
		11-10	CAPIS[1:0]	0x0	H0	R/W				
		9-8	CAPTRG[1:0]	0x0	H0	R/W				
		7	–	0	–	R				
		6	TOUTMT	0	H0	R/W				
		5	TOUTO	0	H0	R/W				
		4-2	TOUTMD[2:0]	0x0	H0	R/W				
0x5052	T16B1CCR0 (T16B Ch.1 Compare/ Capture 0 Data Register)	15-0	CC[15:0]	0x0000	H0	R/W				
		0x5058	T16B1CCCTL1 (T16B Ch.1 Compare/ Capture 1 Control Register)	15	SCS	0		H0	R/W	
				14-12	CBUFMD[2:0]	0x0		H0	R/W	
				11-10	CAPIS[1:0]	0x0		H0	R/W	
				9-8	CAPTRG[1:0]	0x0		H0	R/W	
				7	–	0		–	R	
				6	TOUTMT	0		H0	R/W	
				5	TOUTO	0		H0	R/W	
4-2	TOUTMD[2:0]			0x0	H0	R/W				
0x505a	T16B1CCR1 (T16B Ch.1 Compare /Capture 1 Data Register)	15-0	CC[15:0]	0x0000	H0	R/W				

0x5260–0x526c

16-bit Timer (T16) Ch.2

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5260	T16_2CLK (T16 Ch.2 Clock Control Register)	15-9	–	0x00	–	R	
		8	DBRUN	0	H0	R/W	
		7-4	CLKDIV[3:0]	0x0	H0	R/W	
		3-2	–	0x0	–	R	
		1-0	CLKSRC[1:0]	0x0	H0	R/W	
0x5262	T16_2MOD (T16 Ch.2 Mode Register)	15-8	–	0x00	–	R	
		7-1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x5264	T16_2CTL (T16 Ch.2 Control Register)	15-9	–	0x00	–	R	
		8	PRUN	0	H0	R/W	
		7-2	–	0x00	–	R	
		1	PRESET	0	H0	R/W	
0x5266	T16_2TR (T16 Ch.2 Reload Data Register)	0	MODEN	0	H0	R/W	
		15-0	TR[15:0]	0xffff	H0	R/W	

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5268	T16_2TC (T16 Ch.2 Counter Data Register)	15-0	TC[15:0]	0xffff	H0	R	–
0x526a	T16_2INTF (T16 Ch.2 Interrupt Flag Register)	15-8	–	0x00	–	R	–
		7-1	–	0x00	–	R	–
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x526c	T16_2INTE (T16 Ch.2 Interrupt Enable Register)	15-8	–	0x00	–	R	–
		7-1	–	0x00	–	R	–
		0	UFIE	0	H0	R/W	–

0x5300–0x530a

Sound Generator (SNDA2)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5300	SNDCLK (SNDA2 Clock Control Register)	15-9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	–
		7	–	0	–	R	–
		6-4	CLKDIV[2:0]	0x0	H0	R/W	–
		3-2	–	0x0	–	R	–
		1-0	CLKSRC[1:0]	0x0	H0	R/W	–
0x5302	SNDSEL (SNDA2 Select Register)	15-12	–	0x0	–	R	–
		11-8	STIM[3:0]	0x0	H0	R/W	–
		7-3	–	0x00	–	R	–
		2	SINV	0	H0	R/W	–
		1-0	MOSEL[1:0]	0x0	H0	R/W	–
0x5304	SNDCTL (SNDA2 Control Register)	15-9	–	0x00	–	R	–
		8	SSTP	0	H0	R/W	–
		7-1	–	0x00	–	R	–
		0	MODEN	0	H0	R/W	–
0x5306	SNDDAT (SNDA2 Data Register)	15	MDTI	0	H0	R/W	–
		14	MDRS	0	H0	R/W	–
		13-8	SLEN[5:0]	0x00	H0	R/W	–
		7-0	SFRQ[7:0]	0xff	H0	R/W	–
0x5308	SNDINTF (SNDA2 Interrupt Flag Register)	15-9	–	0x00	–	R	–
		8	SBSY	0	H0	R	–
		7-2	–	0x00	–	R	–
		1	EMIF	1	H0	R	Cleared by writing to the SNDDAT register.
		0	EDIF	0	H0	R/W	Cleared by writing 1 or writing to the SNDDAT register.
0x530a	SNDINTE (SNDA2 Interrupt Enable Register)	15-8	–	0x00	–	R	–
		7-2	–	0x00	–	R	–
		1	EMIE	0	H0	R/W	–
		0	EDIE	0	H0	R/W	–

0x5340–0x5342

Charge Pump Regulator (CHREG)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5340	CHREGCONF (CHREG Configuration Register)	15-0	–	0x0030	–	R	Do not change the value.
0x5342	CHREGCTL (CHREG Control Register)	15-9	–	0x00	–	R	–
		8	BSTONBSY	0	H0	R	–
		7-1	–	0x0	–	R	–
		0	BSTON	0	H0	R/W	–

0x5400–0x5412

LCD Driver (LCD4B)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5400	LCD4CLK (LCD4B Clock Control Register)	15-9	–	0x00	–	R	–
		8	DBRUN	1	H0	R/W	–
		7	–	0	–	R	–
		6-4	CLKDIV[2:0]	0x0	H0	R/W	–
		3-2	–	0x0	–	R	–
		1-0	CLKSRC[1:0]	0x0	H0	R/W	–

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5402	LCD4CTL (LCD4B Control Register)	15-8	–	0x00	–	R	–
		7-2	–	0x00	–	R	
		1	LCDDIS	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5404	LCD4TIM1 (LCD4B Timing Control Register 1)	15-12	–	0x0	–	R	–
		11-8	FRMCNT[3:0]	0x3	H0	R/W	
		7-6	–	0x0	–	R	
		5	COMPOS	0	H0	R/W	
		4-3	–	0x0	–	R	
0x5406	LCD4TIM2 (LCD4B Timing Control Register 2)	15-10	–	0x00	–	R	–
		9-8	BSTC[1:0]	0x1	H0	R/W	
		7-3	–	0x00	–	R	
		2-0	NLINE[2:0]	0x0	H0	R/W	
0x5408	LCD4PWR (LCD4B Power Control Register)	15	EXVCSEL	1	H0	R/W	–
		14-13	RESISEL[1:0]	0x0	H0	R/W	
		12-8	LC[4:0]	0x0	H0	R/W	
		7-5	–	0x0	–	R	
		4	BSTEN	0	H0	R/W	
		3	–	0	–	R	
		2	HVLD	0	H0	R/W	
		1	VCSEL	0	H0	R/W	
0x540a	LCD4DSP (LCD4B Display Control Register)	15-8	–	0x00	–	R	–
		7	–	0	–	R	
		6	SEGREV	1	H0	R/W	
		5	COMREV	1	H0	R/W	
		4	DSPREV	1	H0	R/W	
		3	–	0	–	R	
		2	DSPAR	0	H0	R/W	
		1-0	DSPC[1:0]	0x0	H0	R/W	
0x540c	LCD4COMC0 (LCD4B COM Pin Control Register 0)	15-8	–	0x00	–	R	–
		7-4	–	0x0	–	R	
		3	COM3DEN	1	H0	R/W	
		2	COM2DEN	1	H0	R/W	
		1	COM1DEN	1	H0	R/W	
0x5410	LCD4INTF (LCD4B Interrupt Flag Register)	15-8	–	0x00	–	R	–
		7-1	–	0x00	–	R	
		0	FRMIF	0	H0	R/W	
0x5412	LCD4INTE (LCD4B Interrupt Enable Register)	15-8	–	0x00	–	R	–
		7-1	–	0x00	–	R	
		0	FRMIE	0	H0	R/W	

0x5440–0x5450

R/F Converter (RFC2) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5440	RFC0CLK (RFC2 Ch.0 Clock Control Register)	15-9	–	0x00	–	R	–
		8	DBRUN	1	H0	R/W	
		7-6	–	0x0	–	R	
		5-4	CLKDIV[1:0]	0x0	H0	R/W	
		3-2	–	0x0	–	R	
0x5442	RFC0CTL (RFC2 Ch.0 Control Register)	15-9	–	0x00	–	R	–
		8	RFCLKMD	0	H0	R/W	
		7	CONEN	0	H0	R/W	
		6	EVTEN	0	H0	R/W	
		5-4	SMODE[1:0]	0x0	H0	R/W	
		3-1	–	0x0	–	R	
		0	MODEN	0	H0	R/W	

Appendix A List of Peripheral Circuit Control Registers

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5444	RFC0TRG (RFC2 Ch.0 Oscillation Trigger Register)	15-8	–	0x00	–	R	–
		7-3	–	0x00	–	R	
		2	SSENB	0	H0	R/W	
		1	SSENA	0	H0	R/W	
		0	SREF	0	H0	R/W	
0x5446	RFC0MCL (RFC2 Ch.0 Measurement Counter Low Register)	15-0	MC[15:0]	0x0000	H0	R/W	–
0x5448	RFC0MCH (RFC2 Ch.0 Measurement Counter High Register)	15-8	–	0x00	–	R	–
		7-0	MC[23:16]	0x00	H0	R/W	
0x544a	RFC0TCL (RFC2 Ch.n Time Base Counter Low Register)	15-0	TC[15:0]	0x0000	H0	R/W	–
0x544c	RFC0TCH (RFC2 Ch.n Time Base Counter High Register)	15-8	–	0x00	–	R	–
		7-0	TC[23:16]	0x00	H0	R/W	
0x544e	RFC0INTF (RFC2 Ch.0 Interrupt Flag Register)	15-8	–	0x00	–	R	Cleared by writing 1.
		7-5	–	0x0	–	R	
		4	OVTCIF	0	H0	R/W	
		3	OVMCIF	0	H0	R/W	
		2	ESENBIF	0	H0	R/W	
		1	ESENAIF	0	H0	R/W	
0x5450	RFC0INTE (RFC2 Ch.0 Interrupt Enable Register)	15-8	–	0x00	–	R	–
		7-5	–	0x0	–	R	
		4	OVTCIE	0	H0	R/W	
		3	OVMCIE	0	H0	R/W	
		2	ESENBIE	0	H0	R/W	
		1	ESENAIE	0	H0	R/W	
0	EREFIE	0	H0	R/W			

0xffff90

Debugger (DBG)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0xffff90	DBRAM (Debug RAM Base Register)	31-24	–	0x00	–	R	–
		23-0	DBRAM[23:0]	0x00 07c0	H0	R	

Appendix B Power Saving

Current consumption will vary dramatically, depending on CPU operating mode, operation clock frequency, peripheral circuits being operated, and power generator operating mode. Listed below are the control methods for saving power.

B.1 Operating Status Configuration Examples for Power Saving

Table B.1.1 lists typical examples of operating status configuration with consideration given to power saving.

Table B.1.1 Typical Operating Status Configuration Examples

Operating status configuration	Current consumption	V _{D1}	OSC1	IOSC/OSC3/EXOSC	CPU	Current consumption listed in electrical characteristics
Standby	Low ↑	Economy	OFF	OFF	SLEEP	I _{SLEEP}
Low-speed processing					OSC1 RUN	I _{RUN20}
Peripheral circuit operations	↓ High	Normal	ON	ON	SLEEP or HALT	I _{HALT1}
High-speed processing					IOSC/OSC3/EXOSC RUN	I _{RUN10}

If the current consumption order by the operating status configuration shown in Table B.1.1 is different from one that is listed in “Electrical Characteristics,” check the settings shown below.

PWGVD1CTL.REGMODE[1:0] bits of the power generator

If the PWGVD1CTL.REGMODE[1:0] bits of the power generator is 0x2 (normal mode) when the CPU enters SLEEP mode, current consumption in SLEEP mode will be larger than I_{SLEEP} that is listed in “Electrical Characteristics.” Set the PWGVD1CTL.REGMODE[1:0] bits to 0x3 (economy mode) or 0x0 (automatic mode) before executing the slp instruction.

CLGOSC.IOSCSLPC/OSC1SLPC/OSC3SLPC/EXOSCSLPC bits of the clock generator

Setting the CLGOSC.IOSCSLPC, OSC1SLPC, OSC3SLPC, or EXOSCSLPC bit of the clock generator to 0 disables the oscillator circuit stop control when the slp instruction is executed. To stop the oscillator circuits during SLEEP mode, set these bits to 1.

MODEN bits of the peripheral circuits

Setting the MODEN bit of each peripheral circuit to 1 starts supplying the operating clock enabling the peripheral circuit to operate. To reduce current consumption, set the MODEN bits of unnecessary peripheral circuits to 0.

B.2 Other Power Saving Methods

Supply voltage detector configuration

Continuous operation mode (SVDCTL.SVDMD[1:0] bits = 0x0) always detects the power supply voltage, therefore, it increases current consumption. Set the supply voltage detector to intermittent operation mode or turn it on only when required.

LCD driver configurations

- Setting the LCD voltage regulator to operate with the V_{C1} reference voltage (LCD4PWR.VCSEL bit = 0) increases current consumption. If a desired LCD drive voltage can be obtained, operating with V_{C2} reference voltage (LCD4PWR.VCSEL bit = 1) is recommended.
- The lower booster clock frequency setting (LCD4TIM2.BSTC[1:0] bits) for the LCD voltage booster decreases current consumption. Note, however, that the load characteristic becomes worse.
- Setting the LCD voltage regulator into heavy load protection mode (LCD4PWR.HVLD bit = 1) increases current consumption. Heavy load protection mode should be set only when the display becomes unstable.

Appendix C Mounting Precautions

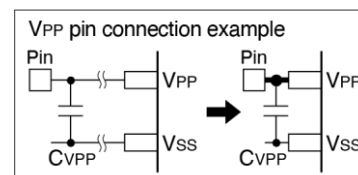
This section describes various precautions for circuit board design and IC mounting.

#RESET pin

Components such as a switch and resistor connected to the #RESET pin should have the shortest connections possible to prevent noise-induced resets.

V_{PP} pin

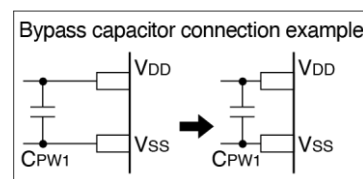
Connect a capacitor C_{VPP} between the V_{SS} and V_{PP} pins to suppress fluctuations within $V_{PP} \pm 1$ V. The C_{VPP} should be placed as close to the V_{PP} pin as possible and use a sufficiently thick wiring pattern that allows current of several tens of mA to flow.



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 V_{DD} and V_{SS} pins should be implemented via the shortest, thickest patterns possible.
- (2) If a bypass capacitor is connected between V_{DD} and V_{SS} , connections between the V_{DD} and V_{SS} pins should be as short as possible.



Signal line location

- (1) To prevent electromagnetically-induced noise arising from mutual induction, large-current signal lines should not be positioned close to pins susceptible to noise, such as oscillator and analog measurement pins.
- (2) 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.

Unused pins

- (1) I/O port (P) pins
Unused pins should be left open. The control registers should be fixed at the initial status.
- (2) V_{C1-3} , C_{P1-2} , $SEGx$, and $COMx$ pins
If the LCD driver is not used, the V_{C1-3} pins should be left open. The C_{P1-2} pins should be configured as general-purpose I/O ports. The control registers should be fixed at the initial status (display off). The unused $SEGx$ and $COMx$ pins that are not required to connect with the LCD panel should be configured as a general-purpose/peripheral circuit I/O port even if the LCD driver is used.

Miscellaneous

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 Measures Against Noise

To improve noise immunity, take measures against noise as follows:

Noise Measures for V_{DD} and V_{SS} Power Supply Pins

When noise falling below the rated voltage is input, an IC malfunction may occur. If desired operations cannot be achieved, take measures against noise on the circuit board, such as designing close patterns for circuit board power supply circuits, adding noise-filtering decoupling capacitors, and adding surge/noise prevention components on the power supply line.

For the recommended patterns on the circuit board, see “Mounting Precautions” in Appendix.

Noise Measures for #RESET Pin

If noise is input to the #RESET pin, the IC may be reset. Therefore, the circuit board must be designed properly taking noise measures into consideration.

For the recommended patterns on the circuit board, see “Mounting Precautions” in Appendix.

Noise Measures for Debug Pins

This product provides the input/output pins (DCLK, DST2, and DSIO) to connect ICDmini (S5U1C17001H) for debugging. If noise is input to these pins with the debugging function enabled, the S1C17 Core may enter DEBUG mode. To prevent unexpected transitions to DEBUG mode caused by extraneous noise, switch the DCLK, DST2, and DSIO pins to general-purpose I/O port pins within the initialization routine when the debug functions are not used.

For details of the pin functions and the function switch control, see the “I/O Ports” chapter.

Note: Do not perform the function switching shown above when the application is under development, as the debug functions must be used. The debugging cannot be performed after the pin function is switched. The above processing must be added after the application development has completed and debugging is no longer necessary.

The DSIO pin should be pulled up with a 10 k Ω resistor when using the debug pin functions.

Noise Measures for Interrupt Input Pins

This product is able to generate a port input interrupt when the input signal changes. The interrupt is generated when an input signal edge is detected, therefore, an interrupt may occur if the signal changes due to extraneous noise. To prevent occurrence of unexpected interrupts due to extraneous noise, enable the chattering filter circuit when using the port input interrupt.

For details of the port input interrupt and chattering filter circuit, see the “I/O Ports” chapter.

Noise Measures for UART Pins

This product includes a UART for asynchronous communications. The UART starts receive operation when it detects a low level input from the SIN n pin. Therefore, a receive operation may be started if the SIN n pin is set to low due to extraneous noise. In this case, a receive error will occur or invalid data will be received.

To prevent the UART from malfunction caused by extraneous noise, take the following measures:

- Stop the UART operations while asynchronous communication is not performed.
- Execute the resending process via software after executing the receive error handler with a parity check.

For details of the pin functions and the function switch control, see the “I/O Ports” chapter. For the UART control and details of receive errors, see the “UART” chapter.

Appendix E Initialization Routine

The following lists typical vector tables and initialization routines:

boot.s

```
.org 0x8000
.section .rodata                                     ...(1)

; =====
; Vector table
; =====

; interrupt vector interrupt
; number offset source

.long BOOT ; 0x00 0x00 reset                               ...(2)
.long unalign_handler ; 0x01 0x04 unalign
.long nmi_handler ; 0x02 0x08 NMI
.long int03_handler ; 0x03 0x0c -
.long svd5_handler ; 0x04 0x00 SVD5
.long pport_handler ; 0x05 0x14 PPORT
.long int06_handler ; 0x06 0x18 -
.long clg_handler ; 0x07 0x1c CLG
.long int08_handler ; 0x08 0x20 -
.long t16_0_handler ; 0x09 0x24 T16 ch0
.long uart3_0_handler ; 0x0a 0x28 UART3 ch0
.long t16_1_handler ; 0x0b 0x2c T16 ch1
.long spia_0_handler ; 0x0c 0x30 SPIA ch0
.long i2c_handler ; 0x0d 0x34 I2C
.long t16b_0_handler ; 0x0e 0x38 T16B ch0
.long t16b_1_handler ; 0x0f 0x3c T16B ch1
.long int10_handler ; 0x10 0x40 -
.long t16_2_handler ; 0x11 0x44 T16 ch2
.long snda2_handler ; 0x12 0x48 SNDA2
.long lcd4b_handler ; 0x13 0x4c LCD4B
.long rfc2_0_handler ; 0x14 0x50 RFC2 ch0
.long eepromc_handler ; 0x15 0x54 EEPROMC
.long int16_handler ; 0x16 0x58 -
.long int17_handler ; 0x17 0x5c -
.long int18_handler ; 0x18 0x60 -
.long int19_handler ; 0x19 0x64 -
.long int1a_handler ; 0x1a 0x68 -
.long int1b_handler ; 0x1b 0x6c -
.long int1c_handler ; 0x1c 0x70 -
.long int1d_handler ; 0x1d 0x74 -
.long int1e_handler ; 0x1e 0x78 -
.long int1f_handler ; 0x1f 0x7c -

; =====
; Program code
; =====

.text                                               ...(3)
.align 1
```

Appendix E Initialization Routine

```
BOOT:
; ===== Initialize =====

; ----- Stack pointer -----
Xld.a  %sp, 0x7c0                                ...(4)

; ----- Memory controller -----
Xld.a  %r2, 0x4000    ; Misc register address
Xld.a  %r0, 0x96
ld.b   [%r2], %r0    ; Disable system protection

; Flash read wait cycle
Xld.a  %r1, 0x41b0    ; FLASHC register address
Xld.a  %r0, 0x00      ; 0x00 = No wait
ld.b   [%r1], %r0    ; [0x41b0] <= 0x00                ...(5)

ld.b   [%r2], %r0    ; Enable system protection

; ===== Main routine =====
...

; =====
;   Interrupt handler
; =====

; ----- Address unalign -----
unalign_handler:
...

; ----- NMI -----
nmi_handler:
...
```

-
- (1) A “.rodata” section is declared to locate the vector table in the “.vector” section.
 - (2) Interrupt handler routine addresses are defined as vectors.
“intXX_handler” can be used for software interrupts.
 - (3) The program code is written in the “.text” section.
 - (4) Sets the stack pointer.
 - (5) Sets the number of Flash memory read cycles.
(See the “Memory and Bus” chapter.)

Revision History

Code No.	Page	Contents
414766900	All	New establishment
414766901	14-4	<p>14.4.1 Initialization Initial settings for comparator mode (Old)</p> <p>4. Set the T16BnMC register. (Set MAX counter data)</p> <p>5. Set the T16BnCCR0 and T16BnCCR1 registers. (Set the counter comparison value)</p> <p>6. Set the following bits when using the interrupt:</p> <ul style="list-style-type: none"> - Write 1 to the interrupt flags in the T16BnINTF register. (Clear interrupt flags) - Set the interrupt enable bits in the T16BnINTE register to 1. (Enable interrupts) <p>7. Set the following T16BnCTL register bits:</p> <ul style="list-style-type: none"> - T16BnCTL.CNTMD[1:0] bits (Select count up/down operation) - T16BnCTL.ONEST bit (Select one-shot/repeat operation) - Set the T16BnCTL.PRESET bit to 1. (Reset counter) - Set the T16BnCTL.RUN bit to 1. (Start counting) <p>(New)</p> <p><u>4.</u> Check to see if the T16BnCTL.MAXBSY bit is set to 0.</p> <p><u>5.</u> Set the T16BnCCR0 and T16BnCCR1 registers. (Set the counter comparison value)</p> <p><u>6.</u> Set the T16BnMC register. (Set MAX counter data)</p> <p><u>7.</u> Set the following bits when using the interrupt:</p> <ul style="list-style-type: none"> - Write 1 to the interrupt flags in the T16BnINTF register. (Clear interrupt flags) - Set the interrupt enable bits in the T16BnINTE register to 1. (Enable interrupts) <p><u>8.</u> Check to see if the T16BnCTL.MAXBSY and the T16BnCTL.RUN bit are set to 0.</p> <p><u>9.</u> Set the following T16BnCTL register bits:</p> <ul style="list-style-type: none"> - T16BnCTL.CNTMD[1:0] bits (Select count up/down operation) - T16BnCTL.ONEST bit (Select one-shot/repeat operation) - Set the T16BnCTL.PRESET bit to 1. (Reset counter) - Set the T16BnCTL.RUN bit to 1. (Start counting)
		<p>14.4.1 Initialization Initial settings for capture mode (Old)</p> <p>4. Set the T16BnMC register. (Set MAX counter data)</p> <p>5. Set the following bits when using the interrupt:</p> <ul style="list-style-type: none"> - Write 1 to the interrupt flags in the T16BnINTF register. (Clear interrupt flags) - Set the interrupt enable bits in the T16BnINTE register to 1. (Enable interrupts) <p>6. Set the following T16BnCTL register bits:</p> <ul style="list-style-type: none"> - T16BnCTL.CNTMD[1:0] bits (Select count up/down operation) - T16BnCTL.ONEST bit (Select one-shot/repeat operation) - Set the T16BnCTL.PRESET bit to 1. (Reset counter) - Set the T16BnCTL.RUN bit to 1. (Start counting) <p>(New)</p> <p><u>4.</u> Check to see if the T16BnCTL.MAXBSY bit is set to 0.</p> <p><u>5.</u> Set the T16BnMC register. (Set MAX counter data)</p> <p><u>6.</u> Set the following bits when using the interrupt:</p> <ul style="list-style-type: none"> - Write 1 to the interrupt flags in the T16BnINTF register. (Clear interrupt flags) - Set the interrupt enable bits in the T16BnINTE register to 1. (Enable interrupts) <p><u>7.</u> Check to see if the T16BnCTL.MAXBSY and the T16BnCTL.RUN bit are set to 0.</p> <p><u>8.</u> Set the following T16BnCTL register bits:</p> <ul style="list-style-type: none"> - T16BnCTL.CNTMD[1:0] bits (Select count up/down operation)

Appendix E Initialization Routine

		<ul style="list-style-type: none">- T16BnCTL.ONEST bit (Select one-shot/repeat operation)- Set the T16BnCTL.PRESET bit to 1. (Reset counter)- Set the T16BnCTL.RUN bit to 1. (Start counting)
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

America

Epson America, Inc.
3131 Katella Ave.
Los Alamitos, CA 90720, USA
Phone: +1-800-463-7766

Europe

Epson Europe Electronics GmbH
Riesstrasse 15, 80992 Munich,
Germany
Phone: +49-89-14005-0

Asia

Epson (China) Co., Ltd.
4th Floor, Tower 1 of China Central Place, 81 Jianguo Road, Chaoyang
District, Beijing 100025 China
Phone: +86-10-8522-1199

Shanghai Branch

Room 601-603, Building A One East, No.325 East Longhua Road,
Shanghai 200023, China
Phone: +86-21-5330-4888

Shenzhen Branch

Room 903, Floor 9, T2 Tower, Ali Center,
No.3331 Keyuan South RD, Nanshan District, Shenzhen Guangdong
518054, China
Phone: +86-755-3299-0588

Epson Taiwan Technology & Trading Ltd.

15F, No.100, Songren Rd, Sinyi Dist, Taipei City 110. Taiwan
Phone: +886-2-8786-6688

Epson Singapore Pte., Ltd.

438B Alexandra Road,
Block B Alexandra TechnoPark, #04-01/04, Singapore 119968
Phone: +65-6586-5500

Epson Korea Co.,Ltd

10F Posco Tower Yeoksam, Teheranno 134 Gangnam-gu,
Seoul, 06235, Korea
Phone: +82-2-3420-6695

Seiko Epson Corp.

Sales & Marketing Division

MD Sales & Marketing Department

421-8 Hino, Hino-shi, Tokyo 191-8501, Japan