

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17M20/M21/M22/M23/M24/M25

Technical Manual

Evaluation board/kit and Development tool important notice

1. This 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. This evaluation board/kit or development tool is intended for use by an electronics engineer and is not a consumer product. The user should use it properly and in a safe manner. Seiko 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 this evaluation board/kit or development tool may be changed without any notice.

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. When exporting the products or technology described in this material, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You are requested not to use, to resell, to export and/or to otherwise dispose of the products (and any technical information furnished, if any) for the development and/or manufacture of weapon of mass destruction or for other military purposes.

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

Preface

This is a technical manual for designers and programmers who develop a product using the S1C17M20/M21/M22/M23/M24/M25. 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_*n*CTL register of the 16-bit timer

If one channel is implemented (Ch.0 only): T16_*n*CTL = T16_0CTL only

If two channels are implemented (Ch.0 and Ch.1): T16_*n*CTL = 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.

– Contents –

Preface	i
Notational conventions and symbols in this manual	i
1 Overview.....	1-1
1.1 Features	1-1
1.2 Block Diagram.....	1-3
1.3 Pins	1-4
1.3.1 S1C17M20/M23 Pin Configuration Diagram	1-4
1.3.2 S1C17M21/M24 Pin Configuration Diagram	1-6
1.3.3 S1C17M22/M25 Pin Configuration Diagram	1-7
1.3.4 Pin Descriptions.....	1-8
2 Power Supply, Reset, and Clocks	2-1
2.1 Power Generator (PWG).....	2-1
2.1.1 Overview	2-1
2.1.2 Pins.....	2-1
2.1.3 V _{D1} Regulator Operation Mode.....	2-1
2.2 System Reset Controller (SRC).....	2-2
2.2.1 Overview	2-2
2.2.2 Input Pin.....	2-2
2.2.3 Reset Sources	2-3
2.2.4 Initialization Conditions (Reset Groups).....	2-3
2.3 Clock Generator (CLG).....	2-4
2.3.1 Overview	2-4
2.3.2 Input/Output Pins	2-5
2.3.3 Clock Sources	2-5
2.3.4 Operations	2-8
2.4 Operating Mode	2-12
2.4.1 Initial Boot Sequence.....	2-12
2.4.2 Transition between Operating Modes.....	2-12
2.5 Interrupts.....	2-14
2.6 Control Registers	2-14
PWG V _{D1} Regulator Control Register	2-14
CLG System Clock Control Register.....	2-15
CLG Oscillation Control Register	2-16
CLG OSC1 Control Register	2-17
CLG OSC3 Control Register	2-18
CLG Interrupt Flag Register	2-19
CLG Interrupt Enable Register	2-20
CLG FOUT Control Register.....	2-21
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	3-3
3.3.3 List of Debugger Input/Output Pins.....	3-3

3.3.4 External Connection	3-3
3.3.5 Flash Security Function	3-4
3.4 Control Register	3-4
MISC PSR Register	3-4
Debug RAM Base Register	3-5
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 RAM	4-3
4.5 Peripheral Circuit Control Registers	4-3
4.5.1 System-Protect Function	4-8
4.6 Control Registers	4-8
MISC System Protect Register	4-8
MISC IRAM Size Register	4-8
FLASHC Flash Read Cycle Register	4-8
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)	5-3
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	5-4
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	5-5
5.8 Control Registers	5-5
MISC Vector Table Address Low Register	5-5
MISC Vector Table Address High Register	5-5
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	6-3
6.2.2 Over Voltage Tolerant Fail-Safe Type I/O Cell	6-3
6.2.3 Pull-Up/Pull-Down	6-3
6.2.4 CMOS Output and High Impedance State	6-3
6.3 Clock Settings	6-3
6.3.1 PPORT Operating Clock	6-3
6.3.2 Clock Supply in SLEEP Mode	6-4
6.3.3 Clock Supply in DEBUG Mode	6-4
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	6-7
	Px Port Data Register	6-7
	Px Port Enable Register	6-7
	Px Port Pull-up/down Control Register	6-8
	Px Port Interrupt Flag Register	6-8
	Px Port Interrupt Control Register	6-8
	Px Port Chattering Filter Enable Register	6-9
	Px Port Mode Select Register	6-9
	Px Port Function Select Register	6-9
	P Port Clock Control Register	6-10
	P Port Interrupt Flag Group Register	6-11
6.7	Control Register and Port Function Configuration of this IC	6-12
6.7.1	P0 Port Group	6-12
6.7.2	P1 Port Group	6-14
6.7.3	P2 Port Group	6-17
6.7.4	P3 Port Group	6-19
6.7.5	P4 Port Group	6-21
6.7.6	Pd Port Group	6-23
6.7.7	Common Registers between Port Groups	6-24
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	Real-Time Clock (RTCA)	9-1
9.1	Overview	9-1
9.2	Output Pin and External Connection	9-1
	9.2.1 Output Pin	9-1
9.3	Clock Settings	9-2
	9.3.1 RTCA Operating Clock	9-2
	9.3.2 Theoretical Regulation Function	9-2
9.4	Operations	9-3
	9.4.1 RTCA Control	9-3
	9.4.2 Real-Time Clock Counter Operations	9-4
	9.4.3 Stopwatch Control	9-4
	9.4.4 Stopwatch Count-up Pattern	9-4
9.5	Interrupts	9-5
9.6	Control Registers	9-6
	RTC Control Register	9-6
	RTC Second Alarm Register	9-7

RTC Hour/Minute Alarm Register	9-8
RTC Stopwatch Control Register	9-8
RTC Second/1Hz Register	9-9
RTC Hour/Minute Register	9-10
RTC Month/Day Register	9-11
RTC Year/Week Register	9-11
RTC Interrupt Flag Register	9-12
RTC Interrupt Enable Register	9-13
10 Supply Voltage Detector (SVD3)	10-1
10.1 Overview	10-1
10.2 Input Pins and External Connection	10-2
10.2.1 Input Pins	10-2
10.2.2 External Connection	10-2
10.3 Clock Settings	10-2
10.3.1 SVD3 Operating Clock	10-2
10.3.2 Clock Supply in SLEEP Mode	10-2
10.3.3 Clock Supply in DEBUG Mode	10-3
10.4 Operations	10-3
10.4.1 SVD3 Control	10-3
10.4.2 SVD3 Operations	10-4
10.5 SVD3 Interrupt and Reset	10-4
10.5.1 SVD3 Interrupt	10-4
10.5.2 SVD3 Reset	10-5
10.6 Control Registers	10-5
SVD3 Clock Control Register	10-5
SVD3 Control Register	10-6
SVD3 Status and Interrupt Flag Register	10-7
SVD3 Interrupt Enable Register	10-8
11 16-bit Timers (T16)	11-1
11.1 Overview	11-1
11.2 Input Pin	11-1
11.3 Clock Settings	11-2
11.3.1 T16 Operating Clock	11-2
11.3.2 Clock Supply in SLEEP Mode	11-2
11.3.3 Clock Supply in DEBUG Mode	11-2
11.3.4 Event Counter Clock	11-2
11.4 Operations	11-2
11.4.1 Initialization	11-2
11.4.2 Counter Underflow	11-3
11.4.3 Operations in Repeat Mode	11-3
11.4.4 Operations in One-shot Mode	11-3
11.4.5 Counter Value Read	11-4
11.5 Interrupt	11-4
11.6 Control Registers	11-4
T16 Ch. <i>n</i> Clock Control Register	11-4
T16 Ch. <i>n</i> Mode Register	11-5
T16 Ch. <i>n</i> Control Register	11-5
T16 Ch. <i>n</i> Reload Data Register	11-6
T16 Ch. <i>n</i> Counter Data Register	11-6
T16 Ch. <i>n</i> Interrupt Flag Register	11-6
T16 Ch. <i>n</i> Interrupt Enable Register	11-7

12 UART (UART3).....	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 Input Pin Pull-Up Function.....	12-2
12.2.4 Output Pin Open-Drain Output Function	12-2
12.2.5 Input/Output Signal Inverting Function.....	12-2
12.3 Clock Settings.....	12-2
12.3.1 UART3 Operating Clock	12-2
12.3.2 Clock Supply in SLEEP Mode	12-3
12.3.3 Clock Supply in DEBUG Mode.....	12-3
12.3.4 Baud Rate Generator.....	12-3
12.4 Data Format	12-3
12.5 Operations	12-4
12.5.1 Initialization	12-4
12.5.2 Data Transmission	12-5
12.5.3 Data Reception	12-6
12.5.4 IrDA Interface.....	12-7
12.5.5 Carrier Modulation	12-7
12.6 Receive Errors.....	12-8
12.6.1 Framing Error	12-8
12.6.2 Parity Error	12-8
12.6.3 Overrun Error	12-9
12.7 Interrupts.....	12-9
12.8 Control Registers	12-9
UART3 Ch. <i>n</i> Clock Control Register	12-9
UART3 Ch. <i>n</i> Mode Register	12-10
UART3 Ch. <i>n</i> Baud-Rate Register	12-11
UART3 Ch. <i>n</i> Control Register	12-12
UART3 Ch. <i>n</i> Transmit Data Register	12-12
UART3 Ch. <i>n</i> Receive Data Register.....	12-12
UART3 Ch. <i>n</i> Status and Interrupt Flag Register	12-13
UART3 Ch. <i>n</i> Interrupt Enable Register.....	12-14
UART3 Ch. <i>n</i> Carrier Waveform Register	12-14
13 Synchronous Serial Interface (SPIA).....	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
13.2.3 Pin Functions in Master Mode and Slave Mode.....	13-3
13.2.4 Input Pin Pull-Up/Pull-Down Function	13-3
13.3 Clock Settings.....	13-3
13.3.1 SPIA Operating Clock.....	13-3
13.3.2 Clock Supply in DEBUG Mode.....	13-4
13.3.3 SPI Clock (SPICLK _{<i>n</i>}) Phase and Polarity	13-4
13.4 Data Format	13-5
13.5 Operations	13-5
13.5.1 Initialization	13-5
13.5.2 Data Transmission in Master Mode	13-5
13.5.3 Data Reception in Master Mode.....	13-7
13.5.4 Terminating Data Transfer in Master Mode.....	13-8

13.5.5 Data Transfer in Slave Mode.....	13-8
13.5.6 Terminating Data Transfer in Slave Mode	13-10
13.6 Interrupts.....	13-10
13.7 Control Registers	13-11
SPIA Ch. <i>n</i> Mode Register	13-11
SPIA Ch. <i>n</i> Control Register.....	13-12
SPIA Ch. <i>n</i> Transmit Data Register	13-13
SPIA Ch. <i>n</i> Receive Data Register	13-13
SPIA Ch. <i>n</i> Interrupt Flag Register	13-13
SPIA Ch. <i>n</i> Interrupt Enable Register	13-14
14 I²C (I2C).....	14-1
14.1 Overview	14-1
14.2 Input/Output Pins and External Connections	14-2
14.2.1 List of Input/Output Pins.....	14-2
14.2.2 External Connections	14-2
14.3 Clock Settings.....	14-3
14.3.1 I2C Operating Clock	14-3
14.3.2 Clock Supply in DEBUG Mode.....	14-3
14.3.3 Baud Rate Generator.....	14-3
14.4 Operations	14-4
14.4.1 Initialization	14-4
14.4.2 Data Transmission in Master Mode	14-5
14.4.3 Data Reception in Master Mode.....	14-7
14.4.4 10-bit Addressing in Master Mode	14-9
14.4.5 Data Transmission in Slave Mode.....	14-10
14.4.6 Data Reception in Slave Mode	14-12
14.4.7 Slave Operations in 10-bit Address Mode.....	14-14
14.4.8 Automatic Bus Clearing Operation	14-14
14.4.9 Error Detection.....	14-15
14.5 Interrupts.....	14-16
14.6 Control Registers	14-17
I2C Ch. <i>n</i> Clock Control Register	14-17
I2C Ch. <i>n</i> Mode Register.....	14-18
I2C Ch. <i>n</i> Baud-Rate Register.....	14-18
I2C Ch. <i>n</i> Own Address Register	14-18
I2C Ch. <i>n</i> Control Register	14-19
I2C Ch. <i>n</i> Transmit Data Register	14-20
I2C Ch. <i>n</i> Receive Data Register.....	14-20
I2C Ch. <i>n</i> Status and Interrupt Flag Register	14-20
I2C Ch. <i>n</i> Interrupt Enable Register	14-21
15 16-bit PWM Timers (T16B)	15-1
15.1 Overview	15-1
15.2 Input/Output Pins.....	15-2
15.3 Clock Settings.....	15-3
15.3.1 T16B Operating Clock	15-3
15.3.2 Clock Supply in SLEEP Mode	15-3
15.3.3 Clock Supply in DEBUG Mode.....	15-3
15.3.4 Event Counter Clock.....	15-3
15.4 Operations	15-4
15.4.1 Initialization	15-4
15.4.2 Counter Block Operations	15-5
15.4.3 Comparator/Capture Block Operations.....	15-8
15.4.4 TOUT Output Control	15-16

15.5	Interrupt.....	15-22
15.6	Control Registers	15-22
	T16B Ch. <i>n</i> Clock Control Register	15-22
	T16B Ch. <i>n</i> Counter Control Register	15-23
	T16B Ch. <i>n</i> Max Counter Data Register.....	15-24
	T16B Ch. <i>n</i> Timer Counter Data Register.....	15-24
	T16B Ch. <i>n</i> Counter Status Register.....	15-25
	T16B Ch. <i>n</i> Interrupt Flag Register.....	15-26
	T16B Ch. <i>n</i> Interrupt Enable Register	15-27
	T16B Ch. <i>n</i> Comparator/Capture <i>m</i> Control Register.....	15-28
	T16B Ch. <i>n</i> Compare/Capture <i>m</i> Data Register.....	15-30
16	Sound Generator (SNDA)	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 Output Pin Drive Mode	16-2
	16.2.3 External Connections	16-2
16.3	Clock Settings.....	16-3
	16.3.1 SNDA Operating Clock	16-3
	16.3.2 Clock Supply in SLEEP Mode	16-3
	16.3.3 Clock Supply in DEBUG Mode.....	16-3
16.4	Operations	16-3
	16.4.1 Initialization	16-3
	16.4.2 Buzzer Output in Normal Buzzer Mode.....	16-3
	16.4.3 Buzzer Output in One-shot Buzzer Mode.....	16-6
	16.4.4 Output in Melody Mode.....	16-7
16.5	Interrupts.....	16-9
16.6	Control Registers	16-9
	SNDA Clock Control Register	16-9
	SNDA Select Register	16-10
	SNDA Control Register.....	16-11
	SNDA Data Register.....	16-11
	SNDA Interrupt Flag Register.....	16-12
	SNDA Interrupt Enable Register.....	16-13
17	IR Remote Controller (REMC3)	17-1
17.1	Overview	17-1
17.2	Input/Output Pins and External Connections	17-1
	17.2.1 Output Pin.....	17-1
	17.2.2 External Connections	17-2
17.3	Clock Settings.....	17-2
	17.3.1 REMC3 Operating Clock	17-2
	17.3.2 Clock Supply in SLEEP Mode	17-2
	17.3.3 Clock Supply in DEBUG Mode.....	17-2
17.4	Operations	17-2
	17.4.1 Initialization	17-2
	17.4.2 Data Transmission Procedures	17-3
	17.4.3 REMO Output Waveform	17-3
	17.4.4 Continuous Data Transmission and Compare Buffers.....	17-5
17.5	Interrupts.....	17-6
17.6	Application Example: Driving EL Lamp.....	17-7
17.7	Control Registers	17-7
	REMC3 Clock Control Register.....	17-7
	REMC3 Data Bit Counter Control Register.....	17-8

REMC3 Data Bit Counter Register	17-9
REMC3 Data Bit Active Pulse Length Register	17-10
REMC3 Data Bit Length Register	17-10
REMC3 Status and Interrupt Flag Register	17-10
REMC3 Interrupt Enable Register	17-11
REMC3 Carrier Waveform Register	17-11
REMC3 Carrier Modulation Control Register	17-12
18 R/F Converter (RFC)	18-1
18.1 Overview	18-1
18.2 Input/Output Pins and External Connections	18-2
18.2.1 List of Input/Output Pins	18-2
18.2.2 External Connections	18-2
18.3 Clock Settings	18-3
18.3.1 RFC Operating Clock	18-3
18.3.2 Clock Supply in SLEEP Mode	18-3
18.3.3 Clock Supply in DEBUG Mode	18-3
18.4 Operations	18-3
18.4.1 Initialization	18-3
18.4.2 Operating Modes	18-4
18.4.3 RFC Counters	18-4
18.4.4 Converting Operations and Control Procedure	18-5
18.4.5 CR Oscillation Frequency Monitoring Function	18-7
18.5 Interrupts	18-7
18.6 Control Registers	18-8
RFC Ch. <i>n</i> Clock Control Register	18-8
RFC Ch. <i>n</i> Control Register	18-8
RFC Ch. <i>n</i> Oscillation Trigger Register	18-9
RFC Ch. <i>n</i> Measurement Counter Low and High Registers	18-10
RFC Ch. <i>n</i> Time Base Counter Low and High Registers	18-10
RFC Ch. <i>n</i> Interrupt Flag Register	18-11
RFC Ch. <i>n</i> Interrupt Enable Register	18-11
19 12-bit A/D Converter (ADC12A)	19-1
19.1 Overview	19-1
19.2 Input Pins and External Connections	19-2
19.2.1 List of Input Pins	19-2
19.2.2 External Connections	19-2
19.3 Clock Settings	19-2
19.3.1 ADC12A Operating Clock	19-2
19.3.2 Sampling Time	19-2
19.4 Operations	19-3
19.4.1 Initialization	19-3
19.4.2 Conversion Start Trigger Source	19-3
19.4.3 Conversion Mode and Analog Input Pin Settings	19-4
19.4.4 A/D Conversion Operations and Control Procedures	19-4
19.5 Interrupts	19-6
19.6 Control Registers	19-6
ADC12A Ch. <i>n</i> Control Register	19-6
ADC12A Ch. <i>n</i> Trigger/Analog Input Select Register	19-7
ADC12A Ch. <i>n</i> Configuration Register	19-8
ADC12A Ch. <i>n</i> Interrupt Flag Register	19-9
ADC12A Ch. <i>n</i> Interrupt Enable Register	19-10
ADC12A Ch. <i>n</i> Result Register <i>m</i>	19-10

20 Multiplier/Divider (COPRO2)	20-1
20.1 Overview	20-1
20.2 Operation Mode and Output Mode	20-1
20.3 Multiplication	20-2
20.4 Division	20-3
20.5 MAC	20-5
20.6 Reading Operation Results	20-7
21 Electrical Characteristics	21-1
21.1 Absolute Maximum Ratings	21-1
21.2 Recommended Operating Conditions	21-1
21.3 Current Consumption	21-2
21.4 System Reset Controller (SRC) Characteristics	21-4
21.5 Clock Generator (CLG) Characteristics	21-4
21.6 Flash Memory Characteristics	21-7
21.7 Input/Output Port (PPORT) Characteristics	21-7
21.8 Supply Voltage Detector (SVD3) Characteristics	21-8
21.9 UART (UART3) Characteristics	21-10
21.10 Synchronous Serial Interface (SPIA) Characteristics	21-10
21.11 I ² C (I2C) Characteristics	21-11
21.12 R/F Converter (RFC) Characteristics	21-12
21.13 12-bit A/D Converter (ADC12A) Characteristics	21-13
22 Basic External Connection Diagram	22-1
23 Package	23-1
Appendix A List of Peripheral Circuit Control Registers	AP-A-1
0x4000–0x4008 Misc Registers (MISC)	AP-A-1
0x4020 Power Generator (PWG)	AP-A-1
0x4040–0x4050 Clock Generator (CLG)	AP-A-1
0x4080–0x4094 Interrupt Controller (ITC)	AP-A-2
0x40a0–0x40a4 Watchdog Timer (WDT2)	AP-A-4
0x40c0–0x40d2 Real-time Clock (RTCA)	AP-A-4
0x4100–0x4106 Supply Voltage Detector (SVD3)	AP-A-6
0x4160–0x416c 16-bit Timer (T16) Ch.0	AP-A-6
0x41b0 Flash Controller (FLASHC)	AP-A-7
0x4200–0x42e2 I/O Ports (PPORT)	AP-A-7
0x4300–0x431e Universal Port Multiplexer (UPMUX)	AP-A-17
0x4380–0x4390 UART (UART3) Ch.0	AP-A-19
0x43a0–0x43ac 16-bit Timer (T16) Ch.1	AP-A-20
0x43b0–0x43ba Synchronous Serial Interface (SPIA) Ch.0	AP-A-20
0x43c0–0x43d2 I ² C (I2C) Ch.0	AP-A-21
0x5000–0x501a 16-bit PWM Timer (T16B) Ch.0	AP-A-22
0x5040–0x505a 16-bit PWM Timer (T16B) Ch.1	AP-A-23
0x5200–0x5210 UART (UART3) Ch.1	AP-A-25
0x5260–0x526c 16-bit Timer (T16) Ch.2	AP-A-26
0x5270–0x527a Synchronous Serial Interface (SPIA) Ch.1	AP-A-27
0x5300–0x530a Sound Generator (SNDA)	AP-A-27
0x5320–0x5332 IR Remote Controller (REMC3)	AP-A-28
0x5440–0x5450 R/F Converter (RFC) Ch.0 (S1C17M22/M25)	AP-A-29
0x5460–0x5470 R/F Converter (RFC) Ch.1 (S1C17M22/M25)	AP-A-30
0x5480–0x548c 16-bit Timer (T16) Ch.3	AP-A-31

0x54a0–0x54ba	12-bit A/D Converter (ADC12A).....	AP-A-31
0xffff90	Debugger (DBG)	AP-A-33
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-2
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 S1C17M20/M21/M22/M23/M24/M25 is a 16-bit embedded Flash MCU that features low power consumption. The embedded Flash memory can also be used as an EEPROM emulation data memory via software. The S1C17M20/M21/M22/M23/M24/M25 includes various serial interfaces, an A/D converter, and various timers as well as a high-performance 16-bit CPU. It is suitable for applications that require an A/D conversion function, such as household equipment and FA products.

1.1 Features

Table 1.1.1 Features

Model	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin PKG	32-pin PKG		
CPU				
CPU core	Seiko Epson original 16-bit RISC CPU core S1C17			
Other	On-chip debugger			
Embedded Flash memory				
Capacity	16K bytes (S1C17M20/M21/M22) 32K bytes (S1C17M23/M24/M25)			
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 Flash programming voltage can be generated internally.			
Embedded RAM				
Capacity	2K bytes			
Clock generator (CLG)				
System clock source	4 sources (IOSC/OSC1/OSC3/EXOSC)			
System clock frequency (operating frequency)	21 MHz (max.)			
IOSC oscillator circuit (boot clock source)	700 kHz (typ.) embedded oscillator 23 μs (max.) starting time (time from cancelation of SLEEP state to vector table read by the CPU)			
OSC1 oscillator circuit	–	32.768 kHz (typ.) crystal oscillator		
	32 kHz (typ.) embedded oscillator			
OSC3 oscillator circuit	–	Oscillation stop detection circuit included		
	–	21 MHz (max.) crystal/ceramic oscillator		
	12, 16, and 20 MHz-switchable embedded oscillator			
	–	Auto-trimming function for the embedded oscillator		
EXOSC clock input	21 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	I/O port	17 bits (max.)	23 bits (max.)	39 bits (max.)
	Output port	1 bit (max.)		
	Other	Pins are shared with the peripheral I/O.		
Number of input interrupt ports	15 bits (max.)	19 bits (max.)	35 bits (max.)	
Number of ports that support universal port multiplexer (UPMUX)	15 bits	19 bits	32 bits	
	A peripheral circuit I/O function selected via software can be assigned to each port.			
Timers				
Watchdog timer (WDT2)	Generates NMI or watchdog timer reset. Programmable NMI/reset generation cycle			
Real-time clock (RTCA)	128–1 Hz counter, second/minute/hour/day/day of the week/month/year counters Theoretical regulation function for 1-second correction Alarm and stopwatch functions			
16-bit timer (T16)	4 channels Generates the SPIA master clocks and the ADC12A trigger signal.			
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/channel			
Supply voltage detector (SVD3)				
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	V _{DD} : 28 levels (1.8 to 5.0 V)/external voltage: 32 levels (1.2 to 5.0 V)			
Other	Intermittent operation mode Generates an interrupt or reset according to the detection level evaluation.			

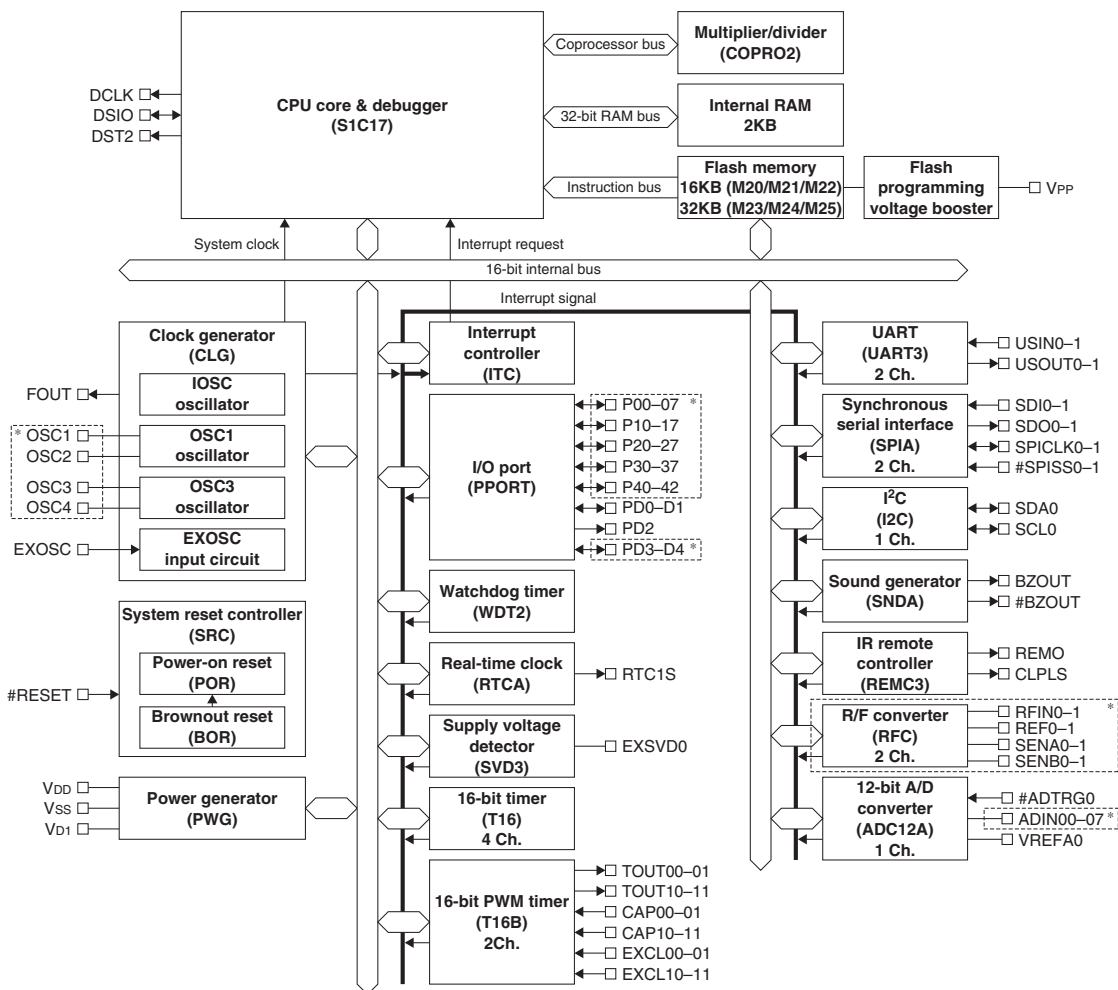
1 OVERVIEW

Model	S1C17M20/M23		S1C17M21/M24		S1C17M22/M25
	24-pin PKG	32-pin PKG			
Serial interfaces					
UART (UART3)	2 channels 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)	2 channels 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 channel Baud-rate generator included				
Sound generator (SNDA)					
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/slur may be specified.				
IR remote controller (REMC3)					
Number of transmitter channels	1 channel				
Other	EL lamp drive waveform can be generated for an application example. Output inversion function				
R/F converter (RFC)					
Conversion method	—				CR oscillation type with 24-bit counters
Number of conversion channels					2 channels (Up to two sensors can be connected to each channel.)
Supported sensors					DC-bias resistive sensors
12-bit A/D converter (ADC12A)					
Conversion method	Successive approximation type				
Resolution	12 bits				
Number of conversion channels	1 channel				
Number of analog signal input ports	4 ports	6 ports			8 ports
Multiplier/divider (COPRO2)					
Arithmetic functions	16-bit × 16-bit multiplier 16-bit × 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.				
Brownout reset	Reset when the power supply voltage drops.				
Key entry reset	Reset when the P00 to P01/P02/P03 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 int.	1 system (8 levels)			19 systems (8 levels)
	Internal int.	17 systems (8 levels)			
Power supply voltage					
V _{DD} operating voltage	1.8 to 5.5 V				
V _{DD} operating voltage for Flash programming	2.4 to 5.5 V (When V _{PP} (7.5 V) is supplied externally) 2.4 to 5.5 V (When V _{PP} is generated internally)				
Operating temperature					
Operating temperature range	-40 to 85°C				
Current consumption (typ. value)					
SLEEP mode	0.36 μA I _{OSC} = OFF, OSC1 = OFF, OSC3 = OFF				
HALT mode	0.7 μA OSC1 = 32.768 kHz (crystal oscillator), RTC = ON				
RUN mode	5 μA OSC1 = 32.768 kHz (crystal oscillator), RTC = ON, CPU = OSC1				
	160 μA				
	OSC3 = 1 MHz (ceramic oscillator), OSC1 = 32.768 kHz (crystal oscillator), RTC = ON, CPU = OSC3				

Model	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin PKG	32-pin PKG		
Shipping form				
Package	*1 SQFN4-24PIN (P-VQFN024-0404-0.50, 4 × 4 mm, t = 1 mm, 0.5 mm pitch)	SQFN5-32PIN (P-VQFN032-0505-0.50, 5 × 5 mm, t = 1 mm, 0.5 mm pitch)	TQFP12-32PIN (P-TQFP032-0707-0.80, 7 × 7 mm, t = 1.2 mm, 0.8 mm pitch)	TQFP12-48PIN (P-TQFP048-0707-0.50, 7 × 7 mm, t = 1.2 mm, 0.5 mm pitch)

*1 Shown in parentheses are JEITA package names.

1.2 Block Diagram



* The pin configuration and peripheral circuit function depends on the model. For detailed information, refer to Section 1.3, "Pins."

Figure 1.2.1 S1C17M20/M21/M22/M23/M24/M25 Block Diagram

1.3 Pins

1.3.1 S1C17M20/M23 Pin Configuration Diagram

SQFN4-24PIN

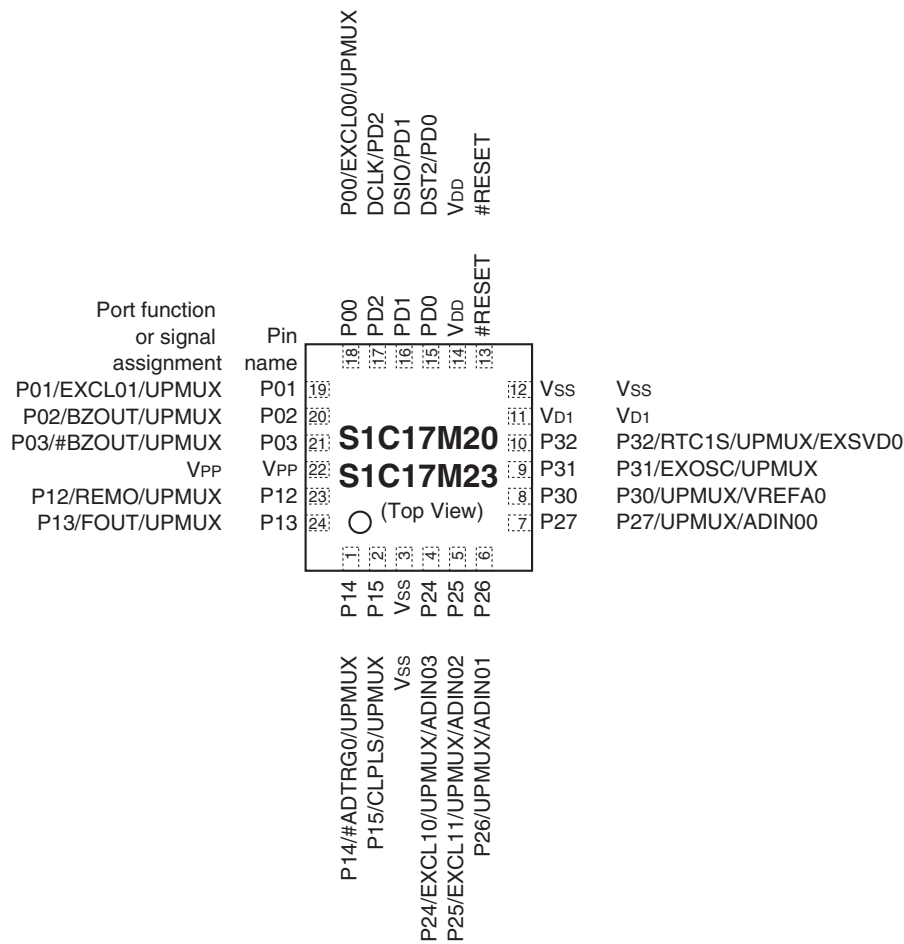


Figure 1.3.1.1 S1C17M20/M23 Pin Configuration Diagram (SQFN4-24PIN)

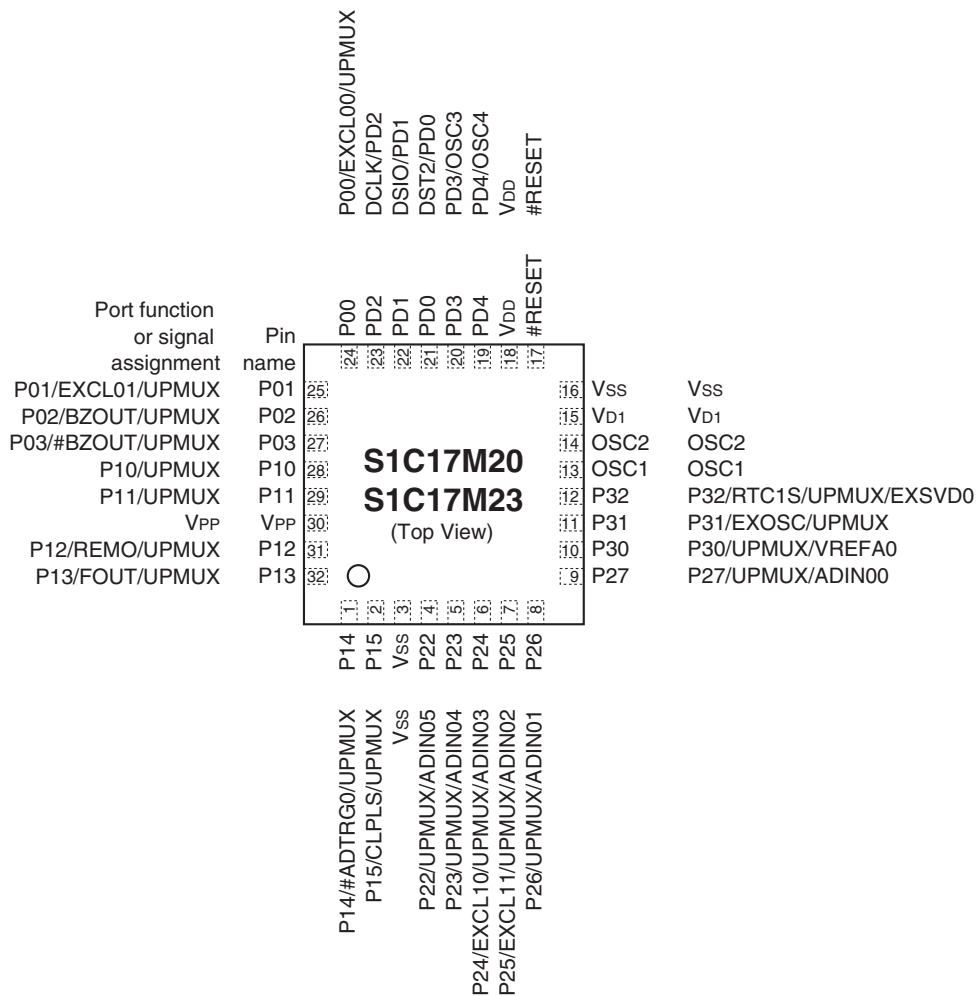
SQFN5-32PIN

Figure 1.3.1.2 S1C17M20/M23 Pin Configuration Diagram (SQFN5-32PIN)

1.3.2 S1C17M21/M24 Pin Configuration Diagram

TQFP12-32PIN

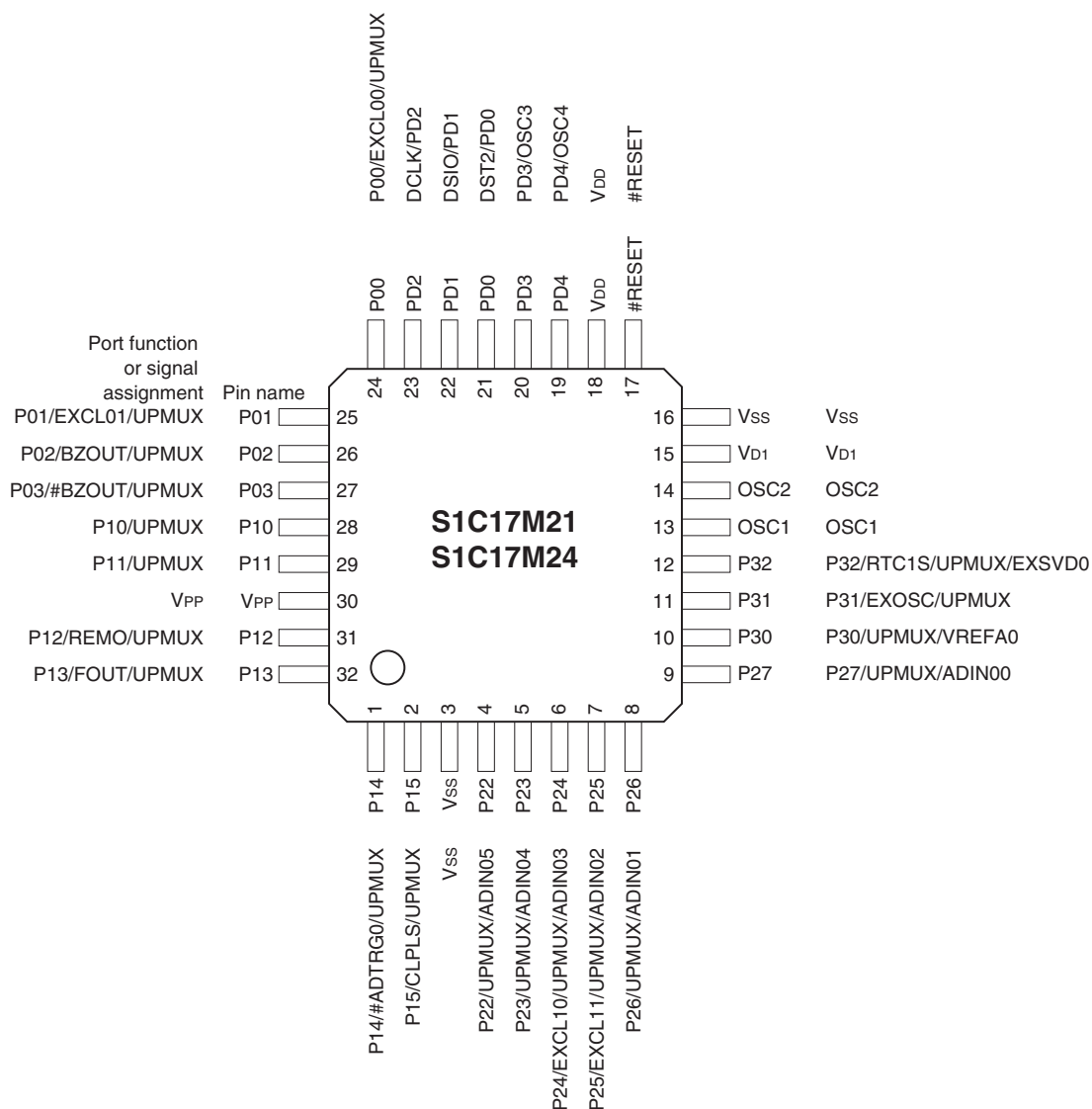


Figure 1.3.2.1 S1C17M21/M24 Pin Configuration Diagram (TQFP12-32PIN)

1.3.3 S1C17M22/M25 Pin Configuration Diagram

TQFP12-48PIN

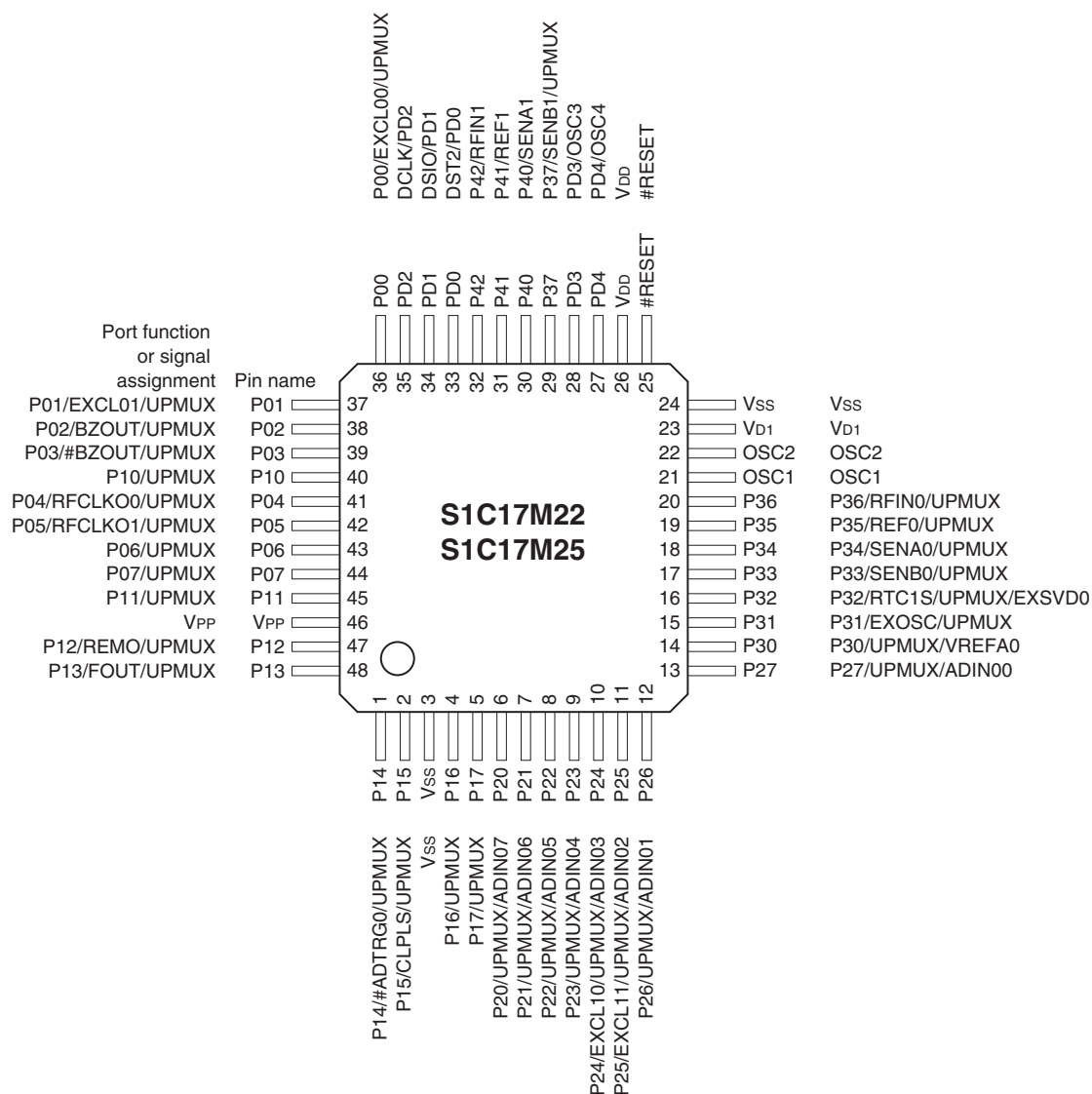


Figure 1.3.3.1 S1C17M22/M25 Pin Configuration Diagram (TQFP12-48PIN)

1.3.4 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 1.3.4.1 Pin description

Pin/pad name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function	M20/M23 (24-pin)	M20/M23 (32-pin)	M21/M24	M22/M25 (48-pin)
V _{DD}	V _{DD}	P	–	–	Power supply (+)	✓	✓	✓	
V _{SS}	V _{SS}	P	–	–	GND	✓	✓	✓	
V _{PP}	V _{PP}	P	–	–	Power supply for Flash programming	✓	✓	✓	
V _{D1}	V _{D1}	A	–	–	V _{D1} regulator output	✓	✓	✓	
OSC1	OSC1	A	–	–	OSC1 oscillator circuit input	–	✓	✓	
OSC2	OSC2	A	–	–	OSC1 oscillator circuit output	–	✓	✓	
#RESET	#RESET	I	I (Pull-up)	–	Reset input	✓	✓	✓	
P00	P00	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)	✓	✓	✓	
P01	P01	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)	✓	✓	✓	
P02	P02	I/O	Hi-Z	✓	I/O port	✓	✓	✓	
	BZOUT	O			Sound generator output	✓	✓	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓	
P03	P03	I/O	Hi-Z	✓	I/O port	✓	✓	✓	
	#BZOUT	O			Sound generator inverted output	✓	✓	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓	
P04	P04	I/O	Hi-Z	✓	I/O port	–	–	✓	
	RFCLK00	O			R/F converter Ch.0 clock monitor output	–	–	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓	
P05	P05	I/O	Hi-Z	✓	I/O port	–	–	✓	
	RFCLK01	O			R/F converter Ch.1 clock monitor output	–	–	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓	
P06	P06	I/O	Hi-Z	✓	I/O port	–	–	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓	
P07	P07	I/O	Hi-Z	✓	I/O port	–	–	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓	
P10	P10	I/O	Hi-Z	✓	I/O port	–	✓	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓	✓	
P11	P11	I/O	Hi-Z	✓	I/O port	–	✓	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓	✓	
P12	P12	I/O	Hi-Z	✓	I/O port	✓	✓	✓	
	REMO	O			IR remote controller transmit data output	✓	✓	✓	
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓	

Pin/pad name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function	M20/M23 (24-pin)	M20/M23 M21/M24 (32-pin)	M22/M25 (48-pin)
P13	P13	I/O	Hi-Z	✓	I/O port	✓	✓	✓
	FOUT	O			Clock external output	✓	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓
P14	P14	I/O	Hi-Z	✓	I/O port	✓	✓	✓
	#ADTRG0	I			12-bit A/D converter Ch.0 trigger input	✓	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓
P15	P15	I/O	Hi-Z	✓	I/O port	✓	✓	✓
	CLPLS	O			IR remote controller clear pulse output	✓	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓
P16	P16	I/O	Hi-Z	✓	I/O port	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓
P17	P17	I/O	Hi-Z	✓	I/O port	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓
P20	P20	I/O	Hi-Z	–	I/O port	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓
	ADIN07	A			12-bit A/D converter Ch.0 analog signal input 7	–	–	✓
P21	P21	I/O	Hi-Z	–	I/O port	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓
	ADIN06	A			12-bit A/D converter Ch.0 analog signal input 6	–	–	✓
P22	P22	I/O	Hi-Z	–	I/O port	–	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓	✓
	ADIN05	A			12-bit A/D converter Ch.0 analog signal input 5	–	✓	✓
P23	P23	I/O	Hi-Z	–	I/O port	–	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	✓	✓
	ADIN04	A			12-bit A/D converter Ch.0 analog signal input 4	–	✓	✓
P24	P24	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)	✓	✓	✓
	ADIN03	A			12-bit A/D converter Ch.0 analog signal input 3	✓	✓	✓
P25	P25	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)	✓	✓	✓
	ADIN02	A			12-bit A/D converter Ch.0 analog signal input 2	✓	✓	✓
P26	P26	I/O	Hi-Z	–	I/O port	✓	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓
	ADIN01	A			12-bit A/D converter Ch.0 analog signal input 1	✓	✓	✓
P27	P27	I/O	Hi-Z	–	I/O port	✓	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓
	ADIN00	A			12-bit A/D converter Ch.0 analog signal input 0	✓	✓	✓
P30	P30	I/O	Hi-Z	–	I/O port	✓	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓
	VREFA0	A			12-bit A/D converter Ch.0 reference voltage input	✓	✓	✓
P31	P31	I/O	Hi-Z	✓	I/O port	✓	✓	✓
	EXOSC	I			Clock generator external clock input	✓	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓
P32	P32	I/O	Hi-Z	✓	I/O port	✓	✓	✓
	RTC1S	O			Real-time clock 1-second cycle pulse output	✓	✓	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	✓	✓	✓
	EXSVD0	A			External power supply voltage detection input	✓	✓	✓
P33	P33	I/O	Hi-Z	✓	I/O port	–	–	✓
	SENB0	A			R/F converter Ch.0 sensor B oscillator pin	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓
P34	P34	I/O	Hi-Z	✓	I/O port	–	–	✓
	SENA0	A			R/F converter Ch.0 sensor A oscillator pin	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓
P35	P35	I/O	Hi-Z	✓	I/O port	–	–	✓
	REF0	A			R/F converter Ch.0 reference oscillator pin	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓

1 OVERVIEW

Pin/pad name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function	M20/M23 (24-pin)	M20/M23 (32-pin) M21/M24	M22/M25 (48-pin)
P36	P36	I/O	Hi-Z	✓	I/O port	–	–	✓
	RFIN0	A			R/F converter Ch.0 oscillation input	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓
P37	P37	I/O	Hi-Z	✓	I/O port	–	–	✓
	SENB1	A			R/F converter Ch.1 sensor B oscillator pin	–	–	✓
	UPMUX	I/O			User-selected I/O (universal port multiplexer)	–	–	✓
P40	P40	I/O	Hi-Z	✓	I/O port	–	–	✓
	SENA1	A			R/F converter Ch.1 sensor A oscillator pin	–	–	✓
P41	P41	I/O	Hi-Z	✓	I/O port	–	–	✓
	REF1	A			R/F converter Ch.1 reference oscillator pin	–	–	✓
P42	P42	I/O	Hi-Z	✓	I/O port	–	–	✓
	RFIN1	A			R/F converter Ch.1 oscillation input	–	–	✓
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	✓	✓	✓
PD3	PD3	I/O	Hi-Z	✓	I/O port	–	✓	✓
	OSC3	A			OSC3 oscillator circuit input	–	✓	✓
PD4	PD4	I/O	Hi-Z	✓	I/O port	–	✓	✓
	OSC4	A			OSC3 oscillator circuit output	–	✓	✓

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 1.3.4.2 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)	SDIn	I	<i>n</i> = 0, 1	SPIA Ch. <i>n</i> data input
	SDOn	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, 1	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 (PWG)

2.1.1 Overview

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

- Embedded V_{D1} regulator
 - The V_{D1} regulator generates the V_{D1} voltage to drive internal circuits, 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.

Figure 2.1.1.1 shows the PWG configuration.

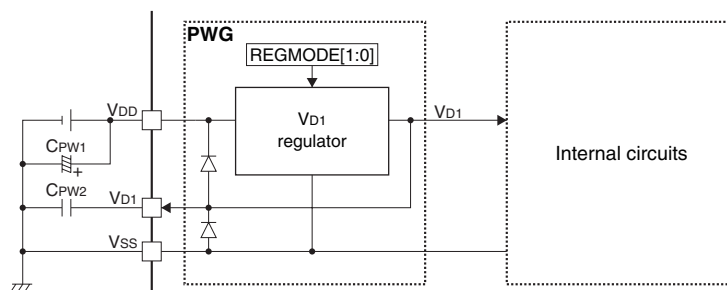


Figure 2.1.1.1 PWG Configuration

2.1.2 Pins

Table 2.1.2.1 lists the PWG pins.

Table 2.1.2.1 List of PWG Pins

Pin name	I/O	Initial status	Function
V_{DD}	P	–	Power supply (+)
V_{SS}	P	–	GND
V_{D1}	A	–	Embedded 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.

2.1.3 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.

2.2 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 and BOR
 - 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.

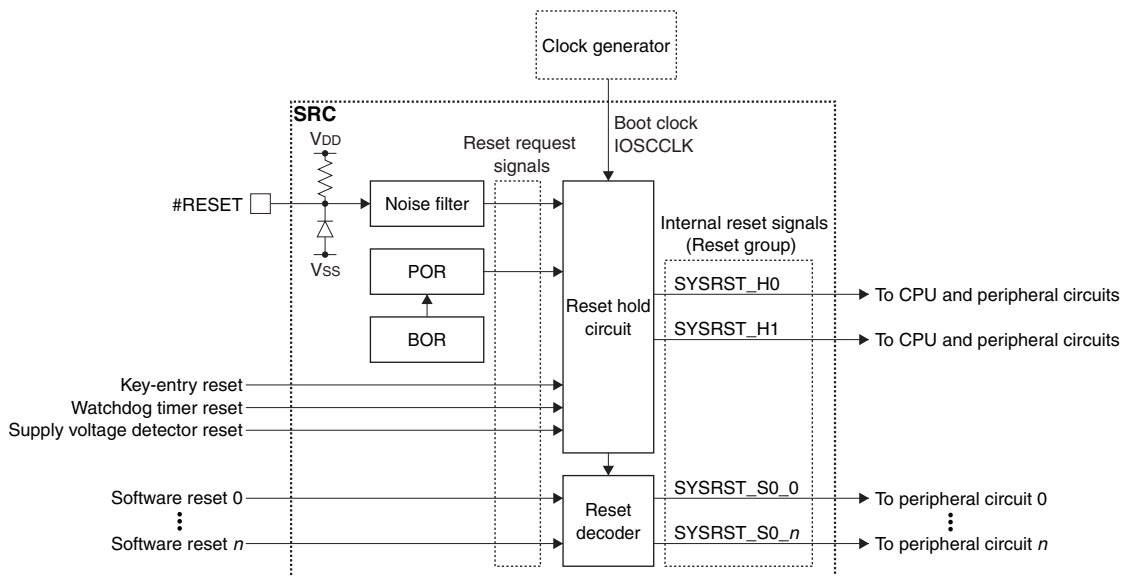


Figure 2.2.1.1 SRC Configuration

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.

2.2.3 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 and BOR

POR (Power On Reset) issues a reset request when the rise of V_{DD} is detected. BOR (Brownout Reset) issues a reset request when a certain V_{DD} voltage level is detected. Reset requests from these circuits ensure that the system will be reset properly when the power is turned on and the supply voltage is out of the operating voltage range. Figure 2.2.3.1 shows an example of POR and BOR internal reset operation according to variations in V_{DD} .

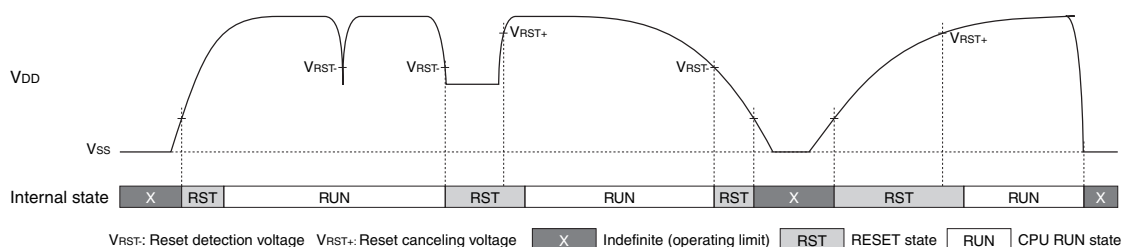


Figure 2.2.3.1 Example of Internal Reset by POR and BOR

For the POR and BOR electrical specifications, refer to “POR/BOR 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 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 POR and BOR Key-entry reset Supply voltage detector reset Watchdog timer reset	Reset state is maintained for the reset hold time t_{RSTR} after the reset request is canceled.
H1	#RESET pin POR and BOR	
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 that oscillates with a fast startup and no external parts required
 - Low-power OSC1 oscillator circuit in which the oscillator type can be specified from high-precision 32.768 kHz crystal oscillator (an external resonator is required) and internal oscillator
 - High-speed OSC3 oscillator circuit in which the oscillator type can be specified from crystal/ceramic oscillator (an external resonator is required) and internal oscillator
 - EXOSC clock input circuit that allows input of square wave and sine wave clock signals
- 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.

Table 2.3.1.1 CLG Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M22/M24/M25
	24-pin package	32-pin package	
IOSC oscillator circuit	Available	Available	Available
OSC1 crystal oscillator circuit	Unavailable	Available	Available
OSC1 internal oscillator circuit	Available	Available	Available
OSC3 crystal/ceramic oscillator circuit	Unavailable	Available	Available
OSC3 internal oscillator circuit	Available	Available	Available
EXOSC clock input	Available	Available	Available

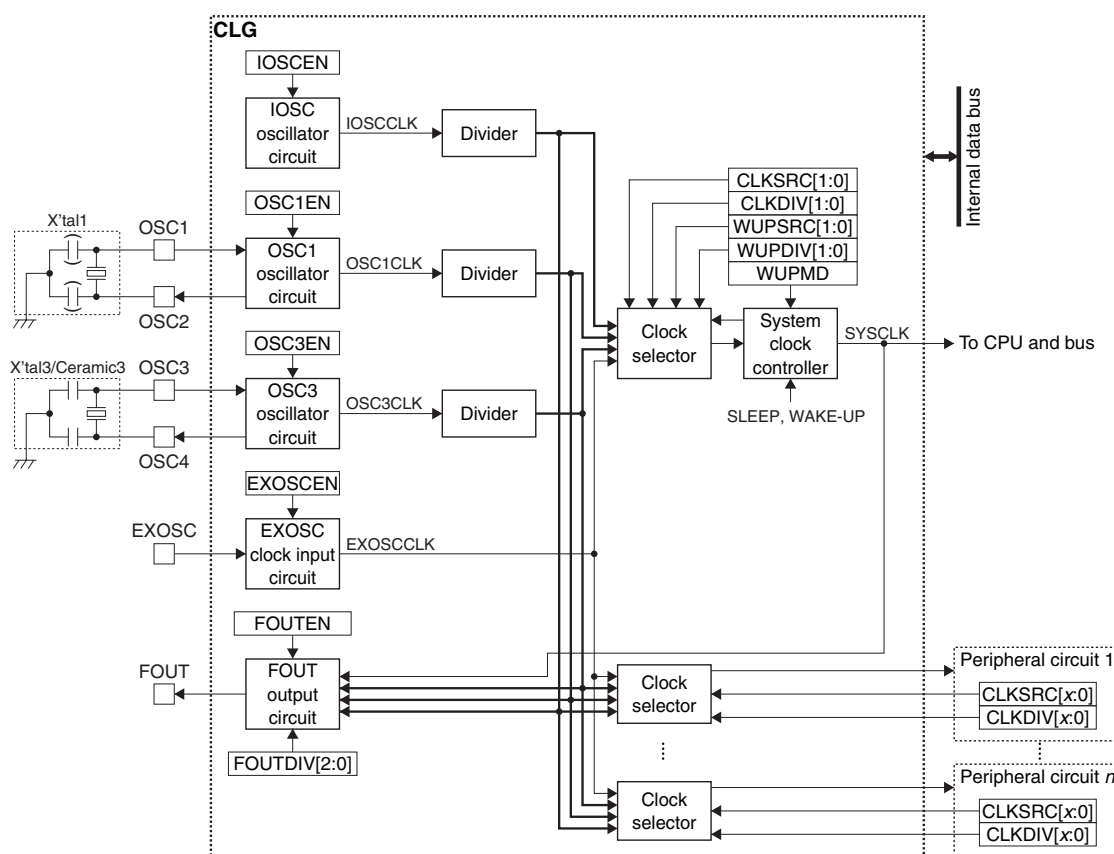


Figure 2.3.1.1 CLG Configuration

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
OSC1	A	–	OSC1 oscillator circuit input
OSC2	A	–	OSC1 oscillator circuit output
OSC3	A	–	OSC3 oscillator circuit input
OSC4	A	–	OSC3 oscillator circuit output
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.

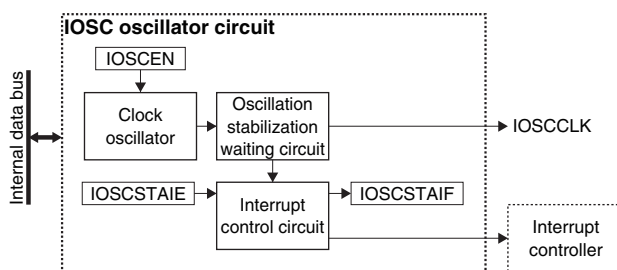


Figure 2.3.3.1 IOSC Oscillator Circuit Configuration

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 low-power oscillator circuit that allows software to select the oscillator type from two different types shown below. Figure 2.3.3.2 shows the configuration of the OSC1 oscillator circuit.

Crystal oscillator

This oscillator circuit includes a gain-controlled oscillation inverter and a variable gate capacitor allowing use of various crystal resonators (32.768 kHz typ.) with ranges from cylinder type through surface-mount type. The oscillator circuit also includes a feedback resistor and a drain resistor, so no external parts are required except for a crystal resonator. The embedded oscillation stop detector, which detects oscillation stop and restarts the oscillator, allows the system to operate in safety under adverse environments that may stop the oscillation. The oscillation startup control circuit operates for a set period of time after the oscillation is enabled to assist the oscillator in initiating, this makes it possible to use a low-power resonator that is difficult to start up.

Note: Depending on the circuit board or the crystal resonator type used, an external gate capacitor C_{G1} and a drain capacitor C_{D1} may be required.

Internal oscillator

This 32 kHz oscillator circuit operates without any external parts.

When the internal oscillator circuit is used, the OSC1 and OSC3 pins must be left open.

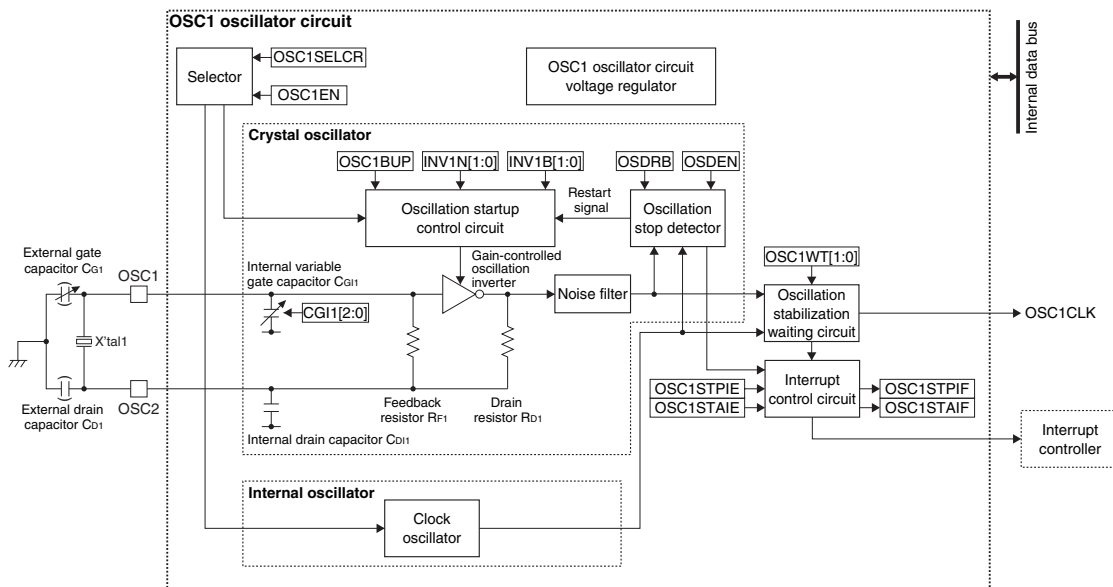


Figure 2.3.3.2 OSC1 Oscillator Circuit Configuration

For the recommended parts and the oscillation characteristics, refer to the “Basic External Connection Diagram” chapter and “OSC1 oscillator circuit characteristics” in the “Electrical Characteristics” chapter, respectively.

OSC3 oscillator circuit

The OSC3 oscillator circuit is a high-speed oscillator circuit that allows software to select the oscillator type from two different types shown below. Figure 2.3.3.3 shows the configuration of the OSC3 oscillator circuit.

Crystal/ceramic oscillator

This oscillator circuit includes a feedback resistor and a drain resistor, so no external part is required except for a crystal/ceramic resonator. The embedded gain-controlled inverter allows selection of the resonator from a wide frequency range.

Internal oscillator

This oscillator circuit features a fast startup and no external parts are required for oscillating. The OSC3CLK frequency can be selected using the CLGOSC3.OSC3FQ[1:0] bits. This oscillator circuit is equipped with an auto-trimming function that automatically adjusts the frequency. This helps reduce frequency deviation due to unevenness in manufacturing quality, temperature, and changes in voltage. For more information on the auto-trimming function, refer to “OSC3 oscillation auto-trimming function” in this chapter.

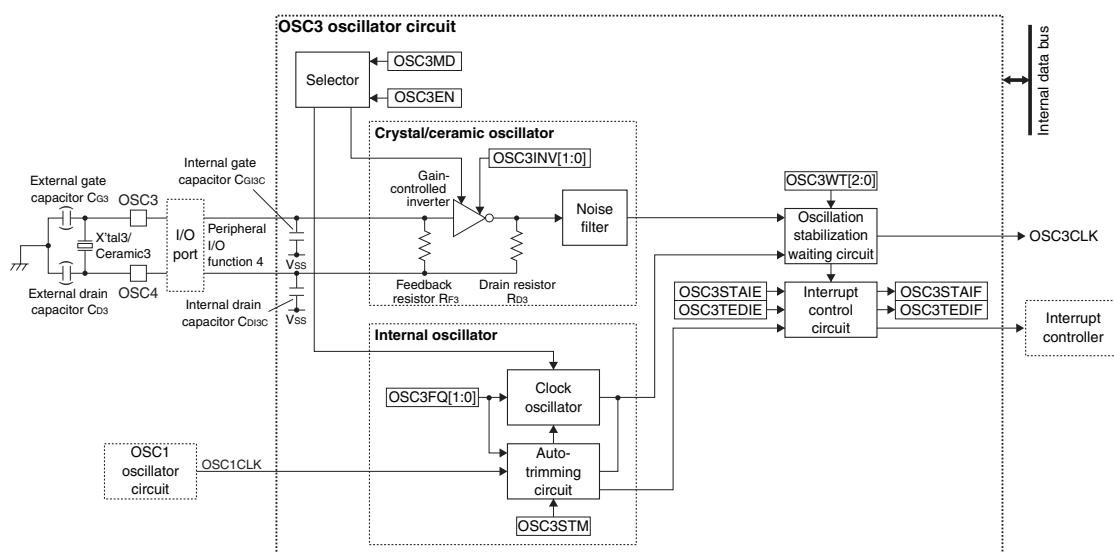


Figure 2.3.3.3 OSC3 Oscillator Circuit Configuration

For the recommended parts and the oscillation characteristics, refer to the “Basic External Connection Diagram” chapter and “OSC3 oscillator circuit characteristics” in the “Electrical Characteristics” chapter, respectively.

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.

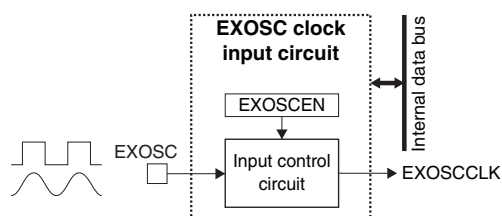


Figure 2.3.3.4 EXOSC Clock Input Circuit

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.

2.3.4 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.

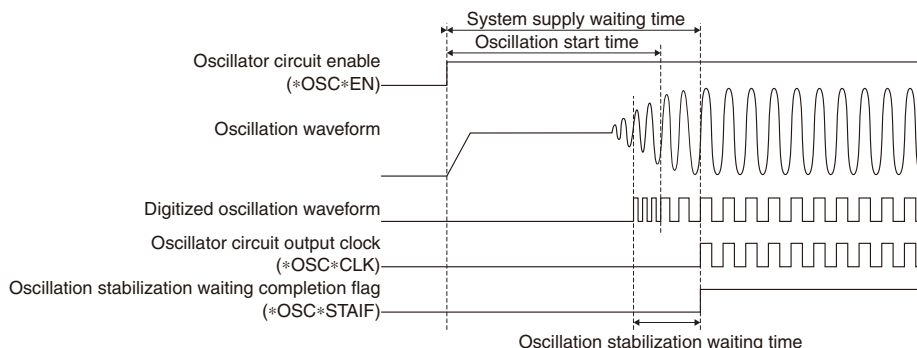


Figure 2.3.4.1 Oscillation Start Time and Oscillation Stabilization Waiting Time

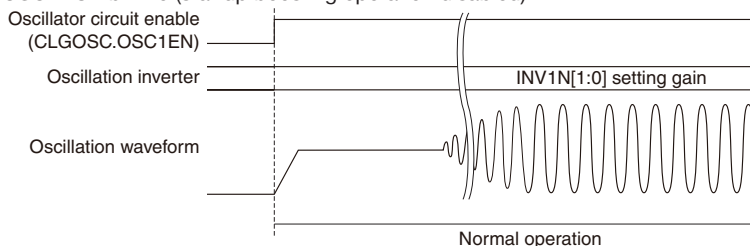
The oscillation stabilization waiting times for the OSC1 and OSC3 oscillator circuits can be set using the CLGOSC1.OSC1WT[1:0] bits and CLGOSC3.OSC3WT[2:0] bits, respectively. To check whether the oscillation stabilization waiting time is set properly and the clock is stabilized immediately after the 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 16,384 OSC1CLK clocks or more when crystal oscillator is selected, or 4,096 OSC1CLK clocks or more when internal oscillator is selected. The oscillation stabilization waiting time for the OSC3 oscillator circuit should be set to 1,024 OSC3CLK clocks or more when crystal/ceramic oscillator is selected, or four OSC3CLK clocks or more when internal oscillator is selected.

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 be cleared to 0.

When the oscillation startup control circuit in the OSC1 crystal oscillator circuit is enabled by setting the CLGOSC1.OSC1BUP bit to 1, it uses the high-gain oscillation inverter for a set period of time (startup boosting operation) after the oscillator circuit is enabled (by setting the CLGOSC.OSC1EN bit to 1) to reduce oscillation start time. Note, however, that the oscillation operation may become unstable if there is a large gain differential between normal operation and startup boosting operation. Furthermore, the oscillation start time being actually reduced depends on the characteristics of the resonator used. Figure 2.3.4.2 shows an operation example when the oscillation startup control circuit is used.

(1) CLGOSC1.OSC1BUP bit = 0 (startup boosting operation disabled)



(2) CLGOSC1.OSC1BUP bit = 1 (startup boosting operation enabled)

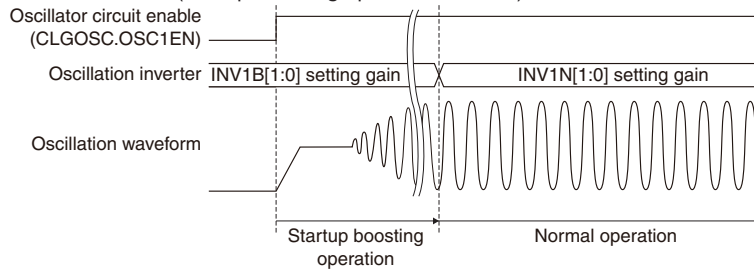


Figure 2.3.4.2 Operation Example when the OSC1 Crystal Oscillation Startup Control Circuit is Used

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 CLGINTF.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. IOSCLK can be used if the CLGINTF.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 CLGINTF.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 following CLGOSC1 register bits:
 - CLGOSC1.OSC1SELCR bit (Select oscillator type)
 - CLGOSC1.OSC1WT[1:0] bits (Set oscillation stabilization waiting time)

In addition to the above, configure the following bits when using the crystal oscillator:

- CLGOSC1.INV1N[1:0] bits (Set oscillation inverter gain)
 - CLGOSC1.CGI1[2:0] bits (Set internal gate capacitor)
 - CLGOSC1.INV1B[1:0] bits (Set oscillation inverter gain for startup boosting period)
 - CLGOSC1.OSC1BUP bit (Enable/disable oscillation startup control circuit)
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 CLGINTF.OSC1STAIF bit = 1 after an interrupt occurs.

The setting values of the CLGOSC1.INV1N[1:0], CLGOSC1.CGI1[2:0], CLGOSC1.OSC1WT[1:0], and CLGOSC1.INV1B[1:0] bits should be determined after performing evaluation using the populated circuit board.

Note: Make sure the CLGOSC.OSC1EN bit is set to 0 (while the OSC3 oscillation is halted) when switching the oscillator within two types.

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.OSC3MD bit (Select oscillator type)
 - CLGOSC3.OSC3WT[2:0] bits (Set oscillation stabilization waiting time)
- In addition to the above, configure the following bits when using the crystal/ceramic oscillator:
- CLGOSC3.OSC3INV[1:0] bits (Set oscillation inverter gain)
- Configure the following bits when using the internal oscillator:
- CLGOSC3.OSC3FQ[1:0] bits (Select oscillation frequency)

5. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
6. When using the crystal/ceramic oscillator, assign the OSC3 oscillator input/output functions to the ports. (Refer to the “I/O Ports” chapter.)
7. Write 1 to the CLGOSC.OSC3EN bit. (Start oscillation)
8. OSC3CLK can be used if the CLGINTF.OSC3STAIF bit = 1 after an interrupt occurs.

The setting values of the CLGOSC3.OSC3INV[1:0] and CLGOSC3.OSC3WT[2:0] bits should be determined after performing evaluation using the populated circuit board.

Note: Make sure the CLGOSC.OSC3EN bit is set to 0 (while the OSC3 oscillation is halted) when switching the oscillator within two types.

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 2.3.4.3 shows a control example.

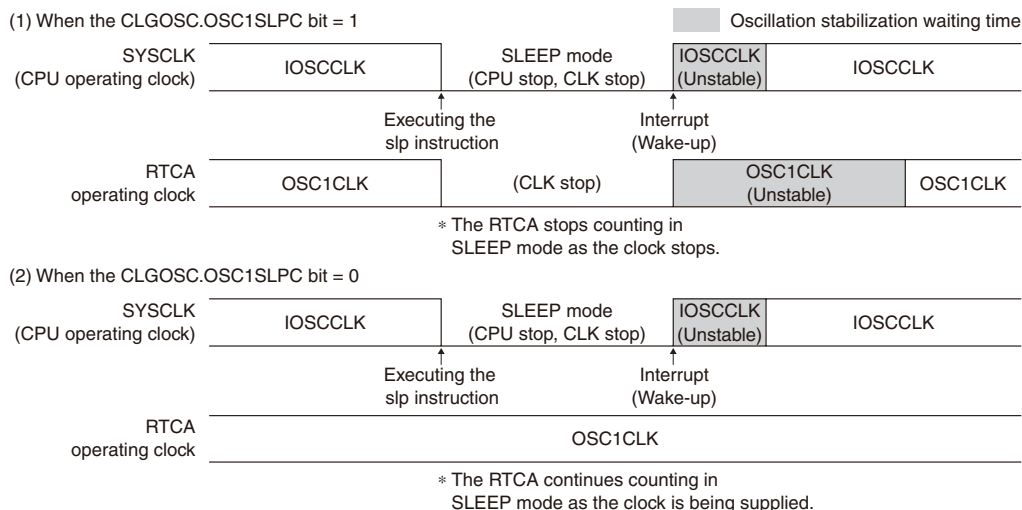


Figure 2.3.4.3 Clock Control Example in SLEEP Mode

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.

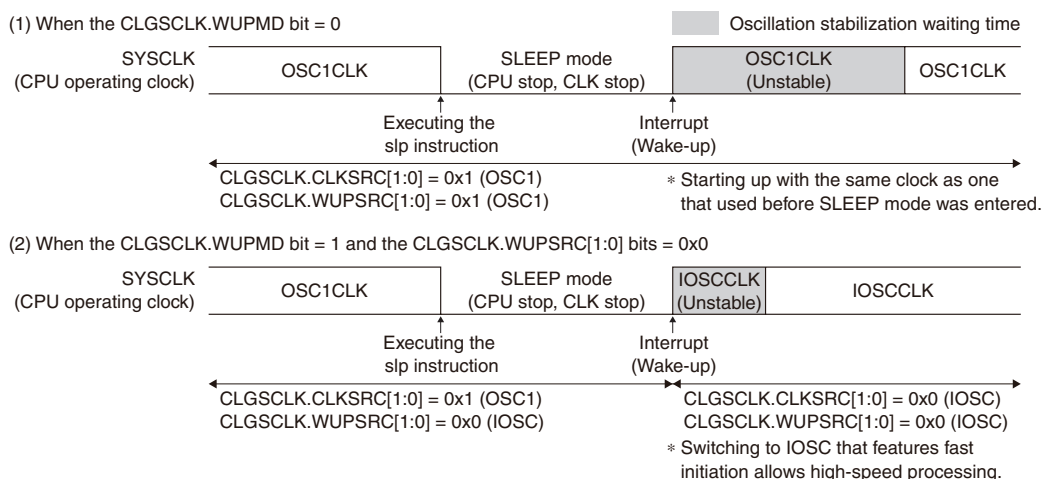


Figure 2.3.4.4 Clock Control Example at SLEEP Cancellation

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)

OSC3 oscillation auto-trimming function

The OSC3 internal oscillator circuit has the auto-trimming function that adjusts the OSC3CLK clock frequency by trimming the clock with reference to the high precision OSC1CLK clock generated by the OSC1 crystal oscillator circuit. Follow the procedure shown below to enable the auto-trimming function.

1. After enabling the OSC1 oscillation, check if the stabilized clock is supplied (CLGINTF.OSC1STAIF bit = 1).
2. After enabling the OSC3 oscillation, check if the stabilized clock is supplied (CLGINTF.OSC3STAIF bit = 1).
3. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
4. If the SYSCLK clock source is OSC3, set the CLGSCLK.CLKSRC[1:0] bits to a value other than 0x2 (OSC3).
5. Write 1 to the CLGINTF.OSC3TEDIF bit. (Clear interrupt flag)
6. Write 1 to the CLGINTF.OSC3TEDIE bit. (Enable interrupt)
7. Write 1 to the CLGOSC3.OSC3STM bit. (Enable OSC3 oscillation auto-trimming)
8. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
9. The trimmed OSC3CLK can be used if the CLGINTF.OSC3TEDIF bit = 1 after an interrupt occurs.

After the trimming operation has completed, the CLGOSC3.OSC3STM bit automatically reverts to 0. Although the trimming time depends on the temperature, an average of several 10 ms is required. When OSC3CLK is being used as the system clock or a peripheral circuit clock, do not use the auto-trimming function.

OSC1 oscillation stop detection function

The oscillation stop detection function restarts the OSC1 oscillator circuit when it detects oscillation stop under adverse environments that may stop the oscillation. Follow the procedure shown below to enable the oscillation stop detection function.

1. After enabling the OSC1 oscillation, check if the stabilized clock is supplied (CLGINTF.OSC1STAIF bit = 1).
2. Write 1 to the CLGINTF.OSC1STPIF bit. (Clear interrupt flag)
3. Write 1 to the CLGINTF.OSC1STPIE bit. (Enable interrupt)
4. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
5. Set the following CLGOSC1 register bits:
 - Set the CLGOSC1.OSDRB bit to 1. (Enable OSC1 restart function)
 - Set the CLGOSC1.OSDEN bit to 1. (Enable oscillation stop detection function)
6. Write a value other than 0x0096 to the MSCPROT.PROT[15:0] bits. (Set system protection)
7. The OSC1 oscillation stops if the CLGINTF.OSC1STPIF bit = 1 after an interrupt occurs.
If the CLGOSC1.OSDRB bit = 1, the hardware restarts the OSC1 oscillator circuit.

Note: Enabling the oscillation stop detection function increase the oscillation stop detector current (I_{OSD1}).

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.

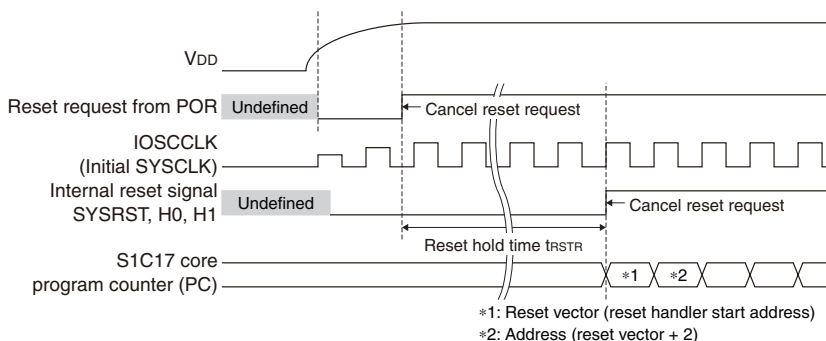


Figure 2.4.1.1 Initial Boot Sequence

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 tr_{STR} , 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.

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 I_{HALT1} , I_{HALT2} , and I_{HALT3} ” in the “Electrical Characteristics” chapter).

DEBUG mode

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

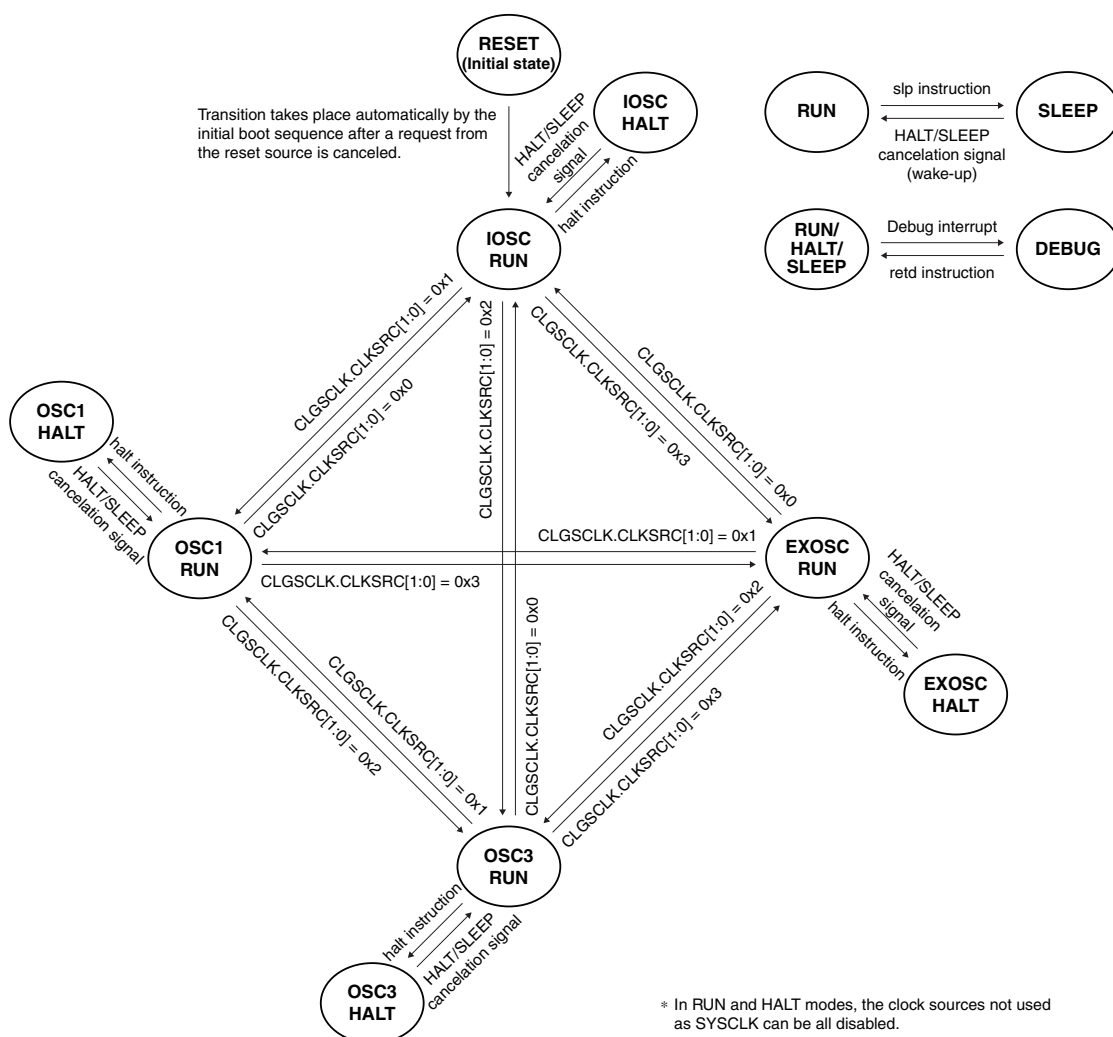


Figure 2.4.2.1 Operating Mode-to-Mode State Transition Diagram

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

2.5 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
OSC1 oscillation stop	CLGINTF.OSC1STPIF	When OSC1CLK is stopped, or when the CLGOSC.OSC1EN or CLGOSC1.OSDEN bit setting is altered from 1 to 0.	Writing 1
OSC3 oscillation auto-trimming completion	CLGINTF.OSC3TEDIF	When the OSC3 oscillation auto-trimming operation has completed	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

Note: Do not alter the initial values of the control bits for the functions that are not supported in the model to be used.

PWG V_{D1} Regulator Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PWGVD1CTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1–0	REGMODE[1:0]	0x0	H0	R/WP	

Bits 15–2 Reserved

Bits 1–0 REGMODE[1:0]

These bits control the internal regulator operating mode.

Table 2.6.1 Internal Regulator Operating Mode

PWGVD1CTL.REGMODE[1:0] bits	Operating mode
0x3	Economy mode
0x2	Normal mode
0x1	Reserved
0x0	Automatic mode

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 2.6.2 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 2.6.3 SYSCCLK 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

CLG OSC1 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGOSC1	15	–	0	–	R	–
	14	OSDRB	1	H0	R/WP	
	13	OSDEN	0	H0	R/WP	
	12	OSC1BUP	1	H0	R/WP	
	11	OSC1SELCR	0	H0	R/WP	
	10–8	CGI1[2:0]	0x0	H0	R/WP	
	7–6	INV1B[1:0]	0x2	H0	R/WP	
	5–4	INV1N[1:0]	0x1	H0	R/WP	
	3–2	–	0x0	–	R	
	1–0	OSC1WT[1:0]	0x2	H0	R/WP	

Bit 15 **Reserved**

Bit 14 **OSDRB**

This bit enables the OSC1 oscillator circuit restart function by the oscillation stop detector when OSC1 crystal oscillation stop is detected.

1 (R/WP): Enable (Restart the OSC1 oscillator circuit when oscillation stop is detected.)

0 (R/WP): Disable

Bit 13 **OSDEN**

This bit controls the oscillation stop detector in the OSC1 oscillator circuit.

1 (R/WP): OSC1 oscillation stop detector on

0 (R/WP): OSC1 oscillation stop detector off

Note: Do not write 1 to the CLGOSC1.OSDEN bit before stabilized OSC1CLK is supplied. Furthermore, the CLGOSC1.OSDEN bit should be set to 0 when the CLGOSC.OSC1EN bit is set to 0.

Bit 12 **OSC1BUP**

This bit enables the oscillation startup control circuit in the OSC1 crystal oscillator circuit.

1 (R/WP): Enable (Activate booster operation at startup.)

0 (R/WP): Disable

Bit 11 **OSC1SELCR**

This bit selects an oscillator type of the OSC1 oscillator circuit.

1 (R/WP): Internal oscillator

0 (R/WP): Crystal oscillator

Bits 10–8 **CGI1[2:0]**

These bits set the internal gate capacitance in the OSC1 crystal oscillator circuit.

Table 2.6.4 OSC1 Internal Gate Capacitance Setting

CLGOSC1.CGI1[2:0] bits	Capacitance
0x7	Max. ↑
0x6	
0x5	
0x4	
0x3	
0x2	
0x1	↓ Min.
0x0	

For more information, refer to “OSC1 oscillator circuit characteristics, Crystal oscillator internal gate capacitance C_{G11C} ” in the “Electrical Characteristics” chapter.

Bits 7–6 **INV1B[1:0]**

These bits set the oscillation inverter gain that will be applied at boost startup of the OSC1 crystal oscillator circuit.

Table 2.6.5 Setting Oscillation Inverter Gain at OSC1 Boost Startup

CLGOSC1.INV1B[1:0] bits	Inverter gain
0x3	Max.
0x2	↑
0x1	↓
0x0	Min.

Note: The CLGOSC1.INV1B[1:0] bits must be set to a value equal to or larger than the CLGOSC1.INV1N[1:0] bits.

Bits 5–4 INV1N[1:0]

These bits set the oscillation inverter gain applied at normal operation of the OSC1 crystal oscillator circuit.

Table 2.6.6 Setting Oscillation Inverter Gain at OSC1 Normal Operation

CLGOSC1.INV1N[1:0] bits	Inverter gain
0x3	Max.
0x2	↑
0x1	↓
0x0	Min.

Bits 3–2 Reserved**Bits 1–0 OSC1WT[1:0]**

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

Table 2.6.7 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	15–12	–	0x0	–	R	–
	11–10	OSC3FQ[1:0]	0x1	H0	R/WP	
	9	OSC3MD	0	H0	R/WP	
	8	–	0	–	R	
	7–6	–	0x0	–	R	
	5–4	OSC3INV[1:0]	0x3	H0	R/WP	
	3	OSC3STM	0	H0	R/WP	
	2–0	OSC3WT[2:0]	0x6	H0	R/WP	

Bits 15–12 Reserved**Bits 11–10 OSC3FQ[1:0]**

These bits set the oscillation frequency of the OSC3 internal oscillator circuit.

Table 2.6.8 Setting Oscillation Frequency of OSC3 Internal Oscillator Circuit

CLGOSC3.OSC3FQ[1:0] bits	Oscillation frequency
0x3	Reserved
0x2	20 MHz
0x1	16 MHz
0x0	12 MHz

Bit 9 OSC3MD

This bit selects an oscillator type of the OSC3 oscillator circuit.

1 (R/WP): Crystal/ceramic oscillator

0 (R/WP): Internal oscillator

Bits 8–6 Reserved

Bits 5–4 OSC3INV[1:0]

These bits set the oscillation inverter gain of the OSC3 crystal/ceramic oscillator circuit.

Table 2.6.9 OSC3 Oscillation Inverter Gain Setting

CLGOSC3.OSC3INV[1:0] bits	Inverter gain
0x3	Max.
0x2	↑
0x1	↓
0x0	Min.

Bit 3 OSC3STM

This bit controls the OSC3 internal oscillator auto-trimming function.

1 (WP): Start trimming

0 (WP): Stop trimming

1 (R): Trimming is executing.

0 (R): Trimming has finished. (Trimming operation inactivated.)

This bit is automatically cleared to 0 when trimming has finished.

- Notes:**
- Do not use OSC3CLK as the system clock or peripheral circuit clocks while the CLGOSC3.OSC3STM bit = 1.
 - The auto-trimming function does not work if the OSC1 oscillator circuit is stopped. Make sure the CLGINTF.OSC1STAIF bit is set to 1 before starting the trimming operation.
 - Do not alter the CLGOSC3.OSC3FQ[1:0] bits while auto-trimming is being executed.
 - Select the 32.768 kHz crystal oscillator for the OSC1 oscillator circuit when using the auto-trimming function. The clock cannot be adjusted properly by the internal oscillator.

Bits 2–0 OSC3WT[2:0]

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

Table 2.6.10 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	–
	7	–	0x0	–	R	
	6	(reserved)	0	H0	R	
	5	OSC1STPIF	0	H0	R/W	Cleared by writing 1.
	4	OSC3TEDIF	0	H0	R/W	
	3	–	0	–	R	–
	2	OSC3STAIF	0	H0	R/W	Cleared by writing 1.
	1	OSC1STAIF	0	H0	R/W	
	0	IOSCSTAIF	0	H0	R/W	

Bits 15–6, 3 Reserved

Bit 5 **OSC1STPIF**
Bit 4 **OSC3TEDIF**
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.OSC1STPIF bit: OSC1 oscillation stop interrupt

CLGINTF.OSC3TEDIF bit: OSC3 oscillation auto-trimming completion interrupt

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	–	0	–	R	
	6	(reserved)	0	H0	R	
	5	OSC1STPIE	0	H0	R/W	
	4	OSC3TEDIE	0	H0	R/W	
	3	–	0	–	R	
	2	OSC3STAIE	0	H0	R/W	
	1	OSC1STAIE	0	H0	R/W	
	0	IOSCSTAIE	0	H0	R/W	

Bits 15–6, 3 Reserved

Bit 5 **OSC1STPIE**
Bit 4 **OSC3TEDIE**
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.OSC1STPIE bit: OSC1 oscillation stop interrupt

CLGINTE.OSC3TEDIE bit: OSC3 oscillation auto-trimming completion interrupt

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

CLGFOUT 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.

Bits 3–2 **FOUTSRC[1:0]**

These bits select the FOUT clock sources.

Table 2.6.11 FOUT Clock Source and Division Ratio Settings

CLGFOUT. FOUTDIV[2:0] bits	CLGFOUT.FOUTSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCLK	OSC1CLK	OSC3CLK	SYSCLK
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.

3 CPU and Debugger

3.1 Overview

This IC incorporates the Seiko Epson original 16-bit CPU core (S1C17) with a debugger. The main features of the CPU core 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.

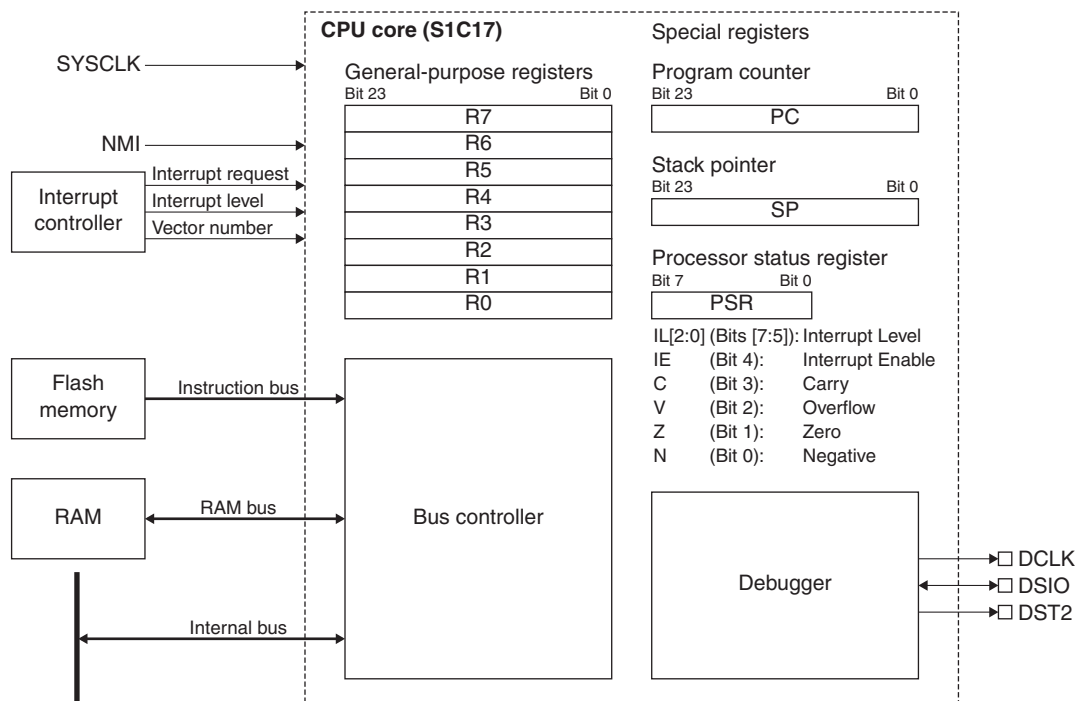


Figure 3.1.1 S1C17 Configuration

3.2 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 retid 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.

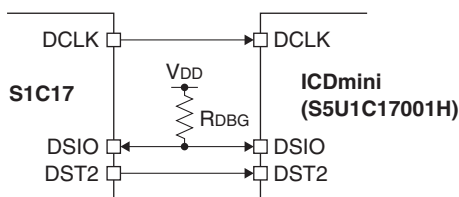


Figure 3.3.4.1 External Connection

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.

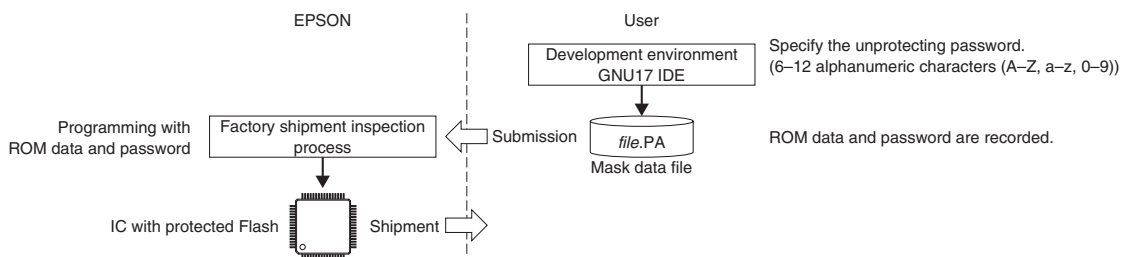


Figure 3.3.5.1 Shipment of IC with ROM Data Programmed and Flash Security Function Setting Flow

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.
- Write-protect function to protect system control registers

Figure 4.1.1 shows the memory map.

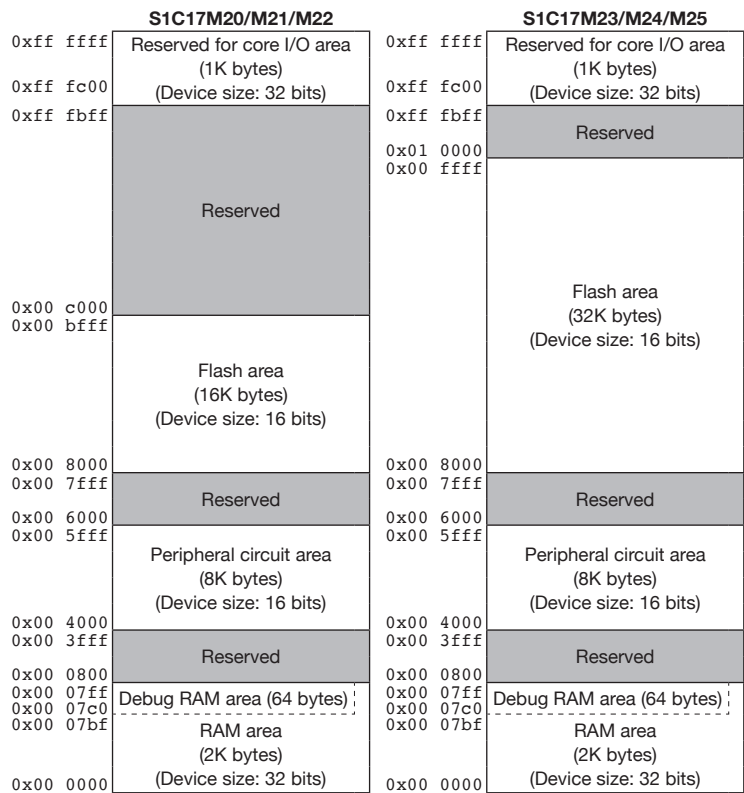


Figure 4.1.1 Memory Map

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)

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 internal RAM area and accesses data in the internal RAM 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: Always leave the V_{PP} pin open except when programming the Flash memory.

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.

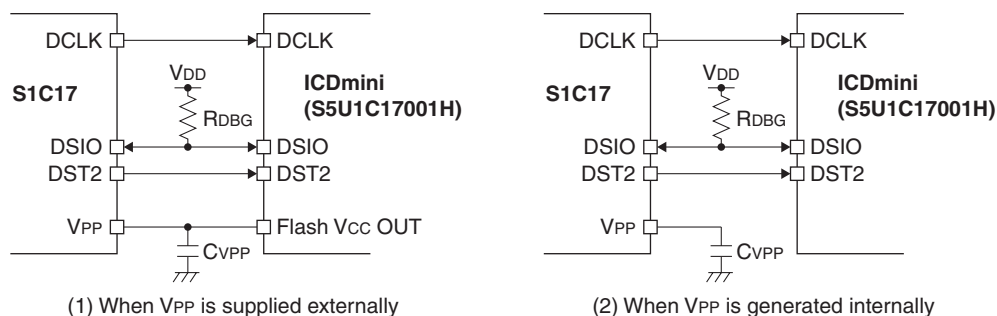


Figure 4.3.3.1 External Connection

The VPP 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 VPP voltage, as ICDmini controls the power supply so that it will be supplied during Flash programming only. The VPP voltage can also be generated by the internal power supply for generating the Flash programming voltage. Be sure to connect CVPP for stabilizing the voltage when the VPP 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: • The Flash programming requires a 2.4 V or higher VDD voltage.

- Be sure to avoid using the VPP pin output for driving external circuits when the VPP voltage is generated internally.

4.4 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.

4.5 Peripheral Circuit Control Registers

The control registers for the peripheral circuits are located in the 8K-byte area beginning with address 0x4000. Table 4.5.1 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 4.5.1 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	MSCTTBRH MISC Vector Table Address High Register
	0x4008	MSCPSR MISC PSR Register
Power generator (PWG)	0x4020	PWGVD1CTL PWG V _{D1} 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
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
	0x4092	ITCLV9 ITC Interrupt Level Setup Register 9
	0x4094	ITCLV10 ITC Interrupt Level Setup Register 10
Watchdog timer (WDT2)	0x40a0	WDTCLK WDT2 Clock Control Register
	0x40a2	WDTCTL WDT2 Control Register
	0x40a4	WDTCMP WDT2 Counter Compare Match Register
Real-time clock (RTCA)	0x40c0	RTCCTL RTC Control Register
	0x40c2	RTCALM1 RTC Second Alarm Register
	0x40c4	RTCALM2 RTC Hour/Minute Alarm Register
	0x40c6	RTCSWCTL RTC Stopwatch Control Register
	0x40c8	RTCSEC RTC Second/1Hz Register
	0x40ca	RTCHUR RTC Hour/Minute Register
	0x40cc	RTCMON RTC Month/Day Register
	0x40ce	RTCYAR RTC Year/Week Register
	0x40d0	RTCINTF RTC Interrupt Flag Register
Supply voltage detector (SVD3)	0x4100	SVDCLK SVD3 Clock Control Register
	0x4102	SVDCTL SVD3 Control Register
	0x4104	SVDINTF SVD3 Status and Interrupt Flag Register
	0x4106	SVDINTE SVD3 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
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
	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

Peripheral circuit	Address	Register name
I/O ports (PPORT)	0x4216	P1INTF P1 Port Interrupt Flag Register
	0x4218	P1INTCTL P1 Port Interrupt Control Register
	0x421a	P1CHATEN P1 Port Chattering Filter Enable Register
	0x421c	P1MODSEL P1 Port Mode Select Register
	0x421e	P1FNCSSEL 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	P2MODSEL P2 Port Mode Select Register
	0x422e	P2FNCSSEL P2 Port Function Select Register
	0x4230	P3DAT P3 Port Data Register
	0x4232	P3IOEN P3 Port Enable Register
	0x4234	P3RCTL P3 Port Pull-up/down Control Register
	0x4236	P3INTF P3 Port Interrupt Flag Register
	0x4238	P3INTCTL P3 Port Interrupt Control Register
	0x423a	P3CHATEN P3 Port Chattering Filter Enable Register
	0x423c	P3MODSEL P3 Port Mode Select Register
	0x423e	P3FNCSSEL P3 Port Function Select Register
	0x4240	P4DAT P4 Port Data Register *1 *2 *3
	0x4242	P4IOEN P4 Port Enable Register *1 *2 *3
	0x4244	P4RCTL P4 Port Pull-up/down Control Register *1 *2 *3
	0x4246	P4INTF P4 Port Interrupt Flag Register *1 *2 *3
	0x4248	P4INTCTL P4 Port Interrupt Control Register *1 *2 *3
	0x424a	P4CHATEN P4 Port Chattering Filter Enable Register *1 *2 *3
	0x424c	P4MODSEL P4 Port Mode Select Register *1 *2 *3
	0x424e	P4FNCSSEL P4 Port Function Select Register *1 *2 *3
	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	PDFNCSSEL 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 *1 *2 *3
	0x4306	P0UPMUX3 P06–07 Universal Port Multiplexer Setting Register *1 *2 *3
	0x4308	P1UPMUX0 P10–11 Universal Port Multiplexer Setting Register *1
	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 *1 *2 *3
	0x4310	P2UPMUX0 P20–21 Universal Port Multiplexer Setting Register *1 *2 *3
	0x4312	P2UPMUX1 P22–23 Universal Port Multiplexer Setting Register *1
	0x4314	P2UPMUX2 P24–25 Universal Port Multiplexer Setting Register
	0x4316	P2UPMUX3 P26–27 Universal Port Multiplexer Setting Register
	0x4318	P3UPMUX0 P30–31 Universal Port Multiplexer Setting Register
	0x431a	P3UPMUX1 P32–33 Universal Port Multiplexer Setting Register
	0x431c	P3UPMUX2 P34–35 Universal Port Multiplexer Setting Register *1 *2 *3
	0x431e	P3UPMUX3 P36–37 Universal Port Multiplexer Setting Register *1 *2 *3
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

4 MEMORY AND BUS

Peripheral circuit	Address	Register name
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
Synchronous serial interface (SPIA) Ch.0	0x43ac	T16_1INTE T16 Ch.1 Interrupt Enable Register
	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
I ² C (I2C)	0x43ba	SPI0INTE SPIA Ch.0 Interrupt Enable Register
	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
	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
UART (UART3) Ch.1	0x5200	UA1CLK UART3 Ch.1 Clock Control Register
	0x5202	UA1MOD UART3 Ch.1 Mode Register
	0x5204	UA1BR UART3 Ch.1 Baud-Rate Register
	0x5206	UA1CTL UART3 Ch.1 Control Register
	0x5208	UA1TXD UART3 Ch.1 Transmit Data Register
	0x520a	UA1RXD UART3 Ch.1 Receive Data Register
	0x520c	UA1INTF UART3 Ch.1 Status and Interrupt Flag Register
	0x520e	UA1INTE UART3 Ch.1 Interrupt Enable Register
	0x4390	UA1CAWF UART3 Ch.1 Carrier Waveform 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

Peripheral circuit	Address	Register name
Synchronous serial interface (SPIA) Ch.1	0x5270	SPI1MOD SPIA Ch.1 Mode Register
	0x5272	SPI1CTL SPIA Ch.1 Control Register
	0x5274	SPI1TXD SPIA Ch.1 Transmit Data Register
	0x5276	SPI1RXD SPIA Ch.1 Receive Data Register
	0x5278	SPI1INTF SPIA Ch.1 Interrupt Flag Register
	0x527a	SPI1INTE SPIA Ch.1 Interrupt Enable Register
Sound generator (SNDA)	0x5300	SNDCCLK SNDA Clock Control Register
	0x5302	SNDSEL SNDA Select Register
	0x5304	SNDCTL SNDA Control Register
	0x5306	SNDDAT SNDA Data Register
	0x5308	SNDINTF SNDA Interrupt Flag Register
	0x530a	SNDINTE SNDA Interrupt Enable Register
IR remote controller (REMC3)	0x5320	REMCCLK REMC3 Clock Control Register
	0x5322	REMCDBCTL REMC3 Data Bit Counter Control Register
	0x5324	REMCDBCNT REMC3 Data Bit Counter Register
	0x5326	REMAPLEN REMC3 Data Bit Active Pulse Length Register
	0x5328	REMCDBLEN REMC3 Data Bit Length Register
	0x532a	REMINTF REMC3 Status and Interrupt Flag Register
	0x532c	REMINTE REMC3 Interrupt Enable Register
	0x5330	REMCARR REMC3 Carrier Waveform Register
	0x5332	REMCCTL REMC3 Carrier Modulation Control Register
R/F converter (RFC) Ch.0	0x5440	RFC0CLK RFC Ch.0 Clock Control Register *1 *2 *3
	0x5442	RFC0CTL RFC Ch.0 Control Register *1 *2 *3
	0x5444	RFC0TRG RFC Ch.0 Oscillation Trigger Register *1 *2 *3
	0x5446	RFC0MCL RFC Ch.0 Measurement Counter Low Register *1 *2 *3
	0x5448	RFC0MCH RFC Ch.0 Measurement Counter High Register *1 *2 *3
	0x544a	RFC0TCL RFC Ch.0 Time Base Counter Low Register *1 *2 *3
	0x544c	RFC0TCH RFC Ch.0 Time Base Counter High Register *1 *2 *3
	0x544e	RFC0INTF RFC Ch.0 Interrupt Flag Register *1 *2 *3
	0x5450	RFC0INTE RFC Ch.0 Interrupt Enable Register *1 *2 *3
R/F converter (RFC) Ch.1	0x5460	RFC1CLK RFC Ch.1 Clock Control Register *1 *2 *3
	0x5462	RFC1CTL RFC Ch.1 Control Register *1 *2 *3
	0x5464	RFC1TRG RFC Ch.1 Oscillation Trigger Register *1 *2 *3
	0x5466	RFC1MCL RFC Ch.1 Measurement Counter Low Register *1 *2 *3
	0x5468	RFC1MCH RFC Ch.1 Measurement Counter High Register *1 *2 *3
	0x546a	RFC1TCL RFC Ch.1 Time Base Counter Low Register *1 *2 *3
	0x546c	RFC1TCH RFC Ch.1 Time Base Counter High Register *1 *2 *3
	0x546e	RFC1INTF RFC Ch.1 Interrupt Flag Register *1 *2 *3
	0x5470	RFC1INTE RFC Ch.1 Interrupt Enable Register *1 *2 *3
16-bit timer (T16) Ch.3	0x5480	T16_3CLK T16 Ch.3 Clock Control Register
	0x5482	T16_3MOD T16 Ch.3 Mode Register
	0x5484	T16_3CTL T16 Ch.3 Control Register
	0x5486	T16_3TR T16 Ch.3 Reload Data Register
	0x5488	T16_3TC T16 Ch.3 Counter Data Register
	0x548a	T16_3INTF T16 Ch.3 Interrupt Flag Register
	0x548c	T16_3INTE T16 Ch.3 Interrupt Enable Register
12-bit A/D converter (ADC12A)	0x54a2	ADC12_0CTL ADC12A Ch.0 Control Register
	0x54a4	ADC12_0TRG ADC12A Ch.0 Trigger/Analog Input Select Register
	0x54a6	ADC12_0CFG ADC12A Ch.0 Configuration Register
	0x54a8	ADC12_0INTF ADC12A Ch.0 Interrupt Flag Register
	0x54aa	ADC12_0INTE ADC12A Ch.0 Interrupt Enable Register
	0x54ac	ADC12_0AD0D ADC12A Ch.0 Result Register 0
	0x54ae	ADC12_0AD1D ADC12A Ch.0 Result Register 1
	0x54b0	ADC12_0AD2D ADC12A Ch.0 Result Register 2
	0x54b2	ADC12_0AD3D ADC12A Ch.0 Result Register 3
	0x54b4	ADC12_0AD4D ADC12A Ch.0 Result Register 4 *1
	0x54b6	ADC12_0AD5D ADC12A Ch.0 Result Register 5 *1
	0x54b8	ADC12_0AD6D ADC12A Ch.0 Result Register 6 *1 *2 *3
	0x54ba	ADC12_0AD7D ADC12A Ch.0 Result Register 7 *1 *2 *3

*1 Cannot be used in the S1C17M20/M23 (24-pin package).

*2 Cannot be used in the S1C17M20/M23 (32-pin package).

*3 Cannot be used in the S1C17M21/M24.

4.5.1 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.

4.6 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 4.6.1 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	15–9	–	0x00	–	R	–
	8	(reserved)	0	H0	R/WP	Always set to 0.
	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 4.6.2 Setting Number of Bus Access Cycles for Flash Read

FLASHCWAIT.RDWAIT[1:0] bits	Number of bus Access cycles	System clock frequency
0x3	4	21.0 MHz (max.)
0x2	3	18.9 MHz (max.)
0x1	2	12.6 MHz (max.)
0x0	1	6.3 MHz (max.)

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

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.

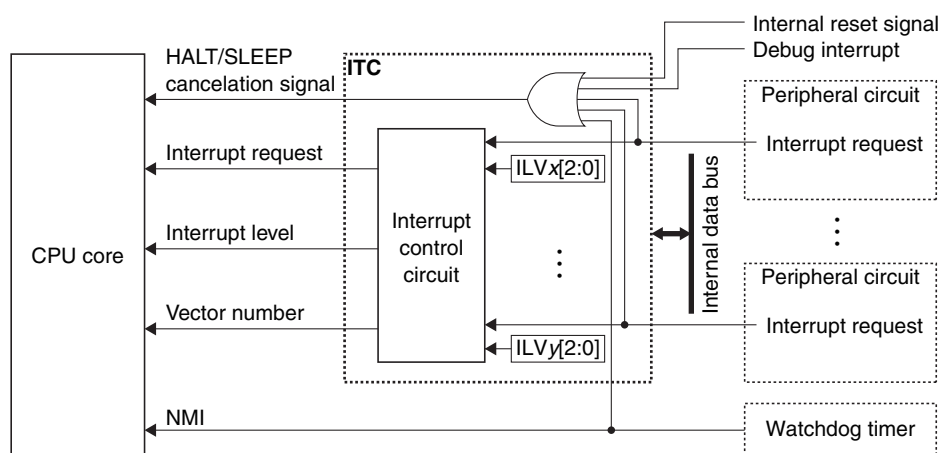


Figure 5.1.1 ITC Configuration

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	Hardware interrupt flag	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 • OSC1 oscillation stop • IOSC oscillation auto-trimming completion 	
8 (0x08)	TTBR + 0x20	Real-time clock interrupt	<ul style="list-style-type: none"> • 1-day, 1-hour, 1-minute, and 1-second • 1/32-second, 1/8-second, 1/4-second, and 1/2-second • Stopwatch 1 Hz, 10 Hz, and 100 Hz • Alarm • Theoretical regulation completion 	
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	UART Ch.1 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 	
17 (0x11)	TTBR + 0x44	Sound generator interrupt	<ul style="list-style-type: none"> • Sound buffer empty • Sound output completion 	
18 (0x12)	TTBR + 0x48	IR remote controller interrupt	<ul style="list-style-type: none"> • Compare AP • Compare DB 	
19 (0x13)	TTBR + 0x4c	reserved	—	
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	R/F converter Ch.1 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 	
22 (0x16)	TTBR + 0x58	16-bit timer Ch.2 interrupt	Underflow	

Vector number/ Software interrupt number	Vector address	Hardware interrupt name	Hardware interrupt flag	Priority
23 (0x17)	TTBR + 0x5c	Synchronous serial interface Ch.1 interrupt	<ul style="list-style-type: none"> • End of transmission • Receive buffer full • Transmit buffer empty • Overrun error 	↓ Low *1
24 (0x18)	TTBR + 0x60	16-bit timer Ch.3 interrupt	Underflow	
25 (0x19)	TTBR + 0x64	12-bit A/D converter interrupt	<ul style="list-style-type: none"> • Analog input signal <i>m</i> A/D conversion completion • Analog input signal <i>m</i> A/D conversion result overwrite error 	
26 (0x1a) : 31 (0x1f)	TTBR + 0x68 : TTBR + 0x7c	reserved : reserved	– : –	

*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 ITCLV_x.ILV_x[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.

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
ITCLVx	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.

5 INTERRUPT CONTROLLER (ITC)

Table 5.8.1 Interrupt Level and Priority Settings

ITCLVx.ILVy[2:0] bits	Interrupt level	Priority
0x7	7	High ↑
0x6	6	
...	...	
0x1	1	↓ Low
0x0	0	

The following shows the ITCLVx register configuration in this IC.

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 (ILVSVD3)
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)
	7–3	–	0x00	–	R	–
	2–0	ILV4[2:0]	0x0	H0	R/W	Real-time clock interrupt (ILVRTCA_0)
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	Sound generator interrupt (ILVSND_A_0)
	7–3	–	0x00	–	R	–
	2–0	ILV12[2:0]	0x0	H0	R/W	UART Ch.1 interrupt (ILVUART3_1)
ITCLV7 (ITC Interrupt Level Setup Register 7)	15–8	–	0x00	–	R	–
	7–3	–	0x00	–	R	–
	2–0	ILV14[2:0]	0x0	H0	R/W	IR remote controller interrupt (ILVREMC3_0)
ITCLV8 (ITC Interrupt Level Setup Register 8)	15–11	–	0x00	–	R	–
	10–8	ILV17[2:0]	0x0	H0	R/W	R/F converter Ch.1 interrupt (ILVRFC_1)
	7–3	–	0x00	–	R	–
	2–0	ILV16[2:0]	0x0	H0	R/W	R/F converter Ch.0 interrupt (ILVRFC_0)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ITCLV9 (ITC Interrupt Level Setup Register 9)	15–11	–	0x00	–	R	–
	10–8	ILV19[2:0]	0x0	H0	R/W	Synchronous serial interface Ch.1 interrupt (ILVSPIA_1)
	7–3	–	0x00	–	R	–
	2–0	ILV18[2:0]	0x0	H0	R/W	16-bit timer Ch.2 interrupt (ILVT16_2)
ITCLV10 (ITC Interrupt Level Setup Register 10)	15–11	–	0x00	–	R	–
	10–8	ILV21[2:0]	0x0	–	R/W	12-bit A/D converter interrupt (ILVADC12A_0)
	7–3	–	0x00	–	R	–
	2–0	ILV20[2:0]	0x0	–	R/W	16-bit timer Ch.3 interrupt (ILVT16_3)

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.
- 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 S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
Port groups included	P0 P0[3:0] (4) *1, *2	P0 P0[3:0] (4) *1, *2	P0 P0[3:0] (4) *1, *2	P0 P0[7:0] (8) *1, *2
	P1 P1[5:2] (4) *1, *2	P1 P1[5:0] (6) *1, *2	P1 P1[5:0] (6) *1, *2	P1 P1[7:0] (8) *1, *2
	P2 P2[7:4] (4) *1, *2	P2 P2[7:2] (6) *1, *2	P2 P2[7:2] (6) *1, *2	P2 P2[7:0] (8) *1, *2
	P3 P3[2:0] (3) *1, *2	P3 P3[2:0] (3) *1, *2	P3 P3[2:0] (3) *1, *2	P3 P3[7:0] (8) *1, *2
	P4 — (0)	— (0)	— (0)	P4 P4[2:0] (3) *1, *2
	Pd Pd[2:0] (3) *1 (Pd2: output only)	Pd Pd[4:0] (5) *1 (Pd2: output only)	Pd Pd[4:0] (5) *1 (Pd2: output only)	Pd Pd[4:0] (5) *1 (Pd2: output only)
Total number of ports	Input/output port: 17 Output port: 1	Input/output port: 23 Output port: 1	Input/output port: 23 Output port: 1	Input/output port: 39 Output port: 1
Ports for debug function	Pd[2:0]			
Key-entry reset function	Supported (P0[3:0])			

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

*2 Ports with interrupt function

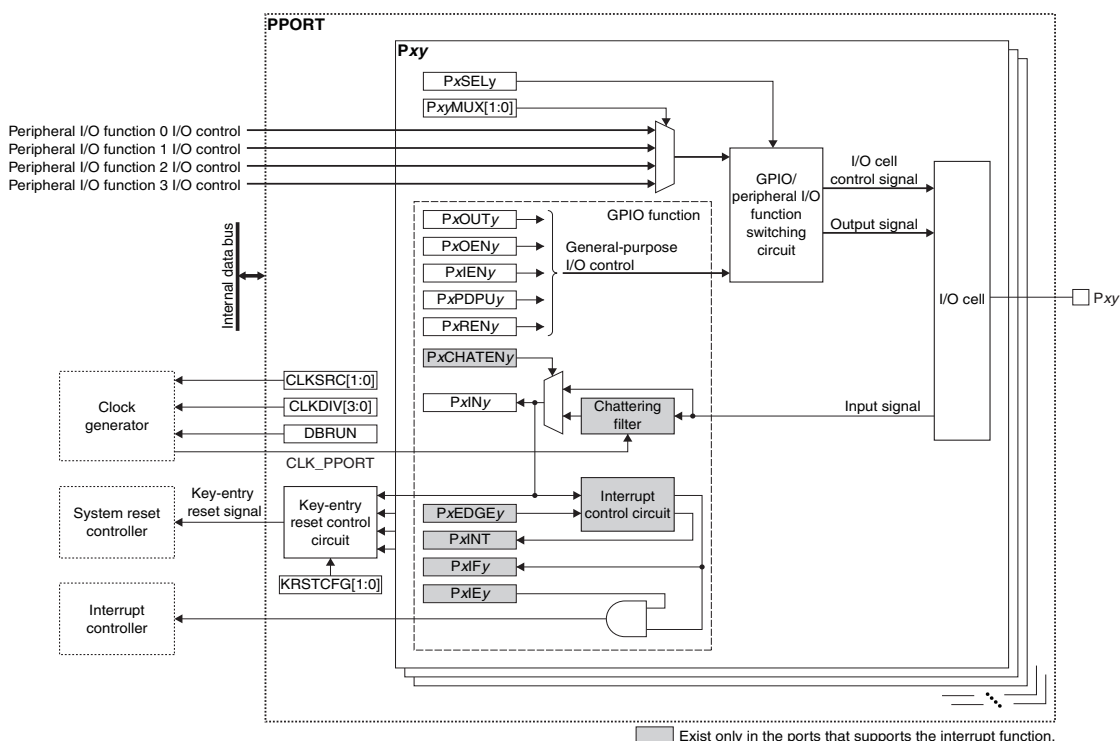


Figure 6.1.1 PPORT Configuration

6.2 I/O Cell Structure and Functions

Figure 6.2.1 shows the I/O cell Configuration.

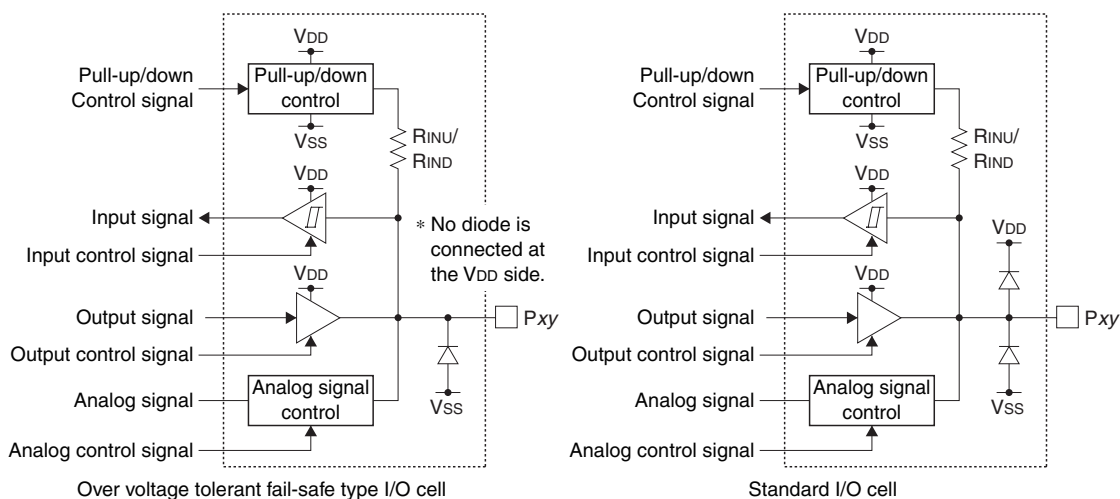


Figure 6.2.1 I/O Cell Configuration

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 VDD is applied to the port. Also unnecessary current is not consumed when the port is externally biased without supplying VDD. 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:

$$\begin{aligned} t_{PR} &= -R_{INU} \times (C_{IN} + C_{BOARD}) \times \ln(1 - V_{T+}/V_{DD}) \\ t_{PF} &= -R_{IND} \times (C_{IN} + C_{BOARD}) \times \ln(1 - V_{T-}/V_{DD}) \end{aligned} \quad (\text{Eq. 6.1})$$

Where

tPR:	Rising time (port level = low → high) [second]
tPF:	Falling time (port level = high → low) [second]
VT+:	High level Schmitt input threshold voltage [V]
VT-:	Low level Schmitt input threshold voltage [V]
RINU/RIND:	Pull-up/pull-down resistance [Ω]
CIN:	Pin capacitance [F]
CBOARD:	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 VDD and Vss levels. Also the GPIO ports may be put into high-impedance (Hi-Z) state.

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 Pxy port for a peripheral I/O function, perform the following software initial settings:

1. Set the following PxIOEN register bits:
 - Set the PxIOEN.PxIENy bit to 0. (Disable input)
 - Set the PxIOEN.PxOENy bit to 0. (Disable output)
2. Set the PxMODSEL.PxSELy bit to 0. (Disable peripheral I/O function)
3. Initialize the peripheral circuit that uses the pin.
4. Set the PxFNCSSEL.PxyMUX[1:0] bits. (Select peripheral I/O function)
5. Set the PxMODSEL.PxSELy 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 Pxy port pin as a general-purpose output pin, perform the following software initial settings:

1. Set the PxIOEN.PxOENy bit to 1. (Enable output)
2. Set the PxMODSEL.PxSELy 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 Px_y 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 PxMODESEL.PxSEL_y bit to 0. (Enable GPIO function)
5. Configure the following bits when using the port input interrupt: *
 - Write 1 to the PxINTF.PxIF_y bit. (Clear interrupt flag)
 - PxINTCTL.PxEDGE_y bit (Select interrupt edge (input rising edge/falling edge))
 - Set the PxINTCTL.PxIE_y bit to 1. (Enable interrupt)
6. Set the following PxIOEN register bits:
 - Set the PxIOEN.PxOEN_y bit to 0. (Disable output)
 - Set the PxIOEN.PxIEN_y 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. PxIEN _y bit	PxIOEN. PxOEN _y bit	PxRCTL. PxREN _y bit	PxRCTL. PxPDU _y bit	Input		Output	Pull-up/pull-down condition
0	0	0	×	Disabled			Off (Hi-Z) *1
0	0	1	0	Disabled			Pulled down
0	0	1	1	Disabled			Pulled up
1	0	0	×	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	×	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.PxSEL_y 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 P_{xy} pin to the P_xDAT.P_xOUT_y bit.

Reading input data from a GPIO port

The data (1 = high input, 0 = low input) input from the P_{xy} pin can be read out from the P_xDAT.P_xIN_y 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 P_xCHATEN.P_xCHATEN_y bit to 1. The input sampling time to remove chattering is determined by the CLK_PPORT 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.

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

Make sure the P_{xy} port interrupt is disabled before altering the PCLK register and P_xCHATEN.P_xCHATEN_y bit settings. A P_{xy} 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_PPORT cycles from enabling the chattering filter function.

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 P_xINTCTL.P_xEDGE_y 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.PxOENy bit = 1), the port pin outputs the data set here. Although data can be written when output is disabled (PxIOEN.PxOENy 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.PxIENy bit = 1). When input is disabled (PxIOEN.PxIENy 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.)

6 I/O PORTS (PPORT)

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	PxDPU[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 PxPDU[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.PxRENy 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.PxOENy bit = 0).

When output is enabled (PxIOEN.PxOENy bit = 1), the PxRCTL.PxRENy bit setting is ineffective regardless of how the PxIOEN.PxIENy 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.

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
PxMODESEL	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
PxFNCSSEL	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 PxMODESEL.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 P0[3:0] inputs = all low
0x2	Reset when P0[2:0] inputs = all low
0x1	Reset when P0[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.

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	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
P0DAT (P0 Port Data Register)	15	P0OUT7	0	H0	R/W	—	—	—	—	✓
	14	P0OUT6	0	H0	R/W		—	—	—	✓
	13	P0OUT5	0	H0	R/W		—	—	—	✓
	12	P0OUT4	0	H0	R/W		—	—	—	✓
	11	P0OUT3	0	H0	R/W		✓	✓	✓	✓
	10	P0OUT2	0	H0	R/W		✓	✓	✓	✓
	9	P0OUT1	0	H0	R/W		✓	✓	✓	✓
	8	P0OUT0	0	H0	R/W		✓	✓	✓	✓
	7	P0IN7	0	H0	R	—	—	—	—	✓
	6	P0IN6	0	H0	R		—	—	—	✓
	5	P0IN5	0	H0	R		—	—	—	✓
	4	P0IN4	0	H0	R		—	—	—	✓
	3	P0IN3	0	H0	R		✓	✓	✓	✓
	2	P0IN2	0	H0	R		✓	✓	✓	✓
	1	P0IN1	0	H0	R		✓	✓	✓	✓
	0	P0IN0	0	H0	R		✓	✓	✓	✓
P0IOEN (P0 Port Enable Register)	15	P0IEN7	0	H0	R/W	—	—	—	—	✓
	14	P0IEN6	0	H0	R/W		—	—	—	✓
	13	P0IEN5	0	H0	R/W		—	—	—	✓
	12	P0IEN4	0	H0	R/W		—	—	—	✓
	11	P0IEN3	0	H0	R/W		✓	✓	✓	✓
	10	P0IEN2	0	H0	R/W		✓	✓	✓	✓
	9	P0IEN1	0	H0	R/W		✓	✓	✓	✓
	8	P0IEN0	0	H0	R/W		✓	✓	✓	✓
	7	P0OEN7	0	H0	R/W	—	—	—	—	✓
	6	P0OEN6	0	H0	R/W		—	—	—	✓
	5	P0OEN5	0	H0	R/W		—	—	—	✓
	4	P0OEN4	0	H0	R/W		—	—	—	✓
	3	P0OEN3	0	H0	R/W		✓	✓	✓	✓
	2	P0OEN2	0	H0	R/W		✓	✓	✓	✓
	1	P0OEN1	0	H0	R/W		✓	✓	✓	✓
	0	P0OEN0	0	H0	R/W		✓	✓	✓	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23 24pin	M21/ 32pin	M24	M22/ M25
P0RCTL (P0 Port Pull-up/down Control Register)	15	P0PDPU7	0	H0	R/W	–	–	–	–	✓
	14	P0PDPU6	0	H0	R/W		–	–	–	✓
	13	P0PDPU5	0	H0	R/W		–	–	–	✓
	12	P0PDPU4	0	H0	R/W		–	–	–	✓
	11	P0PDPU3	0	H0	R/W		✓	✓	✓	✓
	10	P0PDPU2	0	H0	R/W		✓	✓	✓	✓
	9	P0PDPU1	0	H0	R/W		✓	✓	✓	✓
	8	P0PDPU0	0	H0	R/W		✓	✓	✓	✓
	7	P0REN7	0	H0	R/W	–	–	–	–	✓
	6	P0REN6	0	H0	R/W		–	–	–	✓
	5	P0REN5	0	H0	R/W		–	–	–	✓
	4	P0REN4	0	H0	R/W		–	–	–	✓
	3	P0REN3	0	H0	R/W		✓	✓	✓	✓
	2	P0REN2	0	H0	R/W		✓	✓	✓	✓
	1	P0REN1	0	H0	R/W		✓	✓	✓	✓
	0	P0REN0	0	H0	R/W		✓	✓	✓	✓
POINTF (P0 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P0IF7	0	H0	R/W	Cleared by writing 1.	–	–	–	✓
	6	P0IF6	0	H0	R/W		–	–	–	✓
	5	P0IF5	0	H0	R/W		–	–	–	✓
	4	P0IF4	0	H0	R/W		–	–	–	✓
	3	P0IF3	0	H0	R/W		✓	✓	✓	✓
	2	P0IF2	0	H0	R/W		✓	✓	✓	✓
	1	P0IF1	0	H0	R/W		✓	✓	✓	✓
	0	P0IF0	0	H0	R/W		✓	✓	✓	✓
POINTCTL (P0 Port Interrupt Control Register)	15	P0EDGE7	0	H0	R/W	–	–	–	–	✓
	14	P0EDGE6	0	H0	R/W		–	–	–	✓
	13	P0EDGE5	0	H0	R/W		–	–	–	✓
	12	P0EDGE4	0	H0	R/W		–	–	–	✓
	11	P0EDGE3	0	H0	R/W		✓	✓	✓	✓
	10	P0EDGE2	0	H0	R/W		✓	✓	✓	✓
	9	P0EDGE1	0	H0	R/W		✓	✓	✓	✓
	8	P0EDGE0	0	H0	R/W		✓	✓	✓	✓
	7	P0IE7	0	H0	R/W	–	–	–	–	✓
	6	P0IE6	0	H0	R/W		–	–	–	✓
	5	P0IE5	0	H0	R/W		–	–	–	✓
	4	P0IE4	0	H0	R/W		–	–	–	✓
	3	P0IE3	0	H0	R/W		✓	✓	✓	✓
	2	P0IE2	0	H0	R/W		✓	✓	✓	✓
	1	P0IE1	0	H0	R/W		✓	✓	✓	✓
	0	P0IE0	0	H0	R/W		✓	✓	✓	✓
POCHATEN (P0 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	POCHATEN7	0	H0	R/W	–	–	–	–	✓
	6	POCHATEN6	0	H0	R/W		–	–	–	✓
	5	POCHATEN5	0	H0	R/W		–	–	–	✓
	4	POCHATEN4	0	H0	R/W		–	–	–	✓
	3	POCHATEN3	0	H0	R/W		✓	✓	✓	✓
	2	POCHATEN2	0	H0	R/W		✓	✓	✓	✓
	1	POCHATEN1	0	H0	R/W		✓	✓	✓	✓
	0	POCHATEN0	0	H0	R/W		✓	✓	✓	✓

6 I/O PORTS (PPORT)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
P0MODESEL (P0 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P0SEL7	0	H0	R/W	–	–	–	–	✓
	6	P0SEL6	0	H0	R/W	–	–	–	–	✓
	5	P0SEL5	0	H0	R/W	–	–	–	–	✓
	4	P0SEL4	0	H0	R/W	–	–	–	–	✓
	3	P0SEL3	0	H0	R/W	–	✓	✓	✓	✓
	2	P0SEL2	0	H0	R/W	–	✓	✓	✓	✓
	1	P0SEL1	0	H0	R/W	–	✓	✓	✓	✓
	0	P0SEL0	0	H0	R/W	–	✓	✓	✓	✓
P0FNCSSEL (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								M20/M23		M21/	M22/
	GPIO	P0yMUX = 0x0 (Function 0)		P0yMUX = 0x1 (Function 1)		P0yMUX = 0x2 (Function 2)		P0yMUX = 0x3 (Function 3)			24pin	32pin	M24	M25
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin					
P00	P00	T16B Ch.0	EXCL00	UPMUX	*1	–	–	–	–	–	✓	✓	✓	✓
P01	P01	T16B Ch.0	EXCL01	UPMUX	*1	–	–	–	–	–	✓	✓	✓	✓
P02	P02	SNDA	BZOUT	UPMUX	*1	–	–	–	–	–	✓	✓	✓	✓
P03	P03	SNDA	#BZOUT	UPMUX	*1	–	–	–	–	–	✓	✓	✓	✓
P04	P04	RFC Ch.0	RFCLK00	UPMUX	*1	–	–	–	–	–	–	–	–	✓
P05	P05	RFC Ch.1	RFCLK01	UPMUX	*1	–	–	–	–	–	–	–	–	✓
P06	P06	–	–	UPMUX	*1	–	–	–	–	–	–	–	–	✓
P07	P07	–	–	UPMUX	*1	–	–	–	–	–	–	–	–	✓

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

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	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
P1DAT (P1 Port Data Register)	15	P1OUT7	0	H0	R/W	–	–	–	–	✓
	14	P1OUT6	0	H0	R/W	–	–	–	–	✓
	13	P1OUT5	0	H0	R/W	–	✓	✓	✓	✓
	12	P1OUT4	0	H0	R/W	–	✓	✓	✓	✓
	11	P1OUT3	0	H0	R/W	–	✓	✓	✓	✓
	10	P1OUT2	0	H0	R/W	–	✓	✓	✓	✓
	9	P1OUT1	0	H0	R/W	–	–	✓	✓	✓
	8	P1OUT0	0	H0	R/W	–	–	✓	✓	✓
	7	P1IN7	0	H0	R	–	–	–	–	✓
	6	P1IN6	0	H0	R	–	–	–	–	✓
	5	P1IN5	0	H0	R	–	✓	✓	✓	✓
	4	P1IN4	0	H0	R	–	✓	✓	✓	✓
	3	P1IN3	0	H0	R	–	✓	✓	✓	✓
	2	P1IN2	0	H0	R	–	✓	✓	✓	✓
	1	P1IN1	0	H0	R	–	–	✓	✓	✓
	0	P1IN0	0	H0	R	–	–	✓	✓	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23 24pin	M21/ 32pin	M22/ M24	M25
P1IOEN (P1 Port Enable Register)	15	P1IEN7	0	H0	R/W	–	–	–	–	✓
	14	P1IEN6	0	H0	R/W		–	–	–	✓
	13	P1IEN5	0	H0	R/W		✓	✓	✓	✓
	12	P1IEN4	0	H0	R/W		✓	✓	✓	✓
	11	P1IEN3	0	H0	R/W		✓	✓	✓	✓
	10	P1IEN2	0	H0	R/W		✓	✓	✓	✓
	9	P1IEN1	0	H0	R/W		–	✓	✓	✓
	8	P1IEN0	0	H0	R/W		–	✓	✓	✓
	7	P1OEN7	0	H0	R/W	–	–	–	–	✓
	6	P1OEN6	0	H0	R/W		–	–	–	✓
	5	P1OEN5	0	H0	R/W		✓	✓	✓	✓
	4	P1OEN4	0	H0	R/W		✓	✓	✓	✓
	3	P1OEN3	0	H0	R/W		✓	✓	✓	✓
	2	P1OEN2	0	H0	R/W		✓	✓	✓	✓
	1	P1OEN1	0	H0	R/W		–	✓	✓	✓
	0	P1OEN0	0	H0	R/W		–	✓	✓	✓
P1RCTL (P1 Port Pull-up/down Control Register)	15	P1PDPU7	0	H0	R/W	–	–	–	–	✓
	14	P1PDPU6	0	H0	R/W		–	–	–	✓
	13	P1PDPU5	0	H0	R/W		✓	✓	✓	✓
	12	P1PDPU4	0	H0	R/W		✓	✓	✓	✓
	11	P1PDPU3	0	H0	R/W		✓	✓	✓	✓
	10	P1PDPU2	0	H0	R/W		✓	✓	✓	✓
	9	P1PDPU1	0	H0	R/W		–	✓	✓	✓
	8	P1PDPU0	0	H0	R/W		–	✓	✓	✓
	7	P1REN7	0	H0	R/W	–	–	–	–	✓
	6	P1REN6	0	H0	R/W		–	–	–	✓
	5	P1REN5	0	H0	R/W		✓	✓	✓	✓
	4	P1REN4	0	H0	R/W		✓	✓	✓	✓
	3	P1REN3	0	H0	R/W		✓	✓	✓	✓
	2	P1REN2	0	H0	R/W		✓	✓	✓	✓
	1	P1REN1	0	H0	R/W		–	✓	✓	✓
	0	P1REN0	0	H0	R/W		–	✓	✓	✓
P1INTF (P1 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P1IF7	0	H0	R/W	Cleared by writing 1.	–	–	–	✓
	6	P1IF6	0	H0	R/W		–	–	–	✓
	5	P1IF5	0	H0	R/W		✓	✓	✓	✓
	4	P1IF4	0	H0	R/W		✓	✓	✓	✓
	3	P1IF3	0	H0	R/W		✓	✓	✓	✓
	2	P1IF2	0	H0	R/W		✓	✓	✓	✓
	1	P1IF1	0	H0	R/W		–	✓	✓	✓
	0	P1IF0	0	H0	R/W		–	✓	✓	✓
P1INTCTL (P1 Port Interrupt Control Register)	15	P1EDGE7	0	H0	R/W	–	–	–	–	✓
	14	P1EDGE6	0	H0	R/W		–	–	–	✓
	13	P1EDGE5	0	H0	R/W		✓	✓	✓	✓
	12	P1EDGE4	0	H0	R/W		✓	✓	✓	✓
	11	P1EDGE3	0	H0	R/W		✓	✓	✓	✓
	10	P1EDGE2	0	H0	R/W		✓	✓	✓	✓
	9	P1EDGE1	0	H0	R/W		–	✓	✓	✓
	8	P1EDGE0	0	H0	R/W		–	✓	✓	✓
	7	P1IE7	0	H0	R/W	–	–	–	–	✓
	6	P1IE6	0	H0	R/W		–	–	–	✓
	5	P1IE5	0	H0	R/W		✓	✓	✓	✓
	4	P1IE4	0	H0	R/W		✓	✓	✓	✓
	3	P1IE3	0	H0	R/W		✓	✓	✓	✓
	2	P1IE2	0	H0	R/W		✓	✓	✓	✓
	1	P1IE1	0	H0	R/W		–	✓	✓	✓
	0	P1IE0	0	H0	R/W		–	✓	✓	✓

6 I/O PORTS (PPORT)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
P1CHATEN (P1 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P1CHATEN7	0	H0	R/W	–	–	–	–	✓
	6	P1CHATEN6	0	H0	R/W	–	–	–	–	✓
	5	P1CHATEN5	0	H0	R/W	–	✓	✓	✓	✓
	4	P1CHATEN4	0	H0	R/W	–	✓	✓	✓	✓
	3	P1CHATEN3	0	H0	R/W	–	✓	✓	✓	✓
	2	P1CHATEN2	0	H0	R/W	–	✓	✓	✓	✓
	1	P1CHATEN1	0	H0	R/W	–	–	✓	✓	✓
	0	P1CHATEN0	0	H0	R/W	–	–	✓	✓	✓
P1MODESEL (P1 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P1SEL7	0	H0	R/W	–	–	–	–	✓
	6	P1SEL6	0	H0	R/W	–	–	–	–	✓
	5	P1SEL5	0	H0	R/W	–	✓	✓	✓	✓
	4	P1SEL4	0	H0	R/W	–	✓	✓	✓	✓
	3	P1SEL3	0	H0	R/W	–	✓	✓	✓	✓
	2	P1SEL2	0	H0	R/W	–	✓	✓	✓	✓
	1	P1SEL1	0	H0	R/W	–	–	✓	✓	✓
	0	P1SEL0	0	H0	R/W	–	–	✓	✓	✓
P1FNCSSEL (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								M20/M23		M21/	M22/
	GPIO	P1yMUX = 0x0 (Function 0)		P1yMUX = 0x1 (Function 1)		P1yMUX = 0x2 (Function 2)		P1yMUX = 0x3 (Function 3)			24pin	32pin		
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin					
P10	P10	–	–	UPMUX	*1	–	–	–	–	–	–	✓	✓	✓
P11	P11	–	–	UPMUX	*1	–	–	–	–	–	–	✓	✓	✓
P12	P12	REMC3	REMO	UPMUX	*1	–	–	–	–	–	✓	✓	✓	✓
P13	P13	CLG	FOUT	UPMUX	*1	–	–	–	–	–	✓	✓	✓	✓
P14	P14	ADC12A	#ADTRG0	UPMUX	*1	–	–	–	–	–	✓	✓	✓	✓
P15	P15	REMC3	CLPLS	UPMUX	*1	–	–	–	–	–	✓	✓	✓	✓
P16	P16	–	–	UPMUX	*1	–	–	–	–	–	–	–	–	✓
P17	P17	–	–	UPMUX	*1	–	–	–	–	–	–	–	–	✓

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

6.7.3 P2 Port Group

The P2 port group support 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	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
P2DAT (P2 Port Data Register)	15	P2OUT7	0	H0	R/W	–	✓	✓	✓	✓
	14	P2OUT6	0	H0	R/W		✓	✓	✓	✓
	13	P2OUT5	0	H0	R/W		✓	✓	✓	✓
	12	P2OUT4	0	H0	R/W		✓	✓	✓	✓
	11	P2OUT3	0	H0	R/W		–	✓	✓	✓
	10	P2OUT2	0	H0	R/W		–	✓	✓	✓
	9	P2OUT1	0	H0	R/W		–	–	–	✓
	8	P2OUT0	0	H0	R/W		–	–	–	✓
	7	P2IN7	0	H0	R	–	✓	✓	✓	✓
	6	P2IN6	0	H0	R		✓	✓	✓	✓
	5	P2IN5	0	H0	R		✓	✓	✓	✓
	4	P2IN4	0	H0	R		✓	✓	✓	✓
	3	P2IN3	0	H0	R		–	✓	✓	✓
	2	P2IN2	0	H0	R		–	✓	✓	✓
	1	P2IN1	0	H0	R		–	–	–	✓
	0	P2IN0	0	H0	R		–	–	–	✓
P2IOEN (P2 Port Enable Register)	15	P2IEN7	0	H0	R/W	–	✓	✓	✓	✓
	14	P2IEN6	0	H0	R/W		✓	✓	✓	✓
	13	P2IEN5	0	H0	R/W		✓	✓	✓	✓
	12	P2IEN4	0	H0	R/W		✓	✓	✓	✓
	11	P2IEN3	0	H0	R/W		–	✓	✓	✓
	10	P2IEN2	0	H0	R/W		–	✓	✓	✓
	9	P2IEN1	0	H0	R/W		–	–	–	✓
	8	P2IEN0	0	H0	R/W		–	–	–	✓
	7	P2OEN7	0	H0	R/W	–	✓	✓	✓	✓
	6	P2OEN6	0	H0	R/W		✓	✓	✓	✓
	5	P2OEN5	0	H0	R/W		✓	✓	✓	✓
	4	P2OEN4	0	H0	R/W		✓	✓	✓	✓
	3	P2OEN3	0	H0	R/W		–	✓	✓	✓
	2	P2OEN2	0	H0	R/W		–	✓	✓	✓
	1	P2OEN1	0	H0	R/W		–	–	–	✓
	0	P2OEN0	0	H0	R/W		–	–	–	✓
P2RCTL (P2 Port Pull-up/down Control Register)	15	P2PDPU7	0	H0	R/W	–	✓	✓	✓	✓
	14	P2PDPU6	0	H0	R/W		✓	✓	✓	✓
	13	P2PDPU5	0	H0	R/W		✓	✓	✓	✓
	12	P2PDPU4	0	H0	R/W		✓	✓	✓	✓
	11	P2PDPU3	0	H0	R/W		–	✓	✓	✓
	10	P2PDPU2	0	H0	R/W		–	✓	✓	✓
	9	P2PDPU1	0	H0	R/W		–	–	–	✓
	8	P2PDPU0	0	H0	R/W		–	–	–	✓
	7	P2REN7	0	H0	R/W	–	✓	✓	✓	✓
	6	P2REN6	0	H0	R/W		✓	✓	✓	✓
	5	P2REN5	0	H0	R/W		✓	✓	✓	✓
	4	P2REN4	0	H0	R/W		✓	✓	✓	✓
	3	P2REN3	0	H0	R/W		–	✓	✓	✓
	2	P2REN2	0	H0	R/W		–	✓	✓	✓
	1	P2REN1	0	H0	R/W		–	–	–	✓
	0	P2REN0	0	H0	R/W		–	–	–	✓

6 I/O PORTS (PPORT)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
P2INTF (P2 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P2IF7	0	H0	R/W	Cleared by writing 1.	✓	✓	✓	✓
	6	P2IF6	0	H0	R/W		✓	✓	✓	✓
	5	P2IF5	0	H0	R/W		✓	✓	✓	✓
	4	P2IF4	0	H0	R/W		✓	✓	✓	✓
	3	P2IF3	0	H0	R/W		–	✓	✓	✓
	2	P2IF2	0	H0	R/W		–	✓	✓	✓
	1	P2IF1	0	H0	R/W		–	–	–	✓
	0	P2IF0	0	H0	R/W		–	–	–	✓
P2INTCTL (P2 Port Interrupt Control Register)	15	P2EDGE7	0	H0	R/W	–	✓	✓	✓	✓
	14	P2EDGE6	0	H0	R/W		✓	✓	✓	✓
	13	P2EDGE5	0	H0	R/W		✓	✓	✓	✓
	12	P2EDGE4	0	H0	R/W		✓	✓	✓	✓
	11	P2EDGE3	0	H0	R/W		–	✓	✓	✓
	10	P2EDGE2	0	H0	R/W		–	✓	✓	✓
	9	P2EDGE1	0	H0	R/W		–	–	–	✓
	8	P2EDGE0	0	H0	R/W		–	–	–	✓
	7	P2IE7	0	H0	R/W	–	✓	✓	✓	✓
	6	P2IE6	0	H0	R/W		✓	✓	✓	✓
	5	P2IE5	0	H0	R/W		✓	✓	✓	✓
	4	P2IE4	0	H0	R/W		✓	✓	✓	✓
	3	P2IE3	0	H0	R/W		–	✓	✓	✓
	2	P2IE2	0	H0	R/W		–	✓	✓	✓
	1	P2IE1	0	H0	R/W		–	–	–	✓
	0	P2IE0	0	H0	R/W		–	–	–	✓
P2CHATEN (P2 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P2CHATEN7	0	H0	R/W	–	✓	✓	✓	✓
	6	P2CHATEN6	0	H0	R/W		✓	✓	✓	✓
	5	P2CHATEN5	0	H0	R/W		✓	✓	✓	✓
	4	P2CHATEN4	0	H0	R/W		✓	✓	✓	✓
	3	P2CHATEN3	0	H0	R/W		–	✓	✓	✓
	2	P2CHATEN2	0	H0	R/W		–	✓	✓	✓
	1	P2CHATEN1	0	H0	R/W		–	–	–	✓
	0	P2CHATEN0	0	H0	R/W		–	–	–	✓
P2MODSEL (P2 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P2SEL7	0	H0	R/W	–	✓	✓	✓	✓
	6	P2SEL6	0	H0	R/W		✓	✓	✓	✓
	5	P2SEL5	0	H0	R/W		✓	✓	✓	✓
	4	P2SEL4	0	H0	R/W		✓	✓	✓	✓
	3	P2SEL3	0	H0	R/W		–	✓	✓	✓
	2	P2SEL2	0	H0	R/W		–	✓	✓	✓
	1	P2SEL1	0	H0	R/W		–	–	–	✓
	0	P2SEL0	0	H0	R/W		–	–	–	✓
P2FNCSSEL (P2 Port Function Select Register)	15–14	P27MUX[1:0]	0x0	H0	R/W	–	✓	✓	✓	✓
	13–12	P26MUX[1:0]	0x0	H0	R/W		✓	✓	✓	✓
	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	P2SELY = 0 GPIO	P2SELY = 1								M20/M23		M21/ M24	M22/ M25
		P2yMUX = 0x0 (Function 0)		P2yMUX = 0x1 (Function 1)		P2yMUX = 0x2 (Function 2)		P2yMUX = 0x3 (Function 3)					
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	24pin	32pin		
P20	P20	—	—	UPMUX	*1	ADC12A	ADIN07	—	—	—	—	—	✓
P21	P21	—	—	UPMUX	*1	ADC12A	ADIN06	—	—	—	—	—	✓
P22	P22	—	—	UPMUX	*1	ADC12A	ADIN05	—	—	—	✓	✓	✓
P23	P23	—	—	UPMUX	*1	ADC12A	ADIN04	—	—	—	✓	✓	✓
P24	P24	T16B Ch.1	EXCL10	UPMUX	*1	ADC12A	ADIN03	—	—	✓	✓	✓	✓
P25	P25	T16B Ch.1	EXCL11	UPMUX	*1	ADC12A	ADIN02	—	—	✓	✓	✓	✓
P26	P26	—	—	UPMUX	*1	ADC12A	ADIN01	—	—	✓	✓	✓	✓
P27	P27	—	—	UPMUX	*1	ADC12A	ADIN00	—	—	✓	✓	✓	✓

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

6.7.4 P3 Port Group

The P3 port group supports the GPIO and interrupt functions.

Table 6.7.4.1 Control Registers for P3 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/M24	M22/M25
							24pin	32pin		
P3DAT (P3 Port Data Register)	15	P3OUT7	0	H0	R/W	—	—	—	—	✓
	14	P3OUT6	0	H0	R/W		—	—	—	✓
	13	P3OUT5	0	H0	R/W		—	—	—	✓
	12	P3OUT4	0	H0	R/W		—	—	—	✓
	11	P3OUT3	0	H0	R/W		—	—	—	✓
	10	P3OUT2	0	H0	R/W		✓	✓	✓	✓
	9	P3OUT1	0	H0	R/W		✓	✓	✓	✓
	8	P3OUT0	0	H0	R/W		✓	✓	✓	✓
	7	P3IN7	0	H0	R	—	—	—	—	✓
	6	P3IN6	0	H0	R		—	—	—	✓
	5	P3IN5	0	H0	R		—	—	—	✓
	4	P3IN4	0	H0	R		—	—	—	✓
	3	P3IN3	0	H0	R		—	—	—	✓
	2	P3IN2	0	H0	R		✓	✓	✓	✓
	1	P3IN1	0	H0	R		✓	✓	✓	✓
	0	P3IN0	0	H0	R		✓	✓	✓	✓
P3IOEN (P3 Port Enable Register)	15	P3IEN7	0	H0	R/W	—	—	—	—	✓
	14	P3IEN6	0	H0	R/W		—	—	—	✓
	13	P3IEN5	0	H0	R/W		—	—	—	✓
	12	P3IEN4	0	H0	R/W		—	—	—	✓
	11	P3IEN3	0	H0	R/W		—	—	—	✓
	10	P3IEN2	0	H0	R/W		✓	✓	✓	✓
	9	P3IEN1	0	H0	R/W		✓	✓	✓	✓
	8	P3IEN0	0	H0	R/W		✓	✓	✓	✓
	7	P3OEN7	0	H0	R/W	—	—	—	—	✓
	6	P3OEN6	0	H0	R/W		—	—	—	✓
	5	P3OEN5	0	H0	R/W		—	—	—	✓
	4	P3OEN4	0	H0	R/W		—	—	—	✓
	3	P3OEN3	0	H0	R/W		—	—	—	✓
	2	P3OEN2	0	H0	R/W		✓	✓	✓	✓
	1	P3OEN1	0	H0	R/W		✓	✓	✓	✓
	0	P3OEN0	0	H0	R/W		✓	✓	✓	✓

6 I/O PORTS (PPORT)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/M22/	
							24pin	32pin	M24	M25
P3RCTL (P3 Port Pull-up/down Control Register)	15	P3PDPU7	0	H0	R/W	–	–	–	–	✓
	14	P3PDPU6	0	H0	R/W		–	–	–	✓
	13	P3PDPU5	0	H0	R/W		–	–	–	✓
	12	P3PDPU4	0	H0	R/W		–	–	–	✓
	11	P3PDPU3	0	H0	R/W		–	–	–	✓
	10	P3PDPU2	0	H0	R/W		✓	✓	✓	✓
	9	P3PDPU1	0	H0	R/W		✓	✓	✓	✓
	8	P3PDPU0	0	H0	R/W		✓	✓	✓	✓
	7	P3REN7	0	H0	R/W	–	–	–	–	✓
	6	P3REN6	0	H0	R/W		–	–	–	✓
	5	P3REN5	0	H0	R/W		–	–	–	✓
	4	P3REN4	0	H0	R/W		–	–	–	✓
	3	P3REN3	0	H0	R/W		–	–	–	✓
	2	P3REN2	0	H0	R/W		✓	✓	✓	✓
	1	P3REN1	0	H0	R/W		✓	✓	✓	✓
	0	P3REN0	0	H0	R/W		✓	✓	✓	✓
P3INTF (P3 Port Interrupt Flag Register)	15–8	–	0x00	–	R	– Cleared by writing 1.	–	–	–	–
	7	P3IF7	0	H0	R/W		–	–	–	✓
	6	P3IF6	0	H0	R/W		–	–	–	✓
	5	P3IF5	0	H0	R/W		–	–	–	✓
	4	P3IF4	0	H0	R/W		–	–	–	✓
	3	P3IF3	0	H0	R/W		–	–	–	✓
	2	P3IF2	0	H0	R/W		✓	✓	✓	✓
	1	P3IF1	0	H0	R/W		✓	✓	✓	✓
	0	P3IF0	0	H0	R/W		✓	✓	✓	✓
P3INTCTL (P3 Port Interrupt Control Register)	15	P3EDGE7	0	H0	R/W	–	–	–	–	✓
	14	P3EDGE6	0	H0	R/W		–	–	–	✓
	13	P3EDGE5	0	H0	R/W		–	–	–	✓
	12	P3EDGE4	0	H0	R/W		–	–	–	✓
	11	P3EDGE3	0	H0	R/W		–	–	–	✓
	10	P3EDGE2	0	H0	R/W		✓	✓	✓	✓
	9	P3EDGE1	0	H0	R/W		✓	✓	✓	✓
	8	P3EDGE0	0	H0	R/W		✓	✓	✓	✓
	7	P3IE7	0	H0	R/W	–	–	–	–	✓
	6	P3IE6	0	H0	R/W		–	–	–	✓
	5	P3IE5	0	H0	R/W		–	–	–	✓
	4	P3IE4	0	H0	R/W		–	–	–	✓
	3	P3IE3	0	H0	R/W		–	–	–	✓
	2	P3IE2	0	H0	R/W		✓	✓	✓	✓
	1	P3IE1	0	H0	R/W		✓	✓	✓	✓
	0	P3IE0	0	H0	R/W		✓	✓	✓	✓
P3CHATEN (P3 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P3CHATEN7	0	H0	R/W		–	–	–	✓
	6	P3CHATEN6	0	H0	R/W		–	–	–	✓
	5	P3CHATEN5	0	H0	R/W		–	–	–	✓
	4	P3CHATEN4	0	H0	R/W		–	–	–	✓
	3	P3CHATEN3	0	H0	R/W		–	–	–	✓
	2	P3CHATEN2	0	H0	R/W		✓	✓	✓	✓
	1	P3CHATEN1	0	H0	R/W		✓	✓	✓	✓
	0	P3CHATEN0	0	H0	R/W		✓	✓	✓	✓

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
P3MODESEL (P3 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7	P3SEL7	0	H0	R/W	–	–	–	–	✓
	6	P3SEL6	0	H0	R/W	–	–	–	–	✓
	5	P3SEL5	0	H0	R/W	–	–	–	–	✓
	4	P3SEL4	0	H0	R/W	–	–	–	–	✓
	3	P3SEL3	0	H0	R/W	–	–	–	–	✓
	2	P3SEL2	0	H0	R/W	–	✓	✓	✓	✓
	1	P3SEL1	0	H0	R/W	–	✓	✓	✓	✓
	0	P3SEL0	0	H0	R/W	–	✓	✓	✓	✓
P3FNCSSEL (P3 Port Function Select Register)	15–14	P37MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
	13–12	P36MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
	11–10	P35MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
	9–8	P34MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
	7–6	P33MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
	5–4	P32MUX[1:0]	0x0	H0	R/W	–	✓	✓	✓	✓
	3–2	P31MUX[1:0]	0x0	H0	R/W	–	✓	✓	✓	✓
	1–0	P30MUX[1:0]	0x0	H0	R/W	–	✓	✓	✓	✓

Table 6.7.4.2 P3 Port Group Function Assignment

Port name	GPIO	P3SELy = 0		P3SELy = 1						M20/M23		M21/ M24	M22/ M25
		P3yMUX = 0x0 (Function 0)		P3yMUX = 0x1 (Function 1)		P3yMUX = 0x2 (Function 2)		P3yMUX = 0x3 (Function 3)					
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	24pin	32pin		
P30	P30	—	—	UPMUX	*1	ADC12A	VREFA0	—	—	✓	✓	✓	✓
P31	P31	CLG	EXOSC	UPMUX	*1	—	—	—	—	✓	✓	✓	✓
P32	P32	RTCA	RTC1S	UPMUX	*1	SVD3	EXSVD0	—	—	✓	✓	✓	✓
P33	P33	RFC Ch.0	SENB0	UPMUX	*1	—	—	—	—	—	—	—	✓
P34	P34	RFC Ch.0	SENA0	UPMUX	*1	—	—	—	—	—	—	—	✓
P35	P35	RFC Ch.0	REF0	UPMUX	*1	—	—	—	—	—	—	—	✓
P36	P36	RFC Ch.0	RFIN0	UPMUX	*1	—	—	—	—	—	—	—	✓
P37	P37	RFC Ch.1	SENB1	UPMUX	*1	—	—	—	—	—	—	—	✓

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

6.7.5 P4 Port Group

The P4 port group supports the GPIO and interrupt functions.

Table 6.7.5.1 Control Registers for P4 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
P4DAT (P4 Port Data Register)	15–11	–	0x00	–	R	–	–	–	–	–
	10	P4OUT2	0	H0	R/W	–	–	–	–	✓
	9	P4OUT1	0	H0	R/W	–	–	–	–	✓
	8	P4OUT0	0	H0	R/W	–	–	–	–	✓
	7–3	–	0x00	–	R	–	–	–	–	–
	2	P4IN2	0	H0	R	–	–	–	–	✓
	1	P4IN1	0	H0	R	–	–	–	–	✓
	0	P4IN0	0	H0	R	–	–	–	–	✓
P4IOEN (P4 Port Enable Register)	15–11	–	0x00	–	R	–	–	–	–	–
	10	P4IEN2	0	H0	R/W	–	–	–	–	✓
	9	P4IEN1	0	H0	R/W	–	–	–	–	✓
	8	P4IEN0	0	H0	R/W	–	–	–	–	✓
	7–3	–	0x00	–	R	–	–	–	–	–
	2	P4OEN2	0	H0	R/W	–	–	–	–	✓
	1	P4OEN1	0	H0	R/W	–	–	–	–	✓
	0	P4OEN0	0	H0	R/W	–	–	–	–	✓

6 I/O PORTS (PPORT)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/ M24	M22/ M25
							24pin	32pin		
P4RCTL (P4 Port Pull-up/down Control Register)	15–11	–	0x00	–	R	–	–	–	–	–
	10	P4PDPJ2	0	H0	R/W	–	–	–	–	✓
	9	P4PDPJ1	0	H0	R/W	–	–	–	–	✓
	8	P4PDPJ0	0	H0	R/W	–	–	–	–	✓
	7–3	–	0x00	–	R	–	–	–	–	–
	2	P4REN2	0	H0	R/W	–	–	–	–	✓
	1	P4REN1	0	H0	R/W	–	–	–	–	✓
	0	P4REN0	0	H0	R/W	–	–	–	–	✓
P4INTF (P4 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7–3	–	0x00	–	R	–	–	–	–	–
	2	P4IF2	0	H0	R/W	Cleared by writing 1.	–	–	–	✓
	1	P4IF1	0	H0	R/W		–	–	–	✓
	0	P4IF0	0	H0	R/W		–	–	–	✓
P4INTCTL (P4 Port Interrupt Control Register)	15–11	–	0x00	–	R	–	–	–	–	–
	10	P4EDGE2	0	H0	R/W	–	–	–	–	✓
	9	P4EDGE1	0	H0	R/W	–	–	–	–	✓
	8	P4EDGE0	0	H0	R/W	–	–	–	–	✓
	7–3	–	0x00	–	R	–	–	–	–	–
	2	P4IE2	0	H0	R/W	–	–	–	–	✓
	1	P4IE1	0	H0	R/W	–	–	–	–	✓
	0	P4IE0	0	H0	R/W	–	–	–	–	✓
P4CHATEN (P4 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7–3	–	0x00	–	R	–	–	–	–	–
	2	P4CHATEN2	0	H0	R/W	–	–	–	–	✓
	1	P4CHATEN1	0	H0	R/W	–	–	–	–	✓
	0	P4CHATEN0	0	H0	R/W	–	–	–	–	✓
P4MODSEL (P4 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7–3	–	0x00	–	R	–	–	–	–	–
	2	P4SEL2	0	H0	R/W	–	–	–	–	✓
	1	P4SEL1	0	H0	R/W	–	–	–	–	✓
	0	P4SEL0	0	H0	R/W	–	–	–	–	✓
P4FNCSSEL (P4 Port Function Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7–6	–	0x0	–	R	–	–	–	–	–
	5–4	P42MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
	3–2	P41MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
	1–0	P40MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓

Table 6.7.5.2 P4 Port Group Function Assignment

Port name	P4SELY = 0	P4SELY = 1								M20/M23		M21/ M24	M22/ M25
	GPIO	P4yMUX = 0x0 (Function 0)		P4yMUX = 0x1 (Function 1)		P4yMUX = 0x2 (Function 2)		P4yMUX = 0x3 (Function 3)					
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	24pin	32pin		
P40	P40	RFC Ch.1	SENA1	–	–	–	–	–	–	–	–	–	✓
P41	P41	RFC Ch.1	REF1	–	–	–	–	–	–	–	–	–	✓
P42	P42	RFC Ch.1	RFIN1	–	–	–	–	–	–	–	–	–	✓

6.7.6 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 6.7.6.1 Control Registers for Pd Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
PDDAT (Pd Port Data Register)	15–13	–	0x0	–	R	–	–	–	–	–
	12	PDOUT4	0	H0	R/W	–	–	✓	✓	✓
	11	PDOUT3	0	H0	R/W	–	–	✓	✓	✓
	10	PDOUT2	0	H0	R/W	–	✓	✓	✓	✓
	9	PDOUT1	0	H0	R/W	–	✓	✓	✓	✓
	8	PDOUT0	0	H0	R/W	–	✓	✓	✓	✓
	7–5	–	0	–	R	–	–	–	–	–
	4	PDIN4	X	H0	R	–	–	✓	✓	✓
	3	PDIN3	X	H0	R	–	–	✓	✓	✓
	2	–	0	–	R	–	–	–	–	–
	1	PDIN1	X	H0	R	–	✓	✓	✓	✓
	0	PDIN0	X	H0	R	–	✓	✓	✓	✓
PDIOEN (Pd Port Enable Register)	15–13	–	0x0	–	R	–	–	–	–	–
	12	PDIEN4	0	H0	R/W	–	–	✓	✓	✓
	11	PDIEN3	0	H0	R/W	–	–	✓	✓	✓
	10	(reserved)	0	H0	R/W	–	✓	✓	✓	✓
	9	PDIEN1	0	H0	R/W	–	✓	✓	✓	✓
	8	PDIEN0	0	H0	R/W	–	✓	✓	✓	✓
	7–5	–	0	–	R	–	–	–	–	–
	4	PDOEN4	0	H0	R/W	–	–	✓	✓	✓
	3	PDOEN3	0	H0	R/W	–	–	✓	✓	✓
	2	PDOEN2	0	H0	R/W	–	✓	✓	✓	✓
	1	PDOEN1	0	H0	R/W	–	✓	✓	✓	✓
	0	PDOEN0	0	H0	R/W	–	✓	✓	✓	✓
PDRCTL (Pd Port Pull-up/down Control Register)	15–13	–	0x0	–	R	–	–	–	–	–
	12	PDPDPU4	0	H0	R/W	–	–	✓	✓	✓
	11	PDPDPU3	0	H0	R/W	–	–	✓	✓	✓
	10	(reserved)	0	H0	R/W	–	✓	✓	✓	✓
	9	PDPDPU1	0	H0	R/W	–	✓	✓	✓	✓
	8	PDPDPU0	0	H0	R/W	–	✓	✓	✓	✓
	7–5	–	0	–	R	–	–	–	–	–
	4	PDREN4	0	H0	R/W	–	–	✓	✓	✓
	3	PDREN3	0	H0	R/W	–	–	✓	✓	✓
	2	(reserved)	0	H0	R/W	–	✓	✓	✓	✓
	1	PDREN1	0	H0	R/W	–	✓	✓	✓	✓
	0	PDREN0	0	H0	R/W	–	✓	✓	✓	✓
PDINTF PDINTCTL PDCHATEN	15–0	–	0x0000	–	R	–	–	–	–	–
PDMODESEL (Pd Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7–5	–	0	–	R	–	–	–	–	–
	4	PDSEL4	0	H0	R/W	–	–	✓	✓	✓
	3	PDSEL3	0	H0	R/W	–	–	✓	✓	✓
	2	PDSEL2	1	H0	R/W	–	✓	✓	✓	✓
	1	PDSEL1	1	H0	R/W	–	✓	✓	✓	✓
	0	PDSEL0	1	H0	R/W	–	✓	✓	✓	✓

6 I/O PORTS (PPORT)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
PDFNCSEL (Pd Port Function Select Register)	15–10	–	0x00	–	R	–	–	–	–	–
	9–8	PD4MUX[1:0]	0x0	H0	R/W	–	–	✓	✓	✓
	7–6	PD3MUX[1:0]	0x0	H0	R/W	–	–	✓	✓	✓
	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	–	✓	✓	✓	✓

Table 6.7.6.2 Pd Port Group Function Assignment

Port name	PDSELY = 0		PDSELY = 1								M20/M23		M21/	M22/
	GPIO	PDyMUX = 0x0 (Function 0)		PDyMUX = 0x1 (Function 1)		PDyMUX = 0x2 (Function 2)		PDyMUX = 0x3 (Function 3)			24pin	32pin	M24	M25
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin					
Pd0	PD0	DBG	DST2	–	–	–	–	–	–	✓	✓	✓	✓	✓
Pd1	PD1	DBG	DSIO	–	–	–	–	–	–	✓	✓	✓	✓	✓
Pd2	PD2	DBG	DCLK	–	–	–	–	–	–	✓	✓	✓	✓	✓
Pd3	PD3	–	–	–	–	CLG	OSC3	–	–	–	✓	✓	✓	✓
Pd4	PD4	–	–	–	–	CLG	OSC4	–	–	–	✓	✓	✓	✓

6.7.7 Common Registers between Port Groups

Table 6.7.7.1 Control Registers for Common Use with Port Groups

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
							24pin	32pin	M24	M25
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	–	✓	✓	✓	✓
PINTFRP (P Port Interrupt Flag Group Register)	15–8	–	0x00	–	R	–	–	–	–	–
	7–5	–	0x0	–	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	–	✓	✓	✓	✓

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, P1, P2, and P3 port groups.
- The peripheral I/O function assigned via UPMUX is enabled by setting the PxFNCSSEL.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, 2, 3) and 'y' refers to a port number (y = 0, 1, 2, ..., 7).

Figure 7.1.1 shows the configuration of UPMUX.

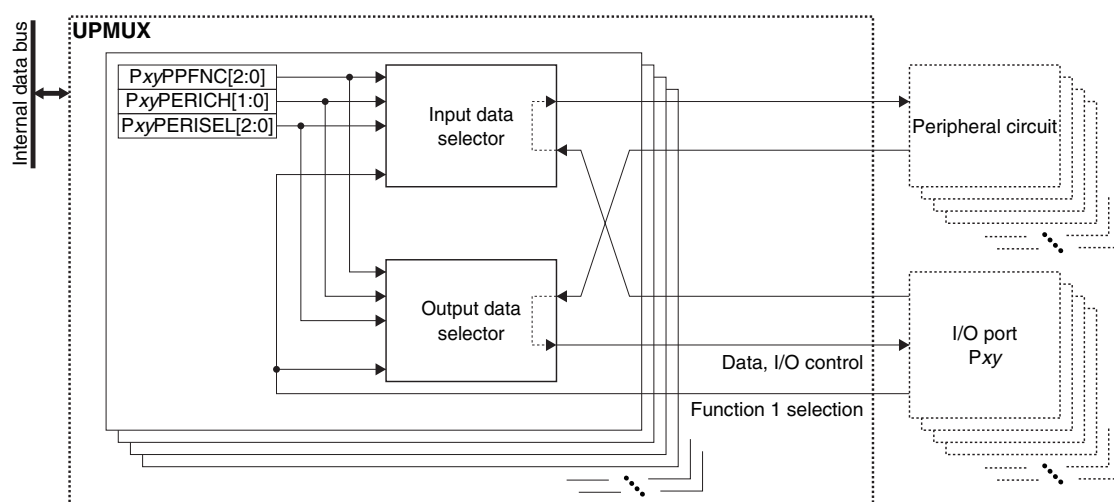


Figure 7.1.1 UPMUX Configuration

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 PxMODSEL.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 PxFNCSSEL.PxyMUX[1:0] bits of the I/O port to 0x1. (Select peripheral I/O function 1)
6. Set the PxMODSEL.PxSELY bit of the I/O port to 1. (Enable peripheral I/O function)

7.3 Control Registers

Pxy-xz Universal Port Multiplexer Setting Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PxUPMUXn	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 <i>n</i> . PxyPPFNC[2:0] bits (Peripheral I/O function)	PxUPMUX <i>n</i> .PxyPERISEL[2:0] bits (Peripheral circuit)								
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	
	None *	I2C	SPIA	UART3	T16B	Reserved	Reserved	Reserved	
	PxUPMUX <i>n</i> .PxyPERICH[1:0] bits (Peripheral circuit channel)								
	–	0x0	0x0, 0x1	0x0, 0x1	0x0, 0x1	–	–	–	
–	Ch.0	Ch.0, 1	Ch.0, 1	Ch.0, 1	Ch.0, 1	–	–	–	
0x0	None *	None *	None *	None *	None *	None *	None *	None *	
0x1	Reserved	SCL <i>n</i>	SDI <i>n</i>	USI <i>n</i>	TOUT <i>n</i> 0/ CAP <i>n</i> 0	Reserved	Reserved	Reserved	
0x2		SDA <i>n</i>	SDO <i>n</i>	USOUT <i>n</i>	TOUT <i>n</i> 1/ CAP <i>n</i> 1				
0x3		Reserved	SPICLK <i>n</i>	Reserved	Reserved				
0x4			#SPISS <i>n</i>						
0x5									
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.

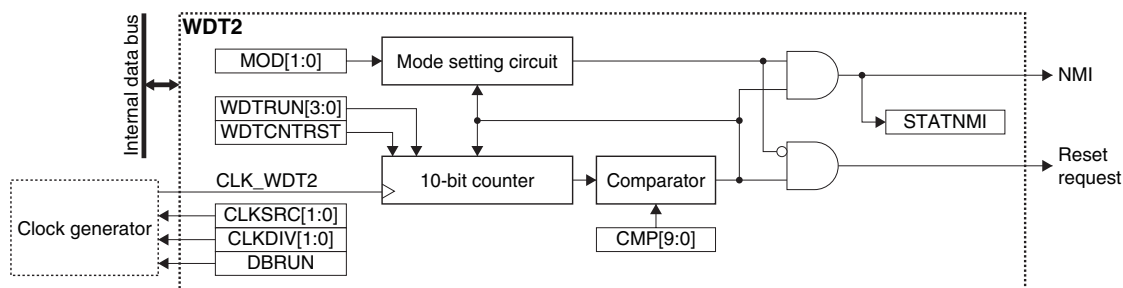


Figure 8.1.1 WDT2 Configuration

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.

8.3 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.

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

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) $t_{WDT} = 2.5$ seconds when $CLK_WDT2 = 256$ 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	1/65,536	1/128	1/65,536	1/1
0x2	1/32,768		1/32,768	
0x1	1/16,384		1/16,384	
0x0	1/8,192		1/8,192	

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

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 Real-Time Clock (RTCA)

9.1 Overview

RTCA is a real-time clock with a perpetual calendar function. The main features of RTCA are outlined below.

- Includes a BCD real-time clock counter to implement a time-of-day clock (second, minute, and hour) and calendar (day, day of the week, month, and year with leap year supported).
- Provides a hold function for reading correct counter values by suspending the real-time clock counter operation.
- 24-hour or 12-hour mode is selectable.
- Capable of controlling the starting and stopping of the time-of-day clock.
- Provides a 30-second correction function to adjust time using a time signal.
- Includes a 1 Hz counter to count 128 to 1 Hz.
- Includes a BCD stopwatch counter with 1/100-second counting supported.
- Provides a theoretical regulation function to correct clock error due to frequency tolerance with no external parts required.

Figure 9.1.1 shows the configuration of RTCA.

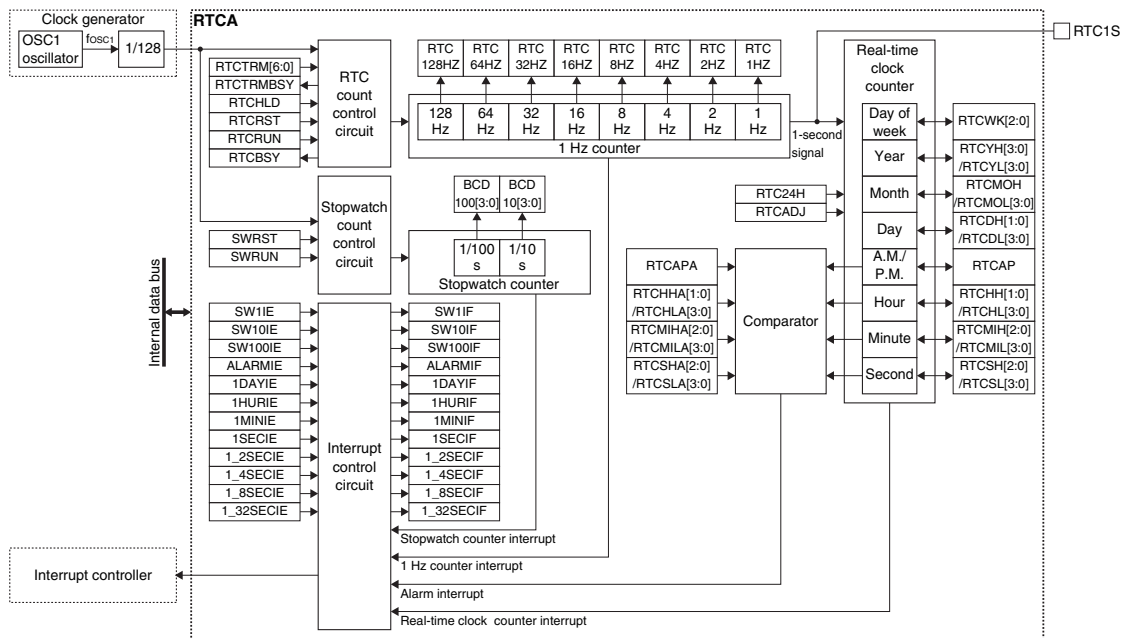


Figure 9.1.1 RTCA Configuration

9.2 Output Pin and External Connection

9.2.1 Output Pin

Table 9.2.1.1 shows the RTCA pin.

Table 9.2.1.1 RTCA Pin

Pin name	I/O*	Initial status*	Function
RTC1S	O	O (L)	1-second signal monitor output pin

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

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

9.3 Clock Settings

9.3.1 RTCA Operating Clock

RTCA uses CLK_RTCA, which is generated by the clock generator from OSC1 as the clock source, as its operating clock. RTCA is operable when OSC1 is enabled.

To continue the RTCA operation during SLEEP mode with OSC1 being activated, the CLGOSC.OSC1SLPC bit must be set to 0.

9.3.2 Theoretical Regulation Function

The time-of-day clock loses accuracy if the OSC1 frequency fosc1 has a frequency tolerance from 32.768 kHz. To correct this error without changing any external part, RTCA provides a theoretical regulation function. Follow the procedure below to perform theoretical regulation.

1. Measure fosc1 and calculate the frequency tolerance correction value

$$m \text{ [ppm]} = -\{(f_{osc1} - 32,768 \text{ [Hz]}) / 32,768 \text{ [Hz]}\} \times 10^6.$$
2. Determine the theoretical regulation execution cycle time “n seconds.”
3. Determine the value to be written to the RTCCTL.RTCTRM[6:0] bits from the results in Steps 1 and 2.
4. Write the value determined in Step 3 to the RTCCTL.RTCTRM[6:0] bits periodically in n-second cycles using an RTCA alarm or second interrupt.
5. Monitor the RTC1S signal to check that every n-second cycle has no error included.

The correction value for theoretical regulation can be specified within the range from -64 to +63 and it should be written to the RTCCTL.RTCTRM[6:0] bits as a two's-complement number. Use Eq. 9.1 to calculate the correction value.

$$RTCTRM[6:0] = \frac{m}{10^6} \times 256 \times n \quad (\text{However, RTCTRM[6:0] is an integer after rounding off to -64 to +63.}) \quad (\text{Eq. 9.1})$$

Where

- n: Theoretical regulation execution cycle time [second] (time interval to write the correct value to the RTCCTL.RTCTRM[6:0] bits periodically via software)
- m: OSC1 frequency tolerance correction value [ppm]

Figure 9.3.2.1 shows the RTC1S signal waveform.

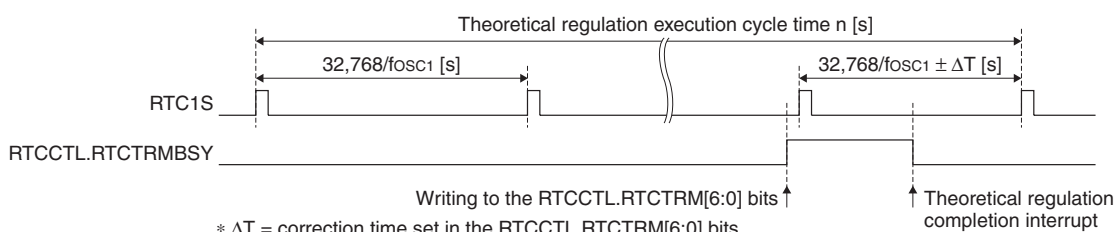


Figure 9.3.2.1 RTC1S Signal Waveform

Table 9.3.2.1 lists the frequency tolerance correction rates when the theoretical regulation execution cycle time n is 4,096 seconds as an example.

Table 9.3.2.1 Correction Rates when Theoretical Regulation Execution Cycle Time n = 4,096 Seconds

RTCCTL.RTCTRM[6:0] bits (two's-complement)	Correction value (decimal)	Correction rate [ppm]	RTCCTL.RTCTRM[6:0] bits (two's-complement)	Correction value (decimal)	Correction rate [ppm]
0x00	0	0.0	0x40	-64	-61.0
0x01	1	1.0	0x41	-63	-60.1
0x02	2	1.9	0x42	-62	-59.1
0x03	3	2.9	0x43	-61	-58.2
...
0x3e	62	59.1	0x7e	-2	-1.9
0x3f	63	60.1	0x7f	-1	-1.0

Minimum resolution: 1 ppm, Correction rate range: -61.0 to 60.1 ppm

- Notes:**
- The theoretical regulation affects only the real-time clock counter and 1 Hz counter. It does not affect the stopwatch counter.
 - After a value is written to the RTCCTL.RTC24H bit, the theoretical regulation correction takes effect on the 1 Hz counter value at the same timing as when the 1 Hz counter changes to 0x7f. Also an interrupt occurs depending on the counter value at this time.

9.4 Operations

9.4.1 RTCA Control

Follow the sequences shown below to set time to RTCA, to read the current time and to set alarm.

Time setting

1. Set RTCA to 12H or 24H mode using the RTCCTL.RTC24H bit.
2. Write 1 to the RTCCTL.RTCRUN bit to enable for the real-time clock counter to start counting up.
3. Check to see if the RTCCTL.RTCBSY bit = 0 that indicates the counter is ready to rewrite. If the RTCCTL.RTCBSY bit = 1, wait until it is set to 0.
4. Write the current date and time in BCD code to the control bits listed below.
 RTCSEC.RTCSH[2:0]/RTCSL[3:0] bits (second)
 RTCHUR.RTCMIH[2:0]/RTCMIL[3:0] bits (minute)
 RTCHUR.RTCHH[1:0]/RTCHL[3:0] bits (hour)
 RTCHUR.RTCAP bit (AM/PM) (effective when RTCCTL.RTC24H bit = 0)
 RTCMON.RTCDH[1:0]/RTCDL[3:0] bits (day)
 RTCMON.RTCMOH/RTCMOL[3:0] bits (month)
 RTCYAR.RTCYH[3:0]/RTCYL[3:0] bits (year)
 RTCYAR.RTCWK[2:0] bits (day of the week)
5. Write 1 to the RTCCTL.RTCADJ bit (execute 30-second correction) using a time signal to adjust the time. (For more information on the 30-second correction, refer to “Real-Time Clock Counter Operations.”)
6. Write 1 to the real-time clock counter interrupt flags in the RTCINTF register to clear them.
7. Write 1 to the interrupt enable bits in the RTCINTE register to enable real-time clock counter interrupts.

Time read

1. Check to see if the RTCCTL.RTCBSY bit = 0. If the RTCCTL.RTCBSY bit = 1, wait until it is set to 0.
2. Write 1 to the RTCCTL.RTCHLD bit to suspend count-up operation of the real-time clock counter.
3. Read the date and time from the control bits listed in “Time setting, Step 4” above.
4. Write 0 to the RTCCTL.RTCHLD bit to resume count-up operation of the real-time clock counter. If a second count-up timing has occurred in the count hold state, the hardware corrects the second counter for +1 second (for more information on the +1 second correction, refer to “Real-Time Clock Counter Operations”).

Alarm setting

1. Write 0 to the RTCINTE.ALARMIE bit to disable alarm interrupts.
2. Write the alarm time in BCD code to the control bits listed below (a time within 24 hours from the current time can be specified).
 RTCALM1.RTCSHA[2:0]/RTCSLA[3:0] bits (second)
 RTCALM2.RTCMIHA[2:0]/RTCMILA[3:0] bits (minute)
 RTCALM2.RTCHHA[1:0]/RTCHLA[3:0] bits (hour)
 RTCALM2.RTCAPA bit (AM/PM) (effective when RTCCTL.RTC24H bit = 0)
3. Write 1 to the RTCINTF.ALARMIF bit to clear the alarm interrupt flag.
4. Write 1 to the RTCINTE.ALARMIE bit to enable alarm interrupts.
 When the real-time clock counter reaches the alarm time set in Step 2, an alarm interrupt occurs.

9.4.2 Real-Time Clock Counter Operations

The real-time clock counter consists of second, minute, hour, AM/PM, day, month, year, and day of the week counters and it performs counting up using the RTC1S signal. It has the following functions as well.

Recognizing leap years

The leap year recognizing algorithm used in RTCA is effective only for Christian Era years. Years within 0 to 99 that can be divided by four without a remainder are recognized as leap years. If the year counter = 0x00, RTCA assumes it as a common year. If a leap year is recognized, the count range of the day counter changes when the month counter is set to February.

Corrective operation when a value out of the effective range is set

When a value out of the effective range is set to the year, day of the week, or hour (in 24H mode) counter, the counter will be cleared to 0 at the next count-up timing. When a such value is set to the month, day, or hour (in 12H mode) counter, the counter will be set to 1 at the next count-up timing.

Note: Do not set the RTCMON.RTCMOL[3:0] bits to 0x0 if the RTCMON.RTCMOH bit = 0.

30-second correction

This function is provided to set the time-of-day clock by the time signal. Writing 1 to the RTCCTL.RTCADJ bit clears the second counter and adds 1 to the minute counter if the second counter represents 30 to 59 seconds, or clears the second counter with the minute counter left unchanged if the second counter represents 0 to 29 seconds.

+1 second correction

If a second count-up timing occurred while the RTCCTL.RTCHLD bit = 1 (count hold state), the real-time clock counter counts up by +1 second (performs +1 second correction) after the counting has resumed by writing 0 to the RTCCTL.RTCHLD bit.

Note: If two or more second count-up timings occurred while the RTCCTL.RTCHLD bit = 1, the counter is always corrected for +1 second only.

9.4.3 Stopwatch Control

Follow the sequences shown below to start counting of the stopwatch and to read the counter.

Count start

1. Write 1 to the RTCSWCTL.SWRST bit to reset the stopwatch counter.
2. Write 1 to the stopwatch interrupt flags in the RTCINTF register to clear them.
3. Write 1 to the interrupt enable bits in the RTCINTE register to enable stopwatch interrupts.
4. Write 1 to the RTCSWCTL.SWRUN bit to start stopwatch count up operation.

Counter read

1. Read the count value from the RTCSWCTL.BCD10[3:0] and BCD100[3:0] bits.
2. Read again.
 - i. If the two read values are the same, assume that the count values are read correctly.
 - ii. If different values are read, perform reading once more and compare the read value with the previous one.

9.4.4 Stopwatch Count-up Pattern

The stopwatch consists of 1/100-second and 1/10-second counters and these counters perform counting up in increments of approximate 1/100 and 1/10 seconds with the count-up patterns shown in Figure 9.4.4.1.

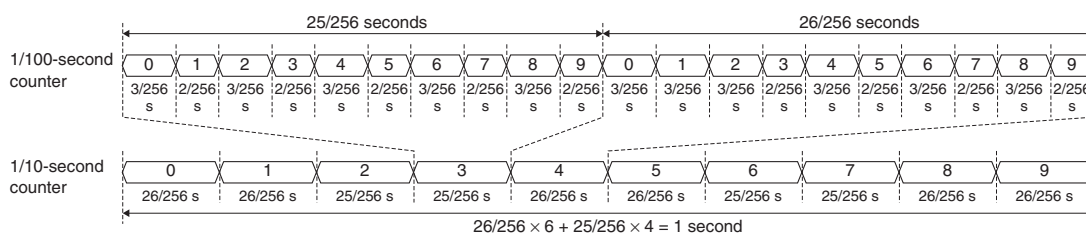


Figure 9.4.4.1 Stopwatch Count-Up Patterns

9.5 Interrupts

RTCA has a function to generate the interrupts shown in Table 9.5.1.

Table 9.5.1 RTCA Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Alarm	RTCINTF.ALARMIF	Matching between the RTCALM1–2 register contents and the real-time clock counter contents	Writing 1
1-day	RTCINTF.1DAYIF	Day counter count up	Writing 1
1-hour	RTCINTF.1HURIF	Hour counter count up	Writing 1
1-minute	RTCINTF.1MINIF	Minute counter count up	Writing 1
1-second	RTCINTF.1SECIF	Second counter count up	Writing 1
1/2-second	RTCINTF.1_2SECIF	See Figure 9.5.1.	Writing 1
1/4-second	RTCINTF.1_4SECIF	See Figure 9.5.1.	Writing 1
1/8-second	RTCINTF.1_8SECIF	See Figure 9.5.1.	Writing 1
1/32-second	RTCINTF.1_32SECIF	See Figure 9.5.1.	Writing 1
Stopwatch 1 Hz	RTCINTF.SW1IF	1/10-second counter overflow	Writing 1
Stopwatch 10 Hz	RTCINTF.SW10IF	1/10-second counter count up	Writing 1
Stopwatch 100 Hz	RTCINTF.SW100IF	1/100-second counter count up	Writing 1
Theoretical regulation completion	RTCINTF.RTCTRMIF	At the end of theoretical regulation operation	Writing 1

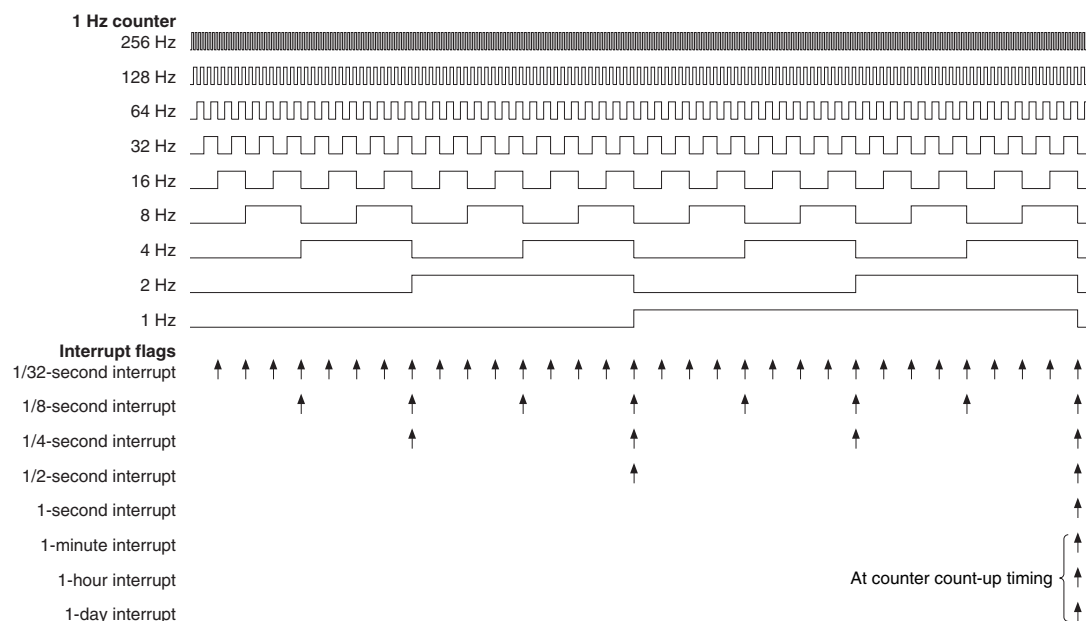


Figure 9.5.1 RTCA Interrupt Timings

- Notes:**
- 1-second to 1/32-second interrupts occur after a lapse of 1/256 second from change of the 1 Hz counter value.
 - An alarm interrupt occurs after a lapse of 1/256 second from matching between the AM/PM (in 12H mode), hour, minute, and second counter value and the alarm setting value.

RTCA 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.

9.6 Control Registers

RTC Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCCTL	15	RTCTRMBYSY	0	H0	R	–
	14–8	RTCTRM[6:0]	0x00	H0	W	Read as 0x00.
	7	–	0	–	R	–
	6	RTCBSY	0	H0	R	–
	5	RTCHLD	0	H0	R/W	Cleared by setting the RTCCTL.RTCRST bit to 1.
	4	RTC24H	0	H0	R/W	–
	3	–	0	–	R	–
	2	RTCADJ	0	H0	R/W	Cleared by setting the RTCCTL.RTCRST bit to 1.
	1	RTCST	0	H0	R/W	–
	0	RTCUN	0	H0	R/W	–

Bit 15 RTCTRMBYSY

This bit indicates whether the theoretical regulation is currently executed or not.

1 (R): Theoretical regulation is executing.

0 (R): Theoretical regulation has finished (or not executed).

This bit goes 1 when a value is written to the RTCCTL.RTCTRM[6:0] bits. The theoretical regulation takes up to 1 second for execution. This bit reverts to 0 automatically after the theoretical regulation has finished execution.

Bits 14–8 RTCTRM[6:0]

Write the correction value for adjusting the 1 Hz frequency to these bits to execute theoretical regulation. For a calculation method of correction value, refer to “Theoretical Regulation Function.”

Notes: • When the RTCCTL.RTCTRMBYSY bit = 1, the RTCCTL.RTCTRM[6:0] bits cannot be rewritten.

• Writing 0x00 to the RTCCTL.RTCTRM[6:0] bits sets the RTCCTL.RTCTRMBYSY bit to 1 as well. However, no correcting operation is performed.

Bit 7 Reserved

Bit 6 RTCBSY

This bit indicates whether the counter is performing count-up operation or not.

1 (R): In count-up operation

0 (R): Idle (ready to rewrite real-time clock counter)

This bit goes 1 when performing 1-second count-up, +1 second correction, or 30-second correction. It retains 1 for 1/256 second and then reverts to 0.

Bit 5 RTCHLD

This bit halts the count-up operation of the real-time clock counter.

1 (R/W): Halt real-time clock counter count-up operation

0 (R/W): Normal operation

Writing 1 to this bit halts the count-up operation of the real-time clock counter, this makes it possible to read the counter value correctly without changing the counter. Write 0 to this bit to resume count-up operation immediately after the counter has been read. Depending on these operation timings, the +1 second correction may be executed after the count-up operation resumes. For more information on the +1 second correction, refer to “Real-Time Clock Counter Operations.”

Note: When the RTCCTL.RTCTRMBSY bit = 1, the RTCCTL.RTCHLD bit cannot be rewritten to 1 (as fixed at 0).

Bit 4 RTC24H

This bit sets the hour counter to 24H mode or 12H mode.

1 (R/W): 24H mode

0 (R/W): 12H mode

This selection changes the count range of the hour counter. Note, however, that the counter value is not updated automatically, therefore, it must be programmed again.

Note: Be sure to avoid writing to this bit when the RTCCTL.RTCRUN bit = 1.

Bit 3 Reserved

Bit 2 RTCADJ

This bit executes the 30-second correction time adjustment function.

1 (W): Execute 30-second correction

0 (W): Ineffective

1 (R): 30-second correction is executing.

0 (R): 30-second correction has finished. (Normal operation)

Writing 1 to this bit executes 30-second correction and an enabled interrupt occurs even if the RTCCTL.RTCRUN bit = 0. The correction takes up to 2/256 seconds. The RTCCTL.RTCADJ bit is automatically cleared to 0 when the correction has finished. For more information on the 30-second correction, refer to “Real-Time Clock Counter Operations.”

Notes: • Be sure to avoid writing to this bit when the RTCCTL.RTCBSY bit = 1.

• Do not write 1 to this bit again while the RTCCTL.RTCADJ bit = 1.

Bit 1 RTCRST

This bit resets the 1 Hz counter, the RTCCTL.RTCADJ bit, and the RTCCTL.RTCHLD bit.

1 (W): Reset

0 (W): Ineffective

1 (R): Reset is being executed.

0 (R): Reset has finished. (Normal operation)

This bit is automatically cleared to 0 after reset has finished.

Bit 0 RTCRUN

This bit starts/stops the real-time clock counter.

1 (R/W): Running/start control

0 (R/W): Idle/stop control

When the real-time clock counter stops counting by writing 0 to this bit, the counter retains the value when it stopped. Writing 1 to this bit again resumes counting from the value retained.

RTC Second Alarm Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCALM1	15	–	0	–	R	–
	14–12	RTCSHA[2:0]	0x0	H0	R/W	
	11–8	RTCSLA[3:0]	0x0	H0	R/W	
	7–0	–	0x00	–	R	

Bit 15 Reserved

Bits 14–12 RTCSHA[2:0]

Bits 11–8 RTCSLA[3:0]

The RTCALM1.RTCSHA[2:0] bits and the RTCALM1.RTCSLA[3:0] bits set the 10-second digit and 1-second digit of the alarm time, respectively. A value within 0 to 59 seconds can be set in BCD code as shown in Table 9.6.1.

9 REAL-TIME CLOCK (RTCA)

Table 9.6.1 Setting Examples in BCD Code

Setting value in BCD code		Alarm (second) setting
RTCALM1.RTCSHA[2:0] bits	RTCALM1.RTCSLA[3:0] bits	
0x0	0x0	00 seconds
0x0	0x1	01 second
...
0x0	0x9	09 seconds
0x1	0x0	10 seconds
...
0x5	0x9	59 seconds

Bits 7–0 **Reserved**

RTC Hour/Minute Alarm Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCALM2	15	–	0	–	R	–
	14	RTCAPA	0	H0	R/W	
	13–12	RTCHHA[1:0]	0x0	H0	R/W	
	11–8	RTCHLA[3:0]	0x0	H0	R/W	
	7	–	0	–	R	
	6–4	RTCMIHA[2:0]	0x0	H0	R/W	
	3–0	RTCMILA[3:0]	0x0	H0	R/W	

Bit 15 **Reserved**

Bit 14 **RTCAPA**

This bit sets A.M. or P.M. of the alarm time in 12H mode (RTCCTL.RTC24H bit = 0).

1 (R/W): P.M.

0 (R/W): A.M.

This setting is ineffective in 24H mode (RTCCTL.RTC24H bit = 1).

Bits 13–12 **RTCHHA[1:0]**

Bits 11–8 **RTCHLA[3:0]**

The RTCALM2.RTCHHA[1:0] bits and the RTCALM2.RTCHLA[3:0] bits set the 10-hour digit and 1-hour digit of the alarm time, respectively. A value within 1 to 12 o'clock in 12H mode or 0 to 23 in 24H mode can be set in BCD code.

Bit 7 **Reserved**

Bits 6–4 **RTCMIHA[2:0]**

Bits 3–0 **RTCMILA[3:0]**

The RTCALM2.RTCMIHA[2:0] bits and the RTCALM2.RTCMILA[3:0] bits set the 10-minute digit and 1-minute digit of the alarm time, respectively. A value within 0 to 59 minutes can be set in BCD code.

RTC Stopwatch Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCSWCTL	15–12	BCD10[3:0]	0x0	H0	R	–
	11–8	BCD100[3:0]	0x0	H0	R	
	7–5	–	0x0	–	R	
	4	SWRST	0	H0	W	Read as 0.
	3–1	–	0x0	–	R	–
	0	SWRUN	0	H0	R/W	

Bits 15–12 **BCD10[3:0]**

Bits 11–8 **BCD100[3:0]**

The 1/10-second and 1/100-second digits of the stopwatch counter can be read as a BCD code from the RTCSWCTL.BCD10[3:0] bits and the RTCSWCTL.BCD100[3:0] bits, respectively.

Note: The counter value may not be read correctly while the stopwatch counter is running. The RTCSWCTL.BCD10[3:0]/BCD100[3:0] bits must be read twice and assume the counter value was read successfully if the two read results are the same.

Bits 7–5 Reserved

Bit 4 SWRST

This bit resets the stopwatch counter to 0x00.

1 (W): Reset

0 (W): Ineffective

0 (R): Always 0 when being read

When the stopwatch counter in running status is reset, it continues counting from count 0x00. The stopwatch counter retains 0x00 if it is reset in idle status.

Bits 3–1 Reserved

Bit 0 SWRUN

This bit starts/stops the stopwatch counter.

1 (R/W): Running/start control

0 (R/W): Idle/stop control

When the stopwatch counter stops counting by writing 0 to this bit, the counter retains the value when it stopped. Writing 1 to this bit again resumes counting from the value retained.

Note: The stopwatch counter stops in sync with the stopwatch clock after 0 is written to the RTCSWCTL.SWRUN bit. Therefore, the counter value may be incremented (+1) from the value at writing 0.

RTC Second/1Hz Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCSEC	15	–	0	–	R	Cleared by setting the RTCCTL.RTCRST bit to 1.
	14–12	RTCSH[2:0]	0x0	H0	R/W	
	11–8	RTCSL[3:0]	0x0	H0	R/W	
	7	RTC1HZ	0	H0	R	
	6	RTC2HZ	0	H0	R	
	5	RTC4HZ	0	H0	R	
	4	RTC8HZ	0	H0	R	
	3	RTC16HZ	0	H0	R	
	2	RTC32HZ	0	H0	R	
	1	RTC64HZ	0	H0	R	
	0	RTC128HZ	0	H0	R	

Bit 15 Reserved

Bits 14–12 RTCSH[2:0]

Bits 11–8 RTCSL[3:0]

The RTCSEC.RTCSH[2:0] bits and the RTCSEC.RTCSL[3:0] bits are used to set and read the 10-second digit and the 1-second digit of the second counter, respectively. The setting/read values are a BCD code within the range from 0 to 59.

Note: Be sure to avoid writing to the RTCSEC.RTCSH[2:0]/RTCSL[3:0] bits while the RTCCTL.RTCBSY bit = 1.

Bit 7	RTC1HZ
Bit 6	RTC2HZ
Bit 5	RTC4HZ
Bit 4	RTC8HZ
Bit 3	RTC16HZ
Bit 2	RTC32HZ
Bit 1	RTC64HZ
Bit 0	RTC128HZ

1 Hz counter data can be read from these bits.

The following shows the correspondence between the bit and frequency:

RTCSEC.RTC1HZ bit:	1 Hz
RTCSEC.RTC2HZ bit:	2 Hz
RTCSEC.RTC4HZ bit:	4 Hz
RTCSEC.RTC8HZ bit:	8 Hz
RTCSEC.RTC16HZ bit:	16 Hz
RTCSEC.RTC32HZ bit:	32 Hz
RTCSEC.RTC64HZ bit:	64 Hz
RTCSEC.RTC128HZ bit:	128 Hz

Note: The counter value may not be read correctly while the 1 Hz counter is running. These bits must be read twice and assume the counter value was read successfully if the two read results are the same.

RTC Hour/Minute Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCHUR	15	–	0	–	R	–
	14	RTCAP	0	H0	R/W	
	13–12	RTCHH[1:0]	0x1	H0	R/W	
	11–8	RTCHL[3:0]	0x2	H0	R/W	
	7	–	0	–	R	
	6–4	RTCMIH[2:0]	0x0	H0	R/W	
	3–0	RTCMIL[3:0]	0x0	H0	R/W	

Bit 15 **Reserved**

Bit 14 **RTCAP**

This bit is used to set and read A.M. or P.M. data in 12H mode (RTCCTL.RTC24H bit = 0).

1 (R/W): P.M.

0 (R/W): A.M.

In 24H mode (RTCCTL.RTC24H bit = 1), this bit is fixed at 0 and writing 1 is ignored. However, if the RTCHUR.RTCAP bit = 1 when changed to 24H mode, it goes 0 at the next count-up timing of the hour counter.

Bits 13–12 **RTCHH[1:0]**

Bits 11–8 **RTCHL[3:0]**

The RTCHUR.RTCHH[1:0] bits and the RTCHUR.RTCHL[3:0] bits are used to set and read the 10-hour digit and the 1-hour digit of the hour counter, respectively. The setting/read values are a BCD code within the range from 1 to 12 in 12H mode or 0 to 23 in 24H mode.

Note: Be sure to avoid writing to the RTCHUR.RTCHH[1:0]/RTCHL[3:0] bits while the RTCCTL.RTCBSY bit = 1.

Bit 7 **Reserved**

Bits 6–4 **RTCMIH[2:0]****Bits 3–0** **RTCMIL[3:0]**

The RTCHUR.RTCMIH[2:0] bits and the RTCHUR.RTCMIL[3:0] bits are used to set and read the 10-minute digit and the 1-minute digit of the minute counter, respectively. The setting/read values are a BCD code within the range from 0 to 59.

Note: Be sure to avoid writing to the RTCHUR.RTCMIH[2:0]/RTCMIL[3:0] bits while the RTCCTL.RTCBSY bit = 1.

RTC Month/Day Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCMON	15–13	–	0x0	–	R	–
	12	RTCMOH	0	H0	R/W	
	11–8	RTCMOL[3:0]	0x1	H0	R/W	
	7–6	–	0x0	–	R	
	5–4	RTCDH[1:0]	0x0	H0	R/W	
	3–0	RTCDL[3:0]	0x1	H0	R/W	

Bits 15–13 Reserved**Bit 12** **RTCMOH****Bits 11–8** **RTCMOL[3:0]**

The RTCMON.RTCMOH bit and the RTCMON.RTCMOL[3:0] bits are used to set and read the 10-month digit and the 1-month digit of the month counter, respectively. The setting/read values are a BCD code within the range from 1 to 12.

Notes: • Be sure to avoid writing to the RTCMON.RTCMOH/RTCMOL[3:0] bits while the RTCCTL.RTCBSY bit = 1.

• Be sure to avoid setting the RTCMON.RTCMOH/RTCMOL[3:0] bits to 0x00.

Bits 7–6 **Reserved****Bits 5–4** **RTCDH[1:0]****Bits 3–0** **RTCDL[3:0]**

The RTCMON.RTCDH[1:0] bits and the RTCMON.RTCDL[3:0] bits are used to set and read the 10-day digit and the 1-day digit of the day counter, respectively. The setting/read values are a BCD code within the range from 1 to 31 (to 28 for February in a common year, to 29 for February in a leap year, or to 30 for April/June/September/November).

Note: Be sure to avoid writing to the RTCMON.RTCDH[1:0]/RTCDL[3:0] bits while the RTCCTL.RTCBSY bit = 1.

RTC Year/Week Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCYAR	15–11	–	0x00	–	R	–
	10–8	RTCWK[2:0]	0x0	H0	R/W	
	7–4	RTCYH[3:0]	0x0	H0	R/W	
	3–0	RTCYL[3:0]	0x0	H0	R/W	

Bits 15–11 Reserved**Bits 10–8** **RTCWK[2:0]**

These bits are used to set and read day of the week.

The day of the week counter is a base-7 counter and the setting/read values are 0x0 to 0x6. Table 9.6.2 lists the correspondence between the count value and day of the week.

Table 9.6.2 Correspondence between the count value and day of the week

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

Note: Be sure to avoid writing to the RTCYAR.RTCWK[2:0] bits while the RTCCTL.RTCBSY bit = 1.

Bits 7–4 **RTCYH[3:0]**

Bits 3–0 **RTCYL[3:0]**

The RTCYAR.RTCYH[3:0] bits and the RTCYAR.RTCYL[3:0] bits are used to set and read the 10-year digit and the 1-year digit of the year counter, respectively. The setting/read values are a BCD code within the range from 0 to 99.

Note: Be sure to avoid writing to the RTCYAR.RTCYH[3:0]/RTCYL[3:0] bits while the RTCCTL.RTCBSY bit = 1.

RTC Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCINTF	15	RTCTRMIF	0	H0	R/W	Cleared by writing 1.
	14	SW1IF	0	H0	R/W	
	13	SW10IF	0	H0	R/W	
	12	SW100IF	0	H0	R/W	
	11–9	–	0x0	–	R	–
	8	ALARMIF	0	H0	R/W	Cleared by writing 1.
	7	1DAYIF	0	H0	R/W	
	6	1HURIF	0	H0	R/W	
	5	1MINIF	0	H0	R/W	
	4	1SECF	0	H0	R/W	
	3	1_2SECF	0	H0	R/W	
	2	1_4SECF	0	H0	R/W	
	1	1_8SECF	0	H0	R/W	
	0	1_32SECF	0	H0	R/W	

Bit 15 **RTCTRMIF**

Bit 14 **SW1IF**

Bit 13 **SW10IF**

Bit 12 **SW100IF**

These bits indicate the real-time clock 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:

RTCINTF.RTCTRMIF bit: Theoretical regulation completion interrupt

RTCINTF.SW1IF bit: Stopwatch 1 Hz interrupt

RTCINTF.SW10IF bit: Stopwatch 10 Hz interrupt

RTCINTF.SW100IF bit: Stopwatch 100 Hz interrupt

Bits 11–9 **Reserved**

Bit 8	ALARMIF
Bit 7	1DAYIF
Bit 6	1HURIF
Bit 5	1MINIF
Bit 4	1SECIF
Bit 3	1_2SECIF
Bit 2	1_4SECIF
Bit 1	1_8SECIF
Bit 0	1_32SECIF

These bits indicate the real-time clock 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:

- RTCINTF.ALARMIF bit: Alarm interrupt
 RTCINTF.1DAYIF bit: 1-day interrupt
 RTCINTF.1HURIF bit: 1-hour interrupt
 RTCINTF.1MINIF bit: 1-minute interrupt
 RTCINTF.1SECIF bit: 1-second interrupt
 RTCINTF.1_2SECIF bit: 1/2-second interrupt
 RTCINTF.1_4SECIF bit: 1/4-second interrupt
 RTCINTF.1_8SECIF bit: 1/8-second interrupt
 RTCINTF.1_32SECIF bit: 1/32-second interrupt

RTC Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCINTE	15	RTCTRMIE	0	H0	R/W	—
	14	SW1IE	0	H0	R/W	
	13	SW10IE	0	H0	R/W	
	12	SW100IE	0	H0	R/W	
	11–9	—	0x0	—	R	
	8	ALARMIE	0	H0	R/W	
	7	1DAYIE	0	H0	R/W	
	6	1HURIE	0	H0	R/W	
	5	1MINIE	0	H0	R/W	
	4	1SECIE	0	H0	R/W	
	3	1_2SECIE	0	H0	R/W	
	2	1_4SECIE	0	H0	R/W	
	1	1_8SECIE	0	H0	R/W	
	0	1_32SECIE	0	H0	R/W	

Bit 15	RTCTRMIE
Bit 14	SW1IE
Bit 13	SW10IE
Bit 12	SW100IE

These bits enable real-time clock interrupts.

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

The following shows the correspondence between the bit and interrupt:

- RTCINTE.RTCTRMIE bit: Theoretical regulation completion interrupt
 RTCINTE.SW1IE bit: Stopwatch 1 Hz interrupt
 RTCINTE.SW10IE bit: Stopwatch 10 Hz interrupt
 RTCINTE.SW100IE bit: Stopwatch 100 Hz interrupt

9 REAL-TIME CLOCK (RTCA)

Bits 11–9 Reserved

Bit 8	ALARMIE
Bit 7	1DAYIE
Bit 6	1HURIE
Bit 5	1MINIE
Bit 4	1SECIE
Bit 3	1_2SECIE
Bit 2	1_4SECIE
Bit 1	1_8SECIE
Bit 0	1_32SECIE

These bits enable real-time clock interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

RTCINTE.ALARMIE bit: Alarm interrupt

RTCINTE.1DAYIE bit: 1-day interrupt

RTCINTE.1HURIE bit: 1-hour interrupt

RTCINTE.1MINIE bit: 1-minute interrupt

RTCINTE.1SECIE bit: 1-second interrupt

RTCINTE.1_2SECIE bit: 1/2-second interrupt

RTCINTE.1_4SECIE bit: 1/4-second interrupt

RTCINTE.1_8SECIE bit: 1/8-second interrupt

RTCINTE.1_32SECIE bit: 1/32-second interrupt

10 Supply Voltage Detector (SVD3)

10.1 Overview

SVD3 is a supply voltage detector to monitor the power supply voltage on the VDD pin or the voltage applied to an external pin. The main features are listed below.

- Power supply voltage to be detected: Selectable from VDD and external power sources (EXSVD0 and EXSVD1) (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 10.1.1 shows the configuration of SVD3.

Table 10.1.1 SVD3 Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
Power supply voltage to be detected	VDD and one externally input voltage (EXSVD0)			
Detectable voltage level	VDD: 28 levels (1.8 to 5.0 V)/external voltage: 32 levels (1.2 to 5.0 V)			

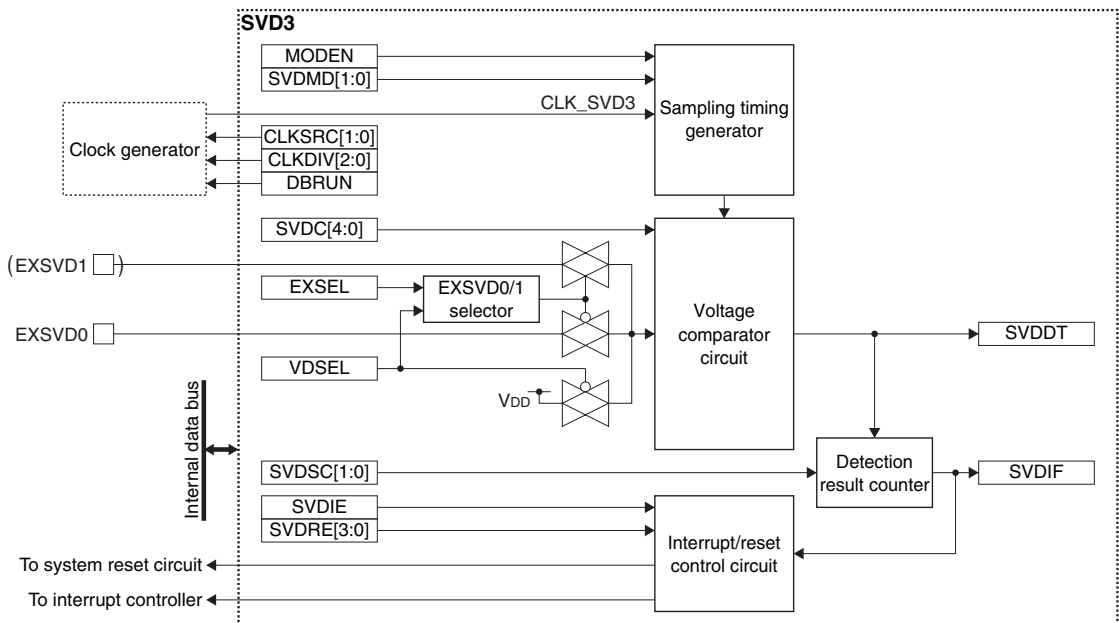


Figure 10.1.1 SVD3 Configuration

10.2 Input Pins and External Connection

10.2.1 Input Pins

Table 10.2.1.1 shows the SVD3 input pins.

Table 10.2.1.1 SVD3 Input Pins

Pin name	I/O*	Initial status*	Function
EXSVD0	A	A (Hi-Z)	External power supply voltage detection pin 0
EXSVD1	A	A (Hi-Z)	External power supply voltage detection pin 1

* Indicates the status when the pin is configured for SVD3.

If the port is shared with the EXSVD0/1 pin and other functions, the EXSVD0/1 function must be assigned to the port before SVD3 can be activated. For more information, refer to the “I/O Ports” chapter.

10.2.2 External Connection

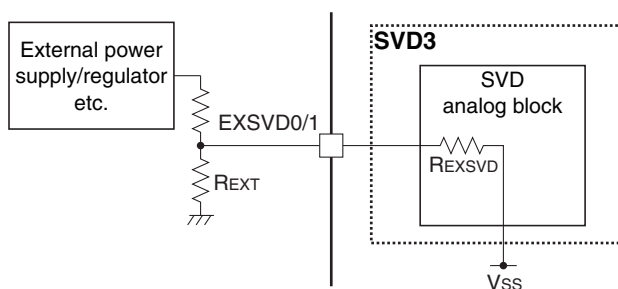


Figure 10.2.2.1 Connection between EXSVD0/1 Pin and External Power Supply

R_{EXT} resistance value must be determined so that it will be sufficiently smaller than the EXSVD input impedance R_{EXSVD} . For the EXSVD0/1 pin input voltage range and the EXSVD input impedance, refer to “Supply Voltage Detector Characteristics” in the “Electrical Characteristics” chapter.

10.3 Clock Settings

10.3.1 SVD3 Operating Clock

When using SVD3, the SVD3 operating clock CLK_SVD3 must be supplied to SVD3 from the clock generator. The CLK_SVD3 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_SVD3 frequency should be set to around 32 kHz.

10.3.2 Clock Supply in SLEEP Mode

When using SVD3 during SLEEP mode, the SVD3 operating clock CLK_SVD3 must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_SVD3 clock source.

If the CLGOSC.xxxxSLPC bit for the CLK_SVD3 clock source is 1, the CLK_SVD3 clock source is deactivated during SLEEP mode and SVD3 stops with the register settings maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_SVD3 is supplied and the SVD3 operation resumes.

10.3.3 Clock Supply in DEBUG Mode

The CLK_SVD3 supply during DEBUG mode should be controlled using the SVDCLK.DBRUN bit.

The CLK_SVD3 supply to SVD3 is suspended when the CPU enters DEBUG mode if the SVDCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_SVD3 supply resumes. Although SVD3 stops operating when the CLK_SVD3 supply is suspended, the registers retain the status before DEBUG mode was entered.

If the SVDCLK.DBRUN bit = 1, the CLK_SVD3 supply is not suspended and SVD3 will keep operating in DEBUG mode.

10.4 Operations

10.4.1 SVD3 Control

Starting detection

SVD3 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} , EXSVD0, or EXSVD1))
 - 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 SVD3 interrupt)
5. Set the SVDCTL.MODEN bit to 1. (Enable SVD3 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 SVD3 operation.

1. Write 0x0096 to the MSCPROT.PROT[15:0] bits. (Remove system protection)
2. Write 0 to the SVDCTL.MODEN bit. (Disable SVD3 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 EXSVD0/1) \geq SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}
- When SVDINTF.SVDDT bit = 1
Power supply voltage (V_{DD} or EXSVD0/1) $<$ 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).

10.4.2 SVD3 Operations

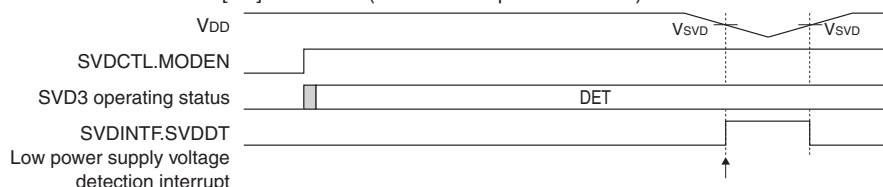
Continuous operation mode

SVD3 operates in continuous operation mode by default (SVDCTL.SVDMMD[1:0] bits = 0x0). In this mode, SVD3 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

SVD3 operates in intermittent operation mode when the SVDCTL.SVDMMD[1:0] bits are set to 0x1 to 0x3. In this mode, SVD3 turns on at an interval set using the SVDCTL.SVDMMD[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 SVD3 has successively detected low power supply voltage the number of times specified by the SVDCTL.SVDSC[1:0] bits.

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



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

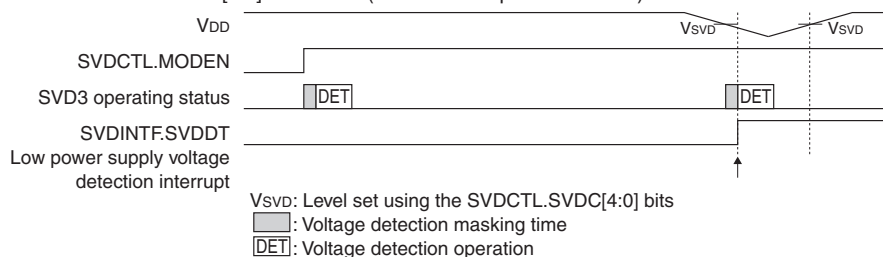


Figure 10.4.2.1 SVD3 Operations

10.5 SVD3 Interrupt and Reset

10.5.1 SVD3 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 10.5.1.1 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

SVD3 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.

10.5.2 SVD3 Reset

Setting the SVDCTL.SVDRE[3:0] bits to 0xa allows use of the SVD3 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, SVD3 enters continuous operation mode even if it was operating in intermittent operation mode, and continues operating. Issuing an SVD3 reset initializes the port assignment. However, when EXSVD0/1 is being detected, the input of the port for the EXSVD0/1 pin is sent to SVD3 so that SVD3 will continue the EXSVD0/1 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, SVD3 resumes operating in the operation mode set previously via the initialization routine.

During reset state, the SVD3 control bits are set as shown in Table 10.5.2.1.

Table 10.5.2.1 SVD3 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 SVD3 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.

10.6 Control Registers

SVD3 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 SVD3 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 SVD3 operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of SVD3.

Table 10.6.1 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	1/512		1/512	
0x4	1/256		1/256	
0x3	1/128		1/128	
0x2	1/64		1/64	
0x1	1/32		1/32	
0x0	1/16		1/16	

(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.

SVD3 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVDCTL	15	VDSEL	0	H1	R/WP	–
	14–13	SVDSC[1:0]	0x0	H0	R/WP	Writing takes effect when the SVDCTL.SVDMMD[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	EXSEL	0	H1	R/WP	–
	2–1	SVDMMD[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 SVD3.

1 (R/WP): Voltage applied to the EXSVD0/1 pin

0 (R/WP): V_{DD}

Bits 14–13 SVDSC[1:0]

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

Table 10.6.2 Interrupt/Reset Generating Condition in Intermittent Operation Mode

SVDCTL.SVDSC[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.SVDMMD[1:0] bits = 0x0).

Bits 12–8 SVDC[4:0]

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

Table 10.6.3 Setting of SVD Detection Voltage V_{SVD} /EXSVD Detection Voltage V_{SVD_EXT}

SVDCTL.SVDC[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 SVD3 reset issuance function, refer to “SVD3 Reset.”

Bit 3 EXSEL

This bit selects the external voltage to be detected when the SVDCTL.VDSEL bit = 1.

1 (R/WP): EXSVD1

0 (R/WP): EXSVD0

Note: The EXSVD1 pin does not exist depending on the model (see “Power supply voltage to be detected” in Table 10.1.1). In this case, the external voltage detection function does not work if the SVDCTL.EXSEL bit is set to 1. When using the external voltage detection function (SVDCTL.VDSEL bit = 1), the SVDCTL.EXSEL bit should be set to 0.

Bits 2–1 SVDMD[1:0]

These bits select intermittent operation mode and its detection cycle.

Table 10.6.4 Intermittent Operation Mode Detection Cycle Selection

SVDCTL.SVDMD[1:0] bits	Operation mode (detection cycle)
0x3	Intermittent operation mode (CLK_SVD3/512)
0x2	Intermittent operation mode (CLK_SVD3/256)
0x1	Intermittent operation mode (CLK_SVD3/128)
0x0	Continuous operation mode

For more information on intermittent and continuous operation modes, refer to “SVD3 Operations.”

Bit 0 MODEN

This bit enables/disables for the SVD3 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 SVD3 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 SVD3 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 SVD3 is in operation after 1 is written to the SVDCTL.MODEN bit.

SVD3 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 EXSVD0/1) < SVD detection voltage V_{SVD}
or EXSVD detection voltage V_{SVD_EXT}

0 (R): Power supply voltage (V_{DD} or EXSVD0/1) \geq SVD detection voltage V_{SVD}
or EXSVD detection voltage V_{SVD_EXT}

10 SUPPLY VOLTAGE DETECTOR (SVD3)

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

Note: The SVD3 internal circuit is initialized if the interrupt flag is cleared while SVD3 is in operation after 1 is written to the SVDCTL.MODEN bit.

SVD3 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.

11 16-bit Timers (T16)

11.1 Overview

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

- 16-bit presettable 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 11.1.1 shows the configuration of a T16 channel.

Table 11.1.1 T16 Channel Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
Number of channels	4 channels (Ch.0–Ch.3)			
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 Ch.2 → Synchronous serial interface Ch.1 master clock Ch.3 → 12-bit A/D converter trigger signal			

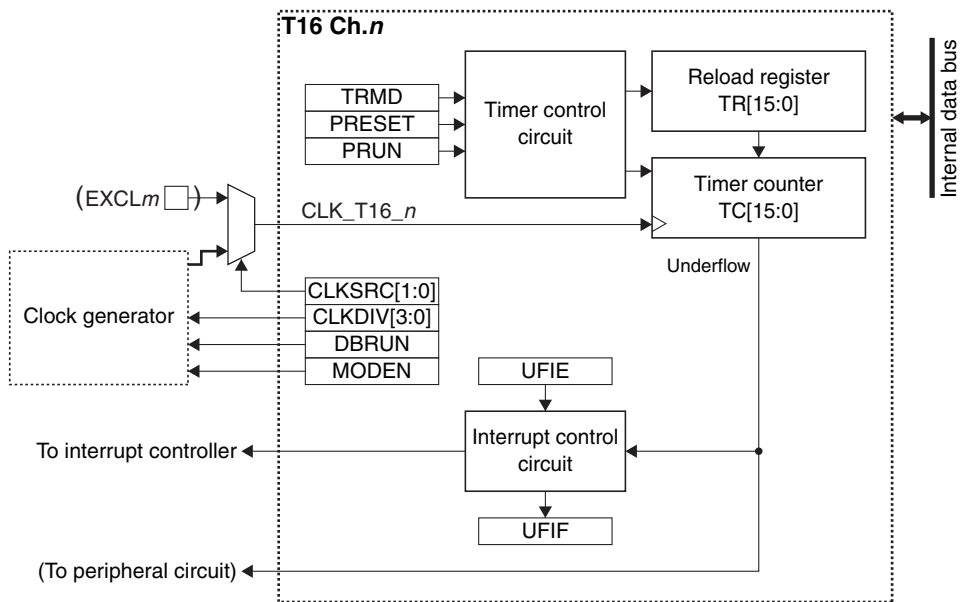


Figure 11.1.1 Configuration of a T16 Channel

11.2 Input Pin

Table 11.2.1 shows the T16 input pin.

Table 11.2.1 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.

11.3 Clock Settings

11.3.1 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)

11.3.2 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.

11.3.3 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.

11.3.4 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.

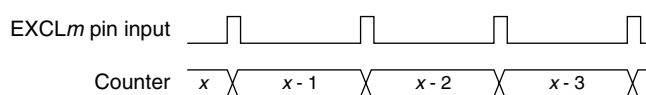


Figure 11.3.4.1 Count Down Timing

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

11.4 Operations

11.4.1 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_nCTL register bits:
 - Set the T16_nCTL.PRESET bit to 1. (Preset reload data to counter)
 - Set the T16_nCTL.PRUN bit to 1. (Start counting)

11.4.2 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_nTR register.

The following shows the equations to calculate the underflow cycle and frequency:

$$T = \frac{TR + 1}{f_{CLK_T16_n}} \quad f_T = \frac{f_{CLK_T16_n}}{TR + 1} \quad (\text{Eq. 11.1})$$

Where

T: Underflow cycle [s]
 f_T: Underflow frequency [Hz]
 TR: T16_nTR register setting
 f_{CLK_T16_n}: T16 Ch.n operating clock frequency [Hz]

11.4.3 Operations in Repeat Mode

T16 Ch.n enters repeat mode by setting the T16_nMOD.TRMD bit to 0.

In repeat mode, the count operation starts by writing 1 to the T16_nCTL.PRUN bit and continues until 0 is written. A counter underflow presets the T16_nTR 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.

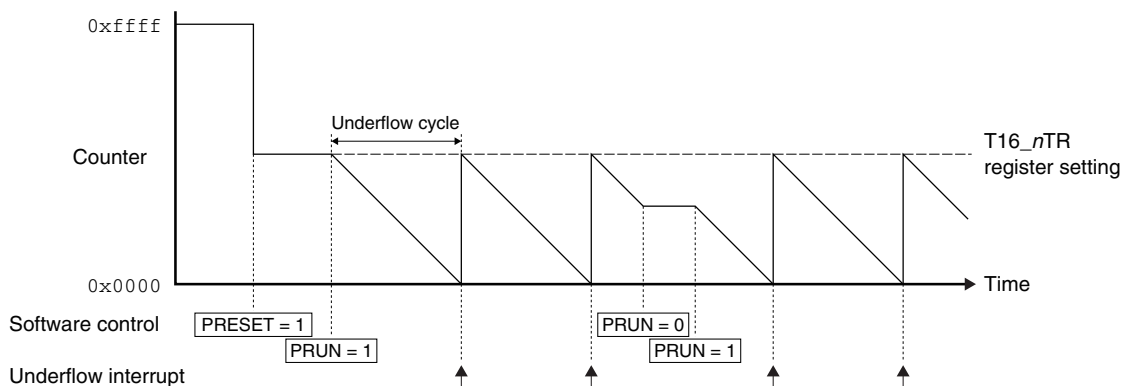


Figure 11.4.3.1 Count Operations in Repeat Mode

11.4.4 Operations in One-shot Mode

T16 Ch.n enters one-shot mode by setting the T16_nMOD.TRMD bit to 1.

In one-shot mode, the count operation starts by writing 1 to the T16_nCTL.PRUN bit and stops after the T16_nTR register value is preset to the counter when an underflow has occurred. At the same time the counter stops, the T16_nCTL.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.

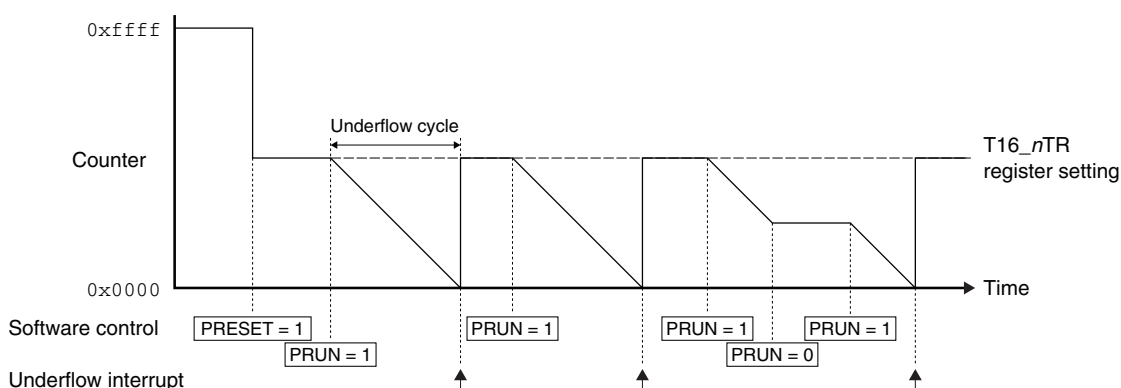


Figure 11.4.4.1 Count Operations in One-shot Mode

11.4.5 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.

11.5 Interrupt

Each T16 channel has a function to generate the interrupt shown in Table 11.5.1.

Table 11.5.1 T16 Interrupt Function

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

T16 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.

11.6 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 11.6.1 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/EXCLm
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, EXCLm 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

11 16-BIT TIMERS (T16)

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.

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_ <i>n</i> INTE	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.

12 UART (UART3)

12.1 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 12.1.1 shows the UART3 configuration.

Table 12.1.1 UART3 Channel Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
Number of channels	2 channels (Ch.0 and Ch.1)			

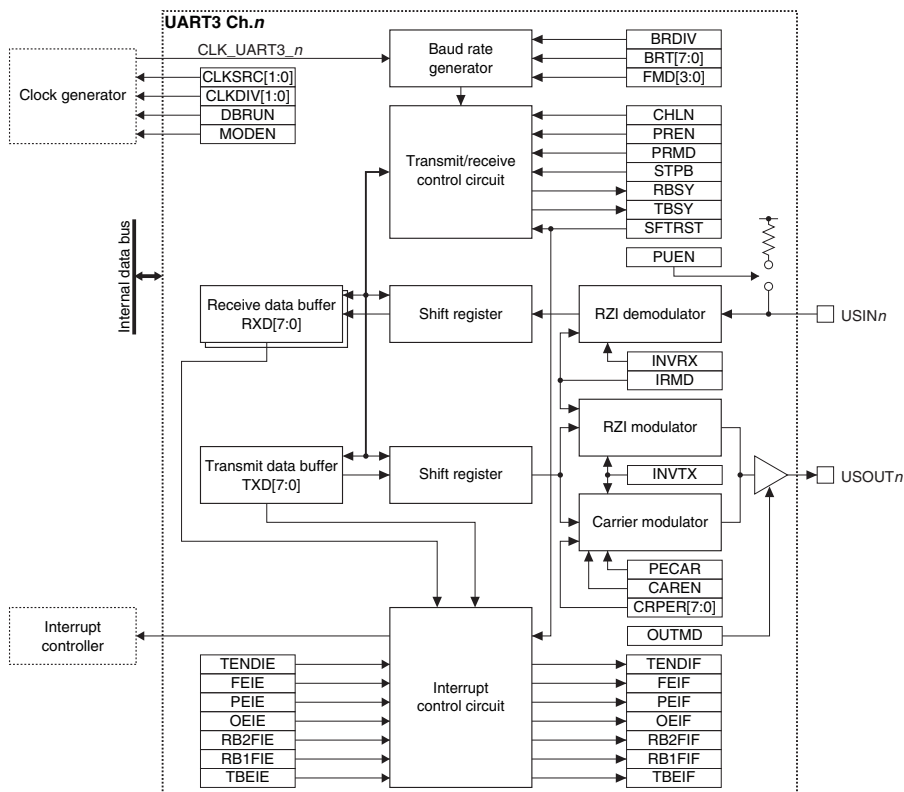


Figure 12.1.1 UART3 Configuration

12.2 Input/Output Pins and External Connections

12.2.1 List of Input/Output Pins

Table 12.2.1.1 lists the UART3 pins.

Table 12.2.1.1 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.

12.2.2 External Connections

Figure 12.2.2.1 shows a connection diagram between the UART3 in this IC and an external UART device.

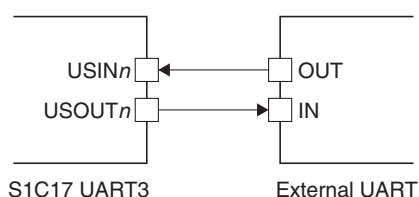


Figure 12.2.2.1 Connections between UART3 and an External UART Device

12.2.3 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.

12.2.4 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.

12.2.5 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).

12.3 Clock Settings

12.3.1 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.

12.3.2 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.

12.3.3 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.

12.3.4 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.

$$\text{bps} = \frac{\text{CLK_UART3}}{\frac{\text{BRT} + 1}{\text{BRDIV}} + \text{FMD}} \quad \text{BRT} = \text{BRDIV} \times \left(\frac{\text{CLK_UART3}}{\text{bps}} - \text{FMD} \right) - 1 \quad (\text{Eq. 12.1})$$

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.

12.4 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 12.4.1 Parity Function Setting

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

12 UART (UART3)

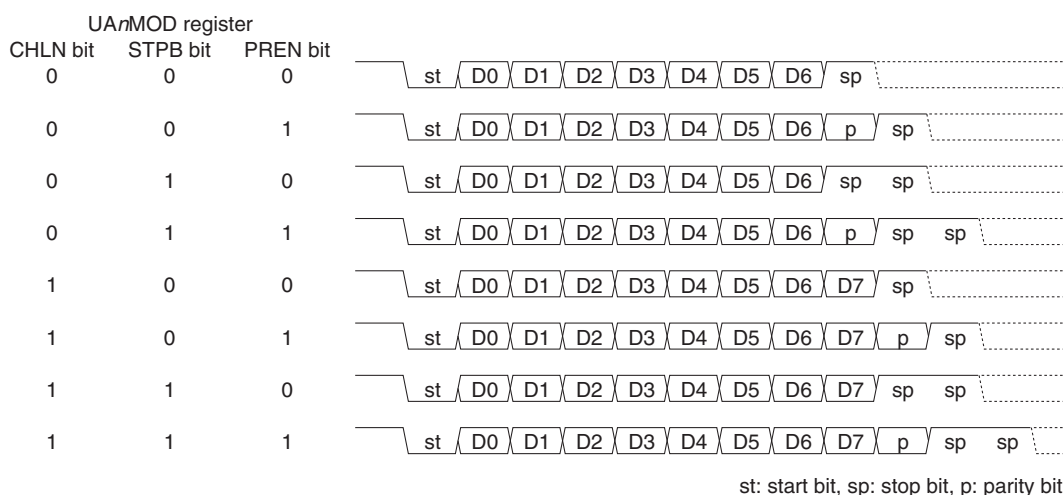


Figure 12.4.1 Data Format

12.5 Operations

12.5.1 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.

12.5.2 Data Transmission

A data sending procedure and the UART3 Ch.*n* operations are shown below. Figures 12.5.2.1 and 12.5.2.2 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 USOUT_{*n*} 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 USOUT_{*n*} 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 USOUT_{*n*} pin, the UAnINTF.TBSY bit is cleared to 0 and the UAnINTF.TENDIF bit is set to 1 (transmission completed).

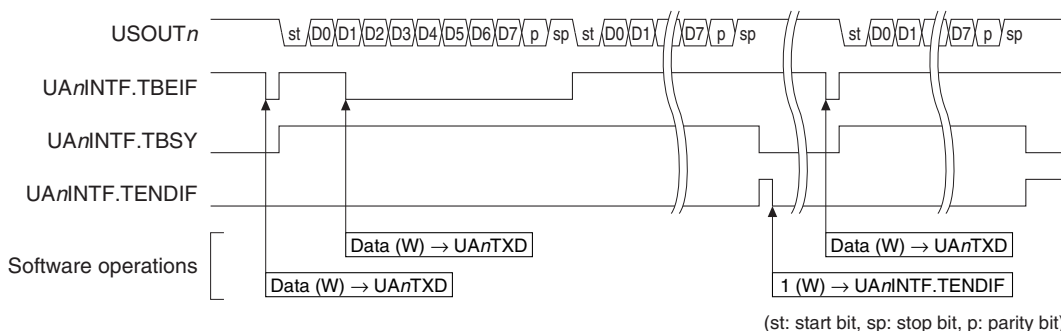


Figure 12.5.2.1 Example of Data Sending Operations

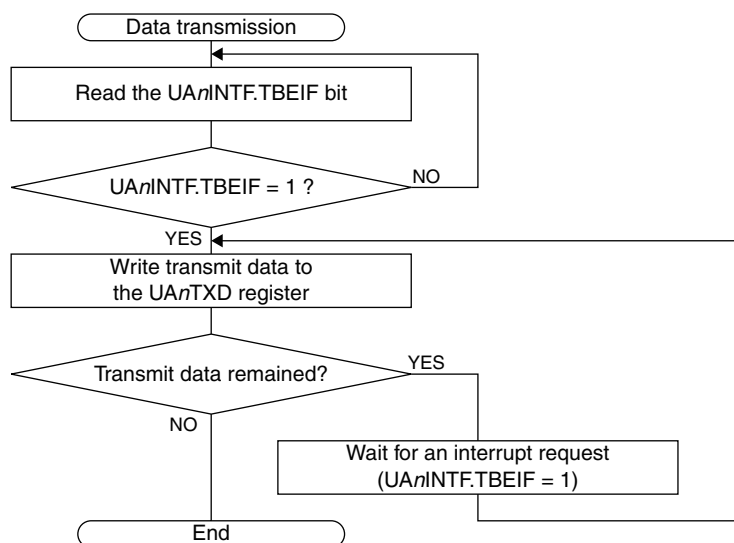


Figure 12.5.2.2 Data Transmission Flowchart

12.5.3 Data Reception

A data receiving procedure and the UART3 Ch.*n* operations are shown below. Figures 12.5.3.1 and 12.5.3.2 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 $UA_nINTF.RB1FIF$ bit is set to 1 (receive buffer one byte full).
3. Read the received data from the UA_nRXD 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 $UA_nINTF.RB2FIF$ bit is set to 1 (receive buffer two bytes full).
3. Read the received data from the UA_nRXD 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 $USIN_n$ 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 $UA_nINTF.RBSY$ bit is set to 1 when the start bit is detected.

The $UA_nINTF.RBSY$ 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 $UA_nINTF.RB1FIF$ bit to 1 (receive buffer one byte full). If the second data is received without reading the first data, the $UA_nINTF.RB2FIF$ bit is set to 1 (receive buffer two bytes full).

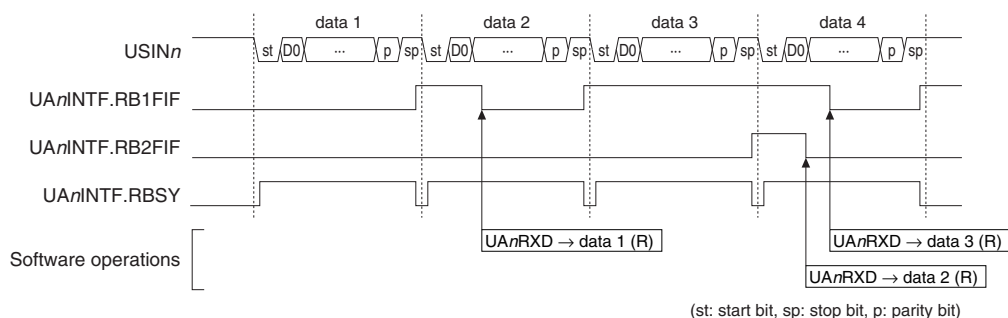


Figure 12.5.3.1 Example of Data Receiving Operations

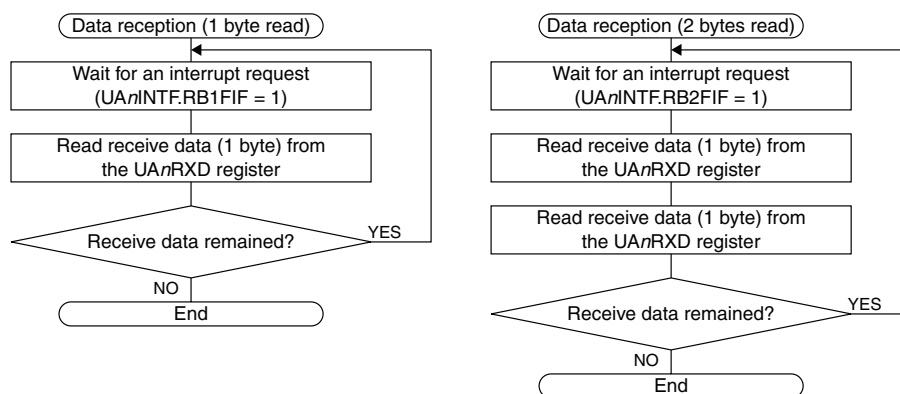


Figure 12.5.3.2 Data Reception Flowcharts

12.5.4 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.

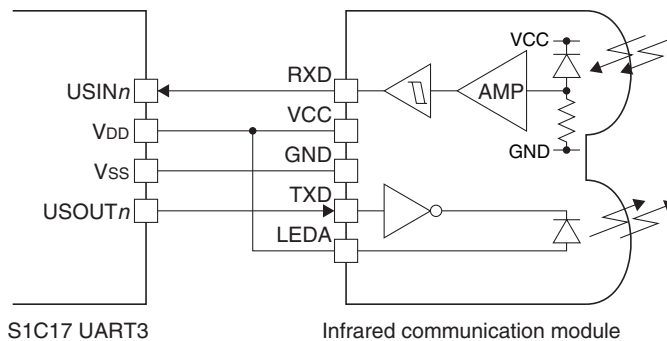


Figure 12.5.4.1 Example of Connections with an Infrared Communication Module

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.

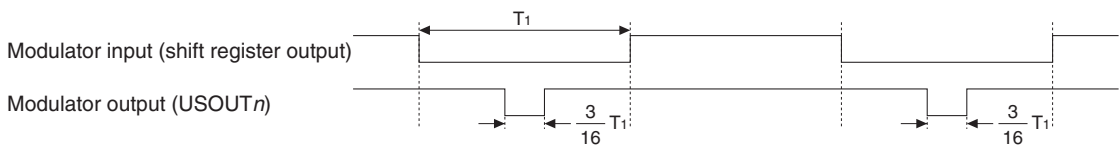


Figure 12.5.4.2 IrDA Transmission Signal Waveform

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 12.5.4.3 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.

12.5.5 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.

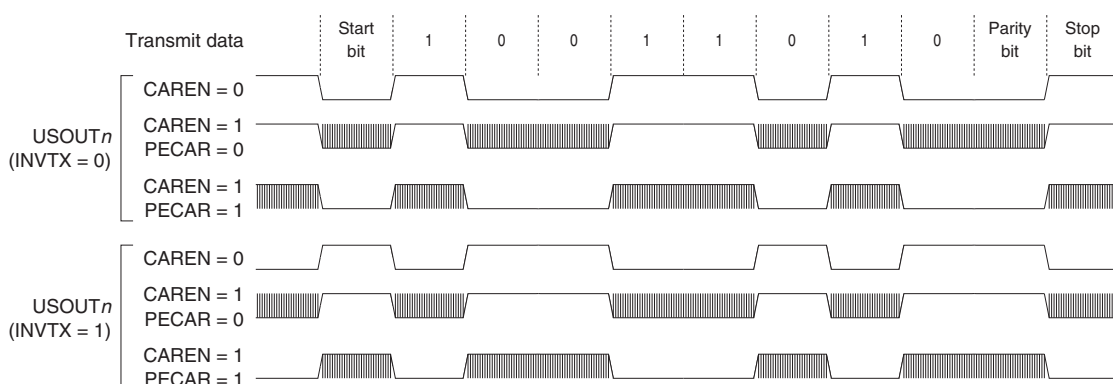


Figure 12.5.5.1 Carrier Modulation Waveform (UAnMOD.CHLN = 1, UAnMOD.STPB = 0, UAnMOD.PREN = 1)

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.

$$\text{Carrier modulation output frequency} = \frac{\text{CLK_UART3}}{(\text{CRPER} + 1) \times 2} \text{ [Hz]} \quad (\text{Eq. 12.2})$$

Where

CLK_UART3: UART3 operating clock frequency [Hz]

CRPER: UAnCAWF.CRPER[7:0] setting value (0 to 255)

12.6 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.

12.6.1 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.

12.6.2 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).

12.6.3 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.

12.7 Interrupts

The UART3 has a function to generate the interrupts shown in Table 12.7.1.

Table 12.7.1 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.

12.8 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

12 UART (UART3)

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the UART3.

Table 12.8.1 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	

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	
	3	OEIF	0	H0/S0	R/W	Cleared by writing 1.
	2	RB2FIF	0	H0/S0	R	Cleared by reading the UAnRXD register.
	1	RB1FIF	0	H0/S0	R	
	0	TBEIF	1	H0/S0	R	Cleared by writing to the UAnTXD register.

Bits 15–10 Reserved

Bit 9 RBSY

This bit indicates the receiving status. (See Figure 12.5.3.1.)

1 (R): During receiving

0 (R): Idle

Bit 8 TBSY

This bit indicates the sending status. (See Figure 12.5.2.1.)

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:

UAnINTF.TENDIF bit: End-of-transmission interrupt

UAnINTF.FEIF bit: Framing error interrupt

UAnINTF.PEIF bit: Parity error interrupt

UAnINTF.OEIF bit: Overrun error interrupt

UAnINTF.RB2FIF bit: Receive buffer two bytes full interrupt

UAnINTF.RB1FIF bit: Receive buffer one byte full interrupt

UAnINTF.TBEIF bit: Transmit buffer empty interrupt

UART3 Ch.n Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UANINTE	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:

UANINTE.TENDIE bit: End-of-transmission interrupt

UANINTE.FEIE bit: Framing error interrupt

UANINTE.PEIE bit: Parity error interrupt

UANINTE.OEIE bit: Overrun error interrupt

UANINTE.RB2FIE bit: Receive buffer two bytes full interrupt

UANINTE.RB1FIE bit: Receive buffer one byte full interrupt

UANINTE.TBEIE bit: Transmit buffer empty interrupt

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.”

13 Synchronous Serial Interface (SPIA)

13.1 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 13.1.1 shows the SPIA configuration.

Table 13.1.1 SPIA Channel Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
Number of channels	2 channels (Ch.0 and Ch.1)			
Internal clock input	Ch.0 ← 16-bit timer Ch.1 Ch.1 ← 16-bit timer Ch.2			

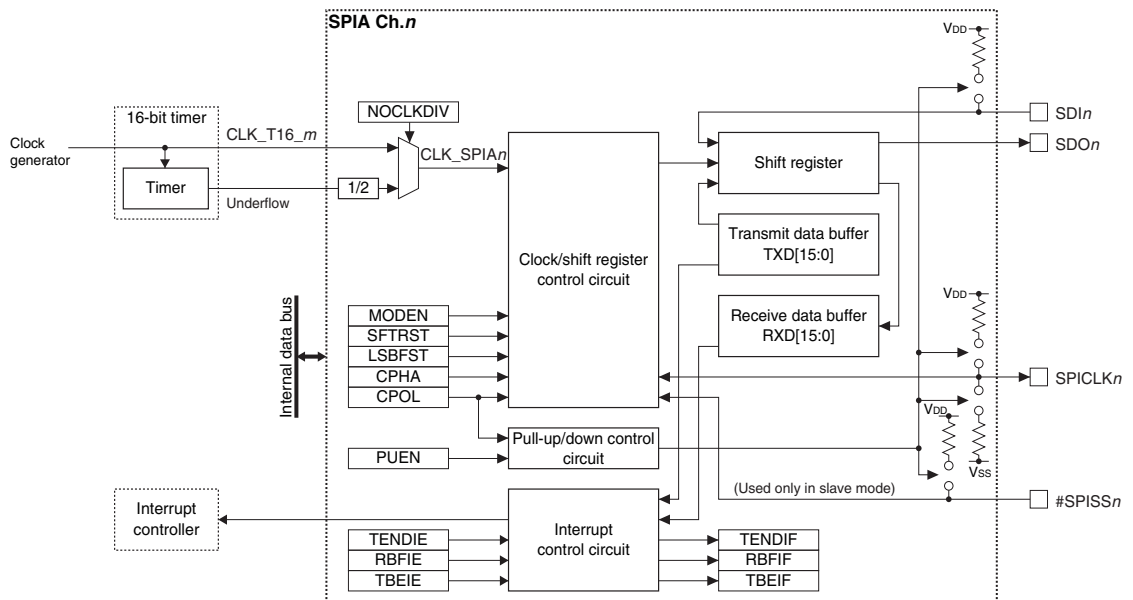


Figure 13.1.1 SPIA Configuration

13.2 Input/Output Pins and External Connections

13.2.1 List of Input/Output Pins

Table 13.2.1.1 lists the SPIA pins.

Table 13.2.1.1 List of SPIA Pins

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

13.2.2 External Connections

SPIA operates in master mode or slave mode. Figures 13.2.2.1 and 13.2.2.2 show connection diagrams between SPIA in each mode and external SPI devices.

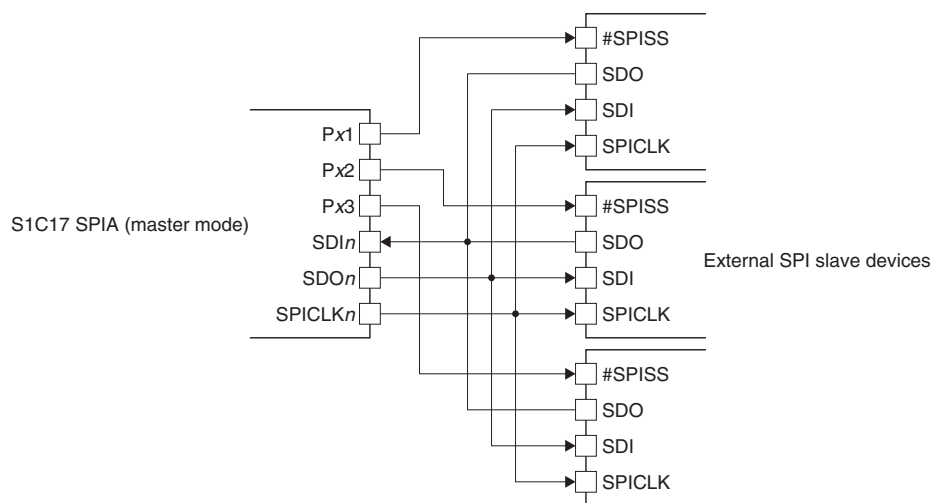


Figure 13.2.2.1 Connections between SPIA in Master Mode and External SPI Slave Devices

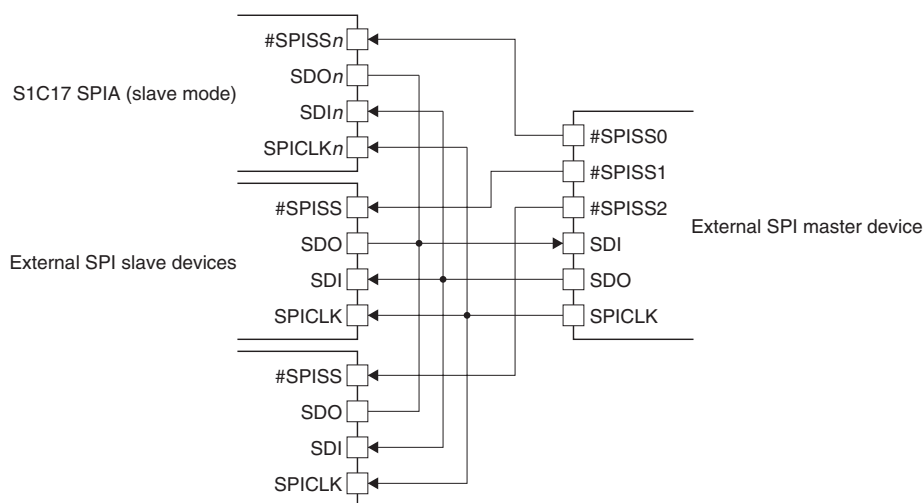


Figure 13.2.2.2 Connections between SPIA in Slave Mode and External SPI Master Device

13.2.3 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 13.2.3.1.

Table 13.2.3.1 Pin Function Differences between Modes

Pin	Function in master mode	Function in slave mode
SDIn	Always placed into input state.	
SDOn	Always placed into output state.	This pin is placed into output state while a low level is applied to the #SPISSn pin or placed into Hi-Z state while a high level is applied to the #SPISSn pin.
SPICLK _n	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.
#SPISSn	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 #SPISSn 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 SDIn pin and the clock input to the SPICLK _n 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.

13.2.4 Input Pin Pull-Up/Pull-Down Function

The SPIA input pins (SDIn in master mode or SDIn, SPICLK_n, and #SPISSn pins in slave mode) have a pull-up or pull-down function as shown in Table 13.2.4.1. This function is enabled by setting the SPInMOD.PUEN bit to 1.

Table 13.2.4.1 Pull-Up or Pull-Down of Input Pins

Pin	Master mode	Slave mode
SDIn	Pull-up	Pull-up
SPICLK _n	—	SPInMOD.CPOL bit = 1: Pull-up SPInMOD.CPOL bit = 0: Pull-down
#SPISSn	—	Pull-up

13.3 Clock Settings

13.3.1 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 SPInMOD.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 SPInMOD.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.

$$f_{\text{SPICLK}} = \frac{f_{\text{CLK_SPIA}}}{2 \times (\text{RLD} + 1)}$$

$$\text{RLD} = \frac{f_{\text{CLK_SPIA}}}{f_{\text{SPICLK}} \times 2} - 1 \quad (\text{Eq. 13.1})$$

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.

13.3.2 Clock Supply in DEBUG Mode

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

The CLK_T16 $_m$ supply to SPIA Ch. n is suspended when the CPU enters DEBUG mode if the T16 $_m$ CLK.DB-RUN 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.

13.3.3 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 13.3.3.1 shows the clock waveform and data input/output timing in each setting.

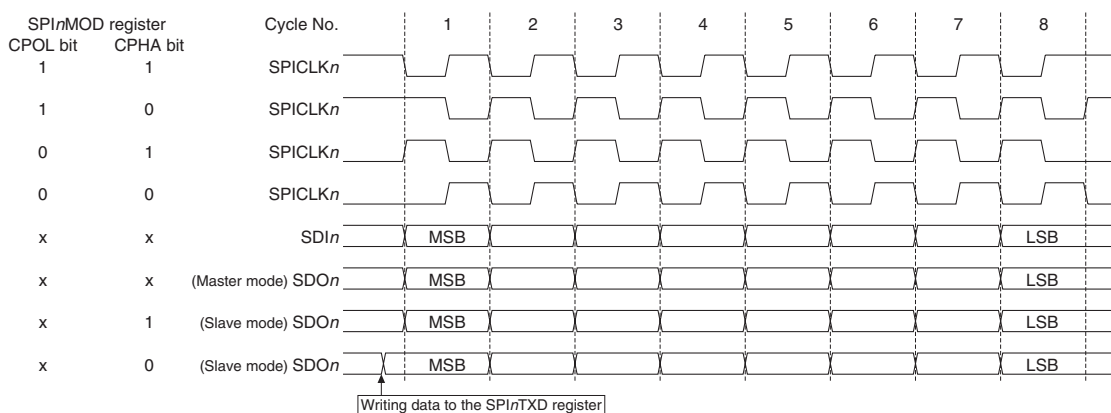


Figure 13.3.3.1 SPI Clock Phase and Polarity (SPI n MOD.LSBFST bit = 0, SPI n MOD.CHLN[3:0] bits = 0x7)

13.4 Data Format

The SPIA data length can be selected from 2 bits to 16 bits by setting the `SPInMOD.CHNLN[3:0]` bits. The input/output permutation is configurable to MSB first or LSB first using the `SPInMOD.LSBFST` bit. Figure 13.4.1 shows a data format example when the `SPInMOD.CHNLN[3:0]` bits = 0x7, the `SPInMOD.CPOL` bit = 0 and the `SPInMOD.CPHA` bit = 0.

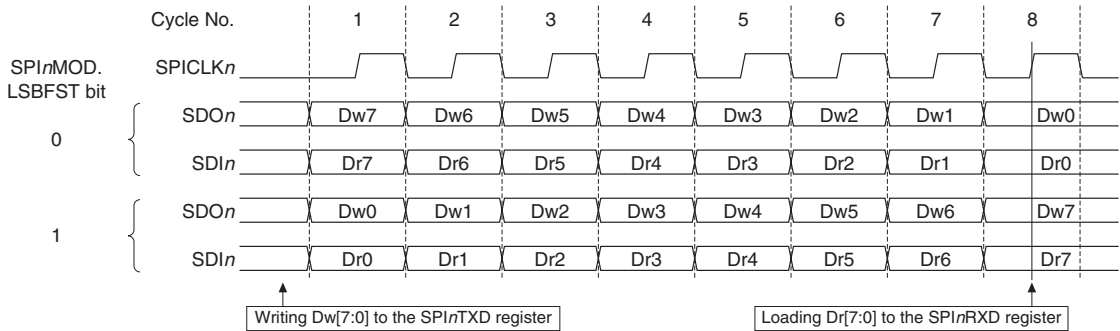


Figure 13.4.1 Data Format Selection Using the `SPInMOD.LSBFST` Bit
(`SPInMOD.CHNLN[3:0]` bits = 0x7, `SPInMOD.CPOL` bit = 0, `SPInMOD.CPHA` bit = 0)

13.5 Operations

13.5.1 Initialization

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

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

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

13.5.2 Data Transmission in Master Mode

A data sending procedure and operations in master mode are shown below. Figures 13.5.2.1 and 13.5.2.2 show a timing chart and a flowchart, respectively.

Data sending procedure

1. Assert the slave select signal by controlling the general-purpose output port (if necessary).
2. Check to see if the `SPInINTF.TBEIF` bit is set to 1 (transmit buffer empty).
3. Write transmit data to the `SPInTXD` register.

4. Wait for an SPIA interrupt when using the interrupt.
5. Repeat Steps 2 to 4 (or 2 and 3) until the end of transmit data.
6. Negate the slave select signal by controlling the general-purpose output port (if necessary).

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.

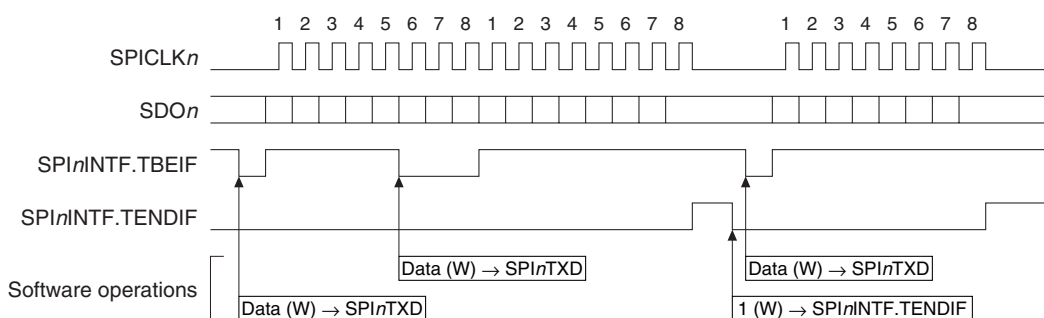


Figure 13.5.2.1 Example of Data Sending Operations in Master Mode (SPI*n*MOD.CHLN[3:0] bits = 0x7)

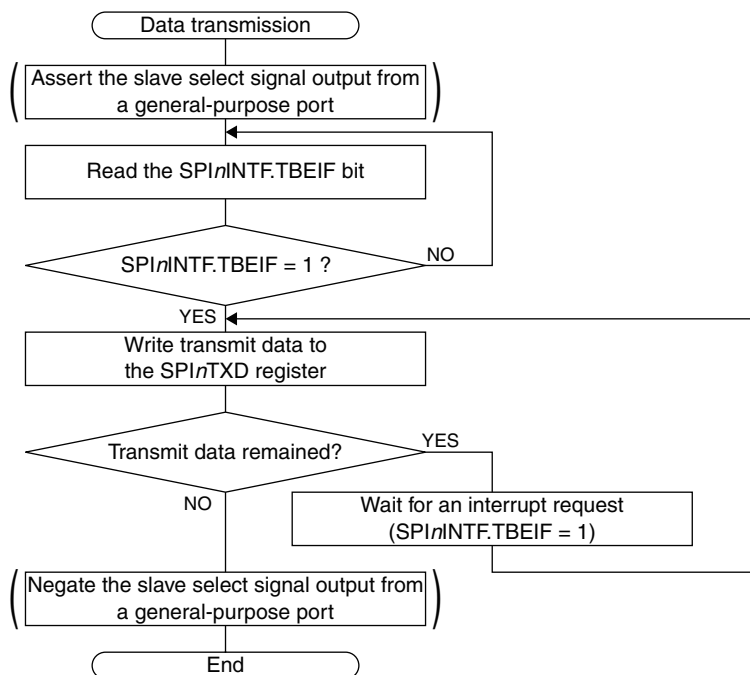


Figure 13.5.2.2 Data Transmission Flowchart in Master Mode

13.5.3 Data Reception in Master Mode

A data receiving procedure and operations in master mode are shown below. Figures 13.5.3.1 and 13.5.3.2 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 $SPI_nINTF.TBEIF$ bit is set to 1 (transmit buffer empty).
3. Write dummy data (or transmit data) to the SPI_nTXD register.
4. Wait for a transmit buffer empty interrupt ($SPI_nINTF.TBEIF$ bit = 1).
5. Write dummy data (or transmit data) to the SPI_nTXD register.
6. Wait for a receive buffer full interrupt ($SPI_nINTF.RBFIF$ bit = 1).
7. Read the received data from the SPI_nRXD 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 $SPICLK_n$, Steps 7 and 5 operations must be completed within the $SPICLK_n$ 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_nTXD register.

The $SPICLK_n$ pin outputs clocks of the number of the bits specified by the $SPI_nMOD.CHLN[3:0]$ bits. The transmit data bits are output in sequence from the $SDOn$ pin in sync with these clocks and the receive data bits input from the $SDIn$ pin are shifted into the shift register.

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

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

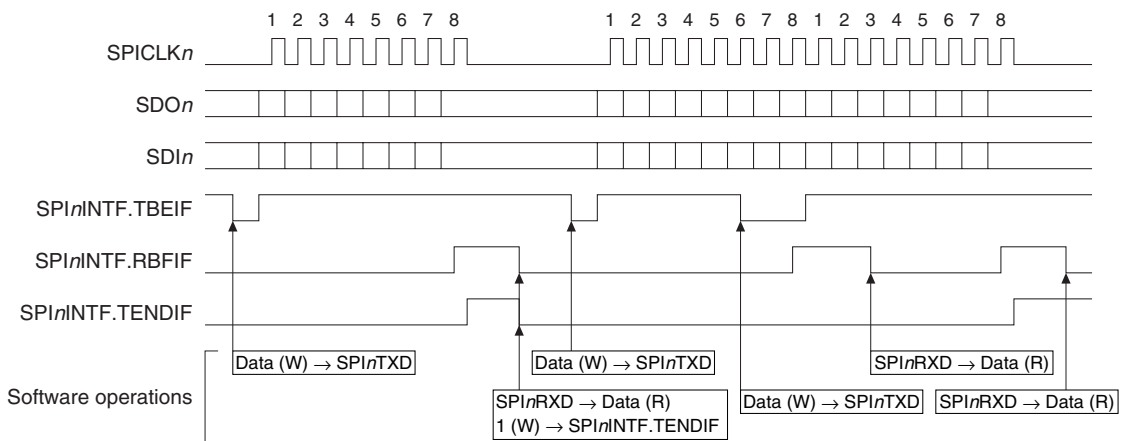


Figure 13.5.3.1 Example of Data Receiving Operations in Master Mode ($SPI_nMOD.CHLN[3:0]$ bits = 0x7)

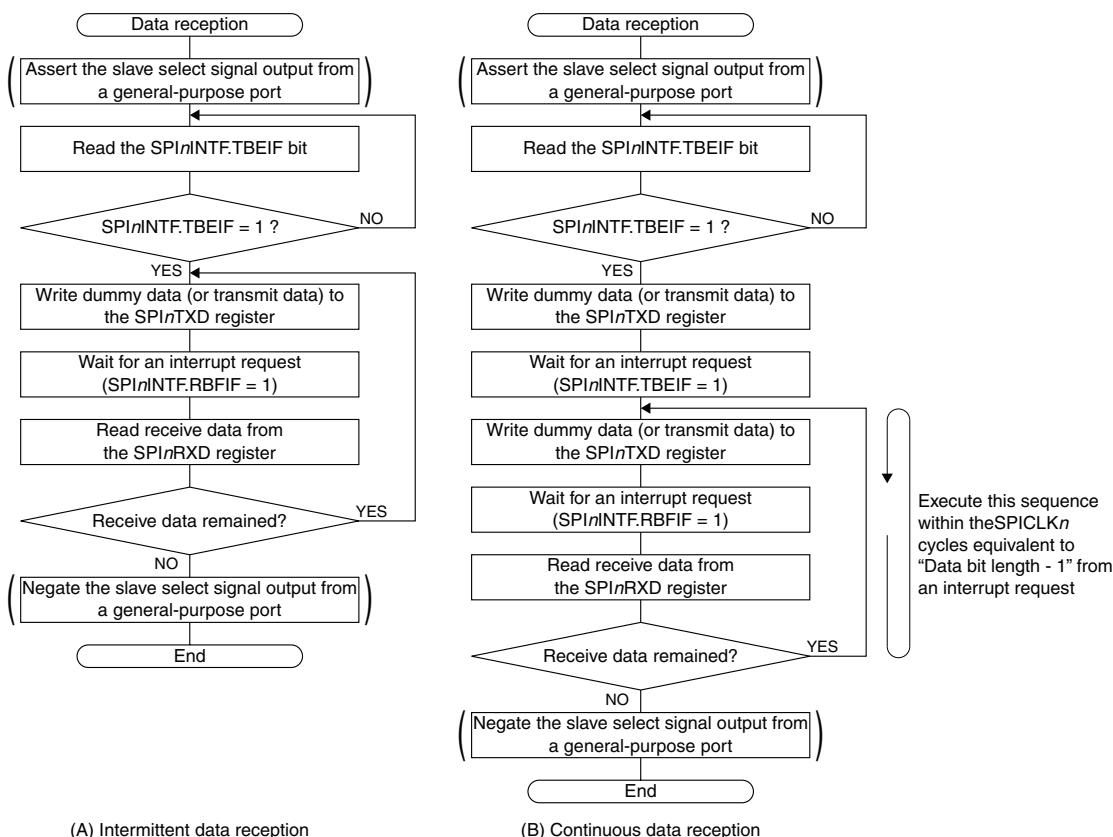


Figure 13.5.3.2 Data Reception Flowcharts in Master Mode

13.5.4 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 (SPInINTF.TENDIF bit = 1).
2. Set the SPInCTL.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*.

13.5.5 Data Transfer in Slave Mode

A data sending/receiving procedure and operations in slave mode are shown below. Figures 13.5.5.1 and 13.5.5.2 show a timing chart and flowcharts, respectively.

Data sending procedure

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

Note: Transmit data must be written to the SPInTXD register after the SPInINTF.TBEIF bit is set to 1 by the time the sending SPInTXD 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 SDO_n 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 SDIn 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.

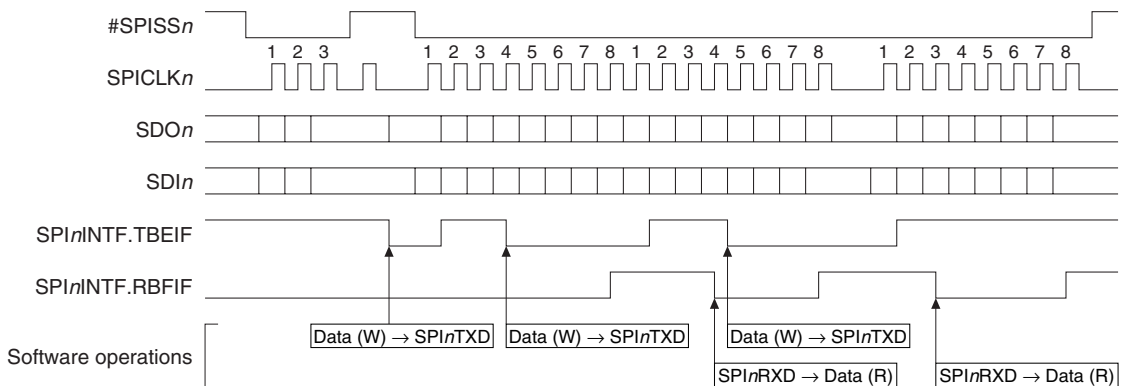


Figure 13.5.5.1 Example of Data Transfer Operations in Slave Mode (SPI n MOD.CHLN[3:0] bits = 0x7)

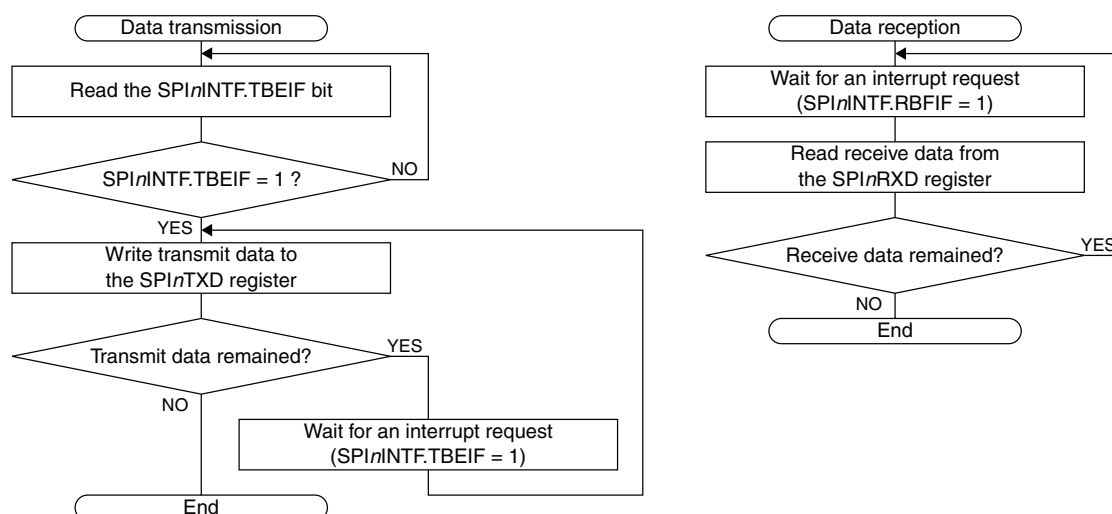


Figure 13.5.5.2 Data Transfer Flowcharts in Slave Mode

13.5.6 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 (SPInINTF.TENDIF bit = 1). Or determine end of transfer via the received data.
2. Set the SPInCTL.MODEN bit to 0 to disable the SPIA Ch.*n* operations.

13.6 Interrupts

SPIA has a function to generate the interrupts shown in Table 13.6.1.

Table 13.6.1 SPIA Interrupt Function

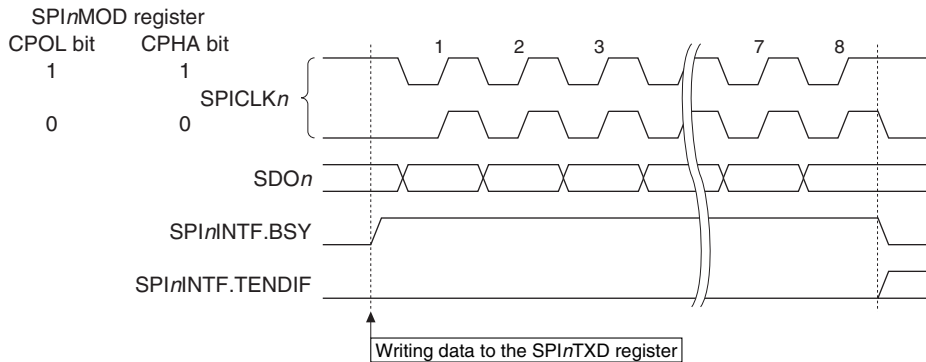
Interrupt	Interrupt flag	Set condition	Clear condition
End of transmission	SPInINTF.TENDIF	When the SPInINTF.TBEIF bit = 1 after data of the specified bit length (defined by the SPInMOD.CHLN[3:0] bits) has been sent	Writing 1
Receive buffer full	SPInINTF.RBFIF	When data of the specified bit length is received and the received data is transferred from the shift register to the received data buffer	Reading the SPInRXD register
Transmit buffer empty	SPInINTF.TBEIF	When transmit data written to the transmit data buffer is transferred to the shift register	Writing to the SPInTXD register
Overrun error	SPInINTF.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.

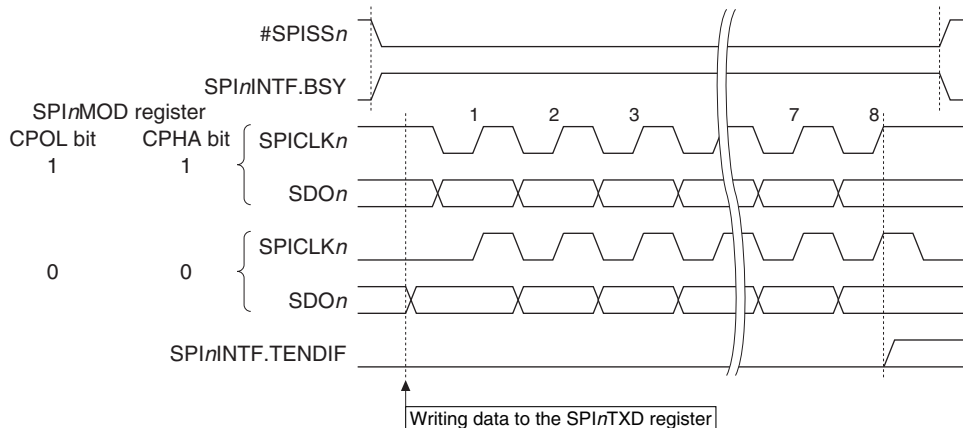
The SPInINTF register also contains the BSY bit that indicates the SPIA operating status.

Figure 13.6.1 shows the SPInINTF.BSY and SPInINTF.TENDIF bit set timings.

Master mode



Slave mode

Figure 13.6.1 SPI n INTF.BSY and SPI n INTF.TENDIF Bit Set Timings (when SPI n MOD.CHLN[3:0] bits = 0x7)

13.7 Control Registers

SPIA Ch. n Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPI n MOD	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.

13 SYNCHRONOUS SERIAL INTERFACE (SPIA)

Table 13.7.1 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 prohibited

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

SPIA Ch.*n* Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
<code>SPI_nTXD</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 `SPInINTF.TBEIF` bit = 1 regardless of whether data is being output from the `SDOn` pin or not.

Note that the upper data bits that exceed the data bit length configured by the `SPInMOD.CHNLN[3:0]` bits will not be output from the `SDOn` pin.

Note: Be sure to avoid writing to the `SPInTXD` register when the `SPInINTF.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>SPI_nRXD</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 `SPInINTF.`

`RBFIF` bit = 1 regardless of whether data is being input from the `SDIn` pin or not. Note that the upper bits that exceed the data bit length configured by the `SPInMOD.CHNLN[3:0]` bits become 0.

Note: The `SPInRXD.RXD[15:0]` bits are cleared to 0x0000 when 1 is written to the `SPInCTL.MODEN` bit or the `SPInCTL.SFTRST` bit.

SPIA Ch.*n* Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
<code>SPI_nINTF</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>SPI_nRXD</code> register.
	0	<code>TBEIF</code>	1	H0/S0	R	Cleared by writing to the <code>SPI_nTXD</code> register.

13 SYNCHRONOUS SERIAL INTERFACE (SPIA)

Bits 15–8 Reserved

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

14 I²C (I2C)

14.1 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.

Figure 14.1.1 shows the I2C configuration.

Table 14.1.1 I2C Channel Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
Number of channels	1 channel (Ch.0)			

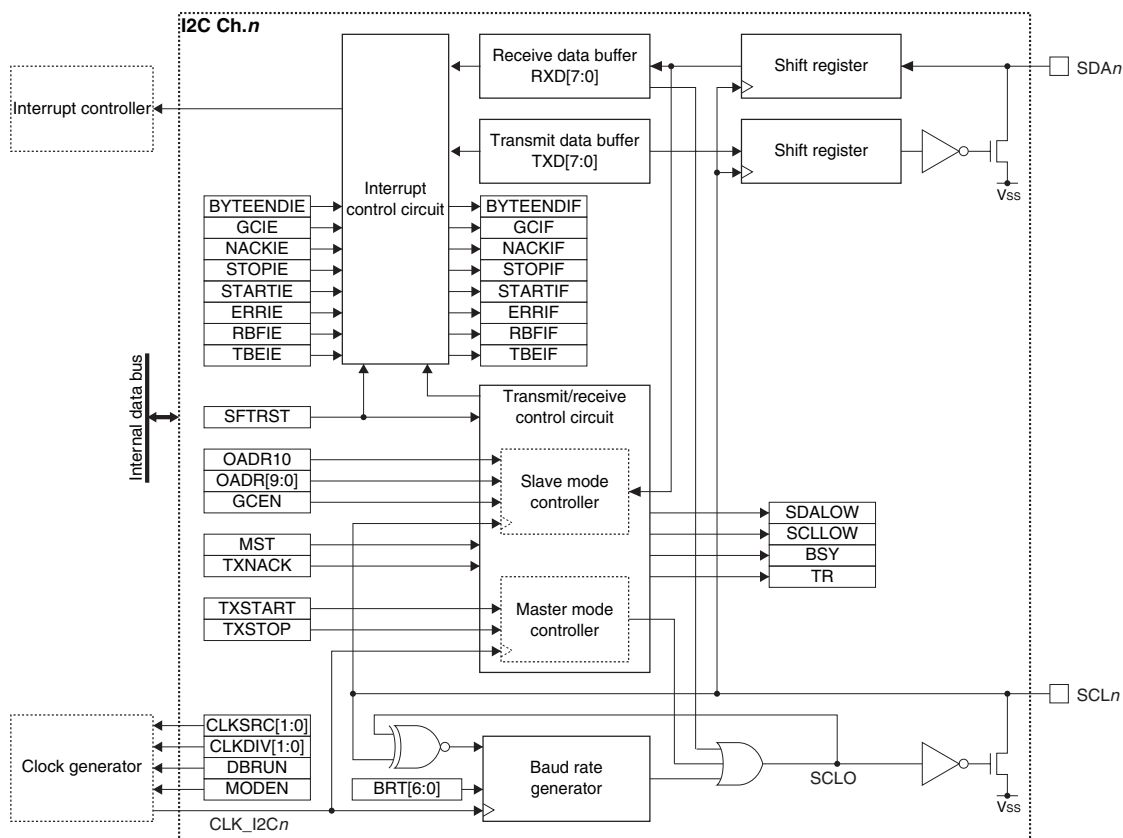


Figure 14.1.1 I2C Configuration

14.2 Input/Output Pins and External Connections

14.2.1 List of Input/Output Pins

Table 14.2.1.1 lists the I²C pins.

Table 14.2.1.1 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.

14.2.2 External Connections

Figure 14.2.2.1 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.

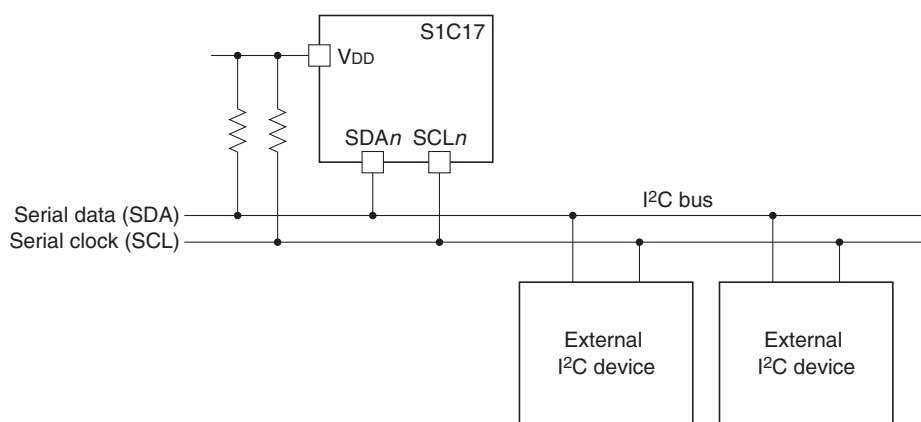


Figure 14.2.2.1 Connections between I²C and External I²C Devices

- 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.

14.3 Clock Settings

14.3.1 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 I2C*n*CLK register bits:
 - I2C*n*CLK.CLKSRC[1:0] bits (Clock source selection)
 - I2C*n*CLK.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 I2C*n*CLK 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.

14.3.2 Clock Supply in DEBUG Mode

In master mode, the CLK_I2C*n* supply during DEBUG mode should be controlled using the I2C*n*CLK.DBRUN bit. The CLK_I2C*n* supply to the I2C Ch.*n* is suspended when the CPU enters DEBUG mode if the I2C*n*CLK.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 I2C*n*CLK.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.

14.3.3 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 I2C*n*BR.BRT[6:0] bit settings. Use the following equations to calculate the setting values for obtaining the desired transfer rate.

$$\text{bps} = \frac{f_{\text{CLK_I2C}n}}{(\text{BRT} + 3) \times 2} \qquad \text{BRT} = \frac{f_{\text{CLK_I2C}n}}{\text{bps} \times 2} - 3 \qquad (\text{Eq. 14.1})$$

Where

bps: Data transfer rate [bit/s]

f_{CLK_I2C*n*}: I2C operating clock frequency [Hz]

BRT: I2C*n*BR.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 14.3.3.1).

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 14.3.3.1 shows the clock generated by the baud rate generator and the clock waveform on the I²C bus.

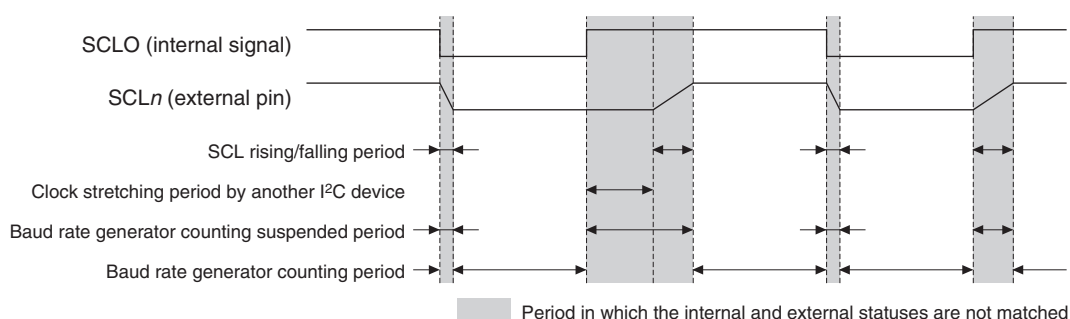


Figure 14.3.3.1 Baud Rate Generator Output Clock and SCL_n Output Waveform

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.

14.4 Operations

14.4.1 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)

14.4.2 Data Transmission in Master Mode

A data sending procedure in master mode and the I2C Ch.*n* operations are shown below. Figures 14.4.2.1 and 14.4.2.2 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.

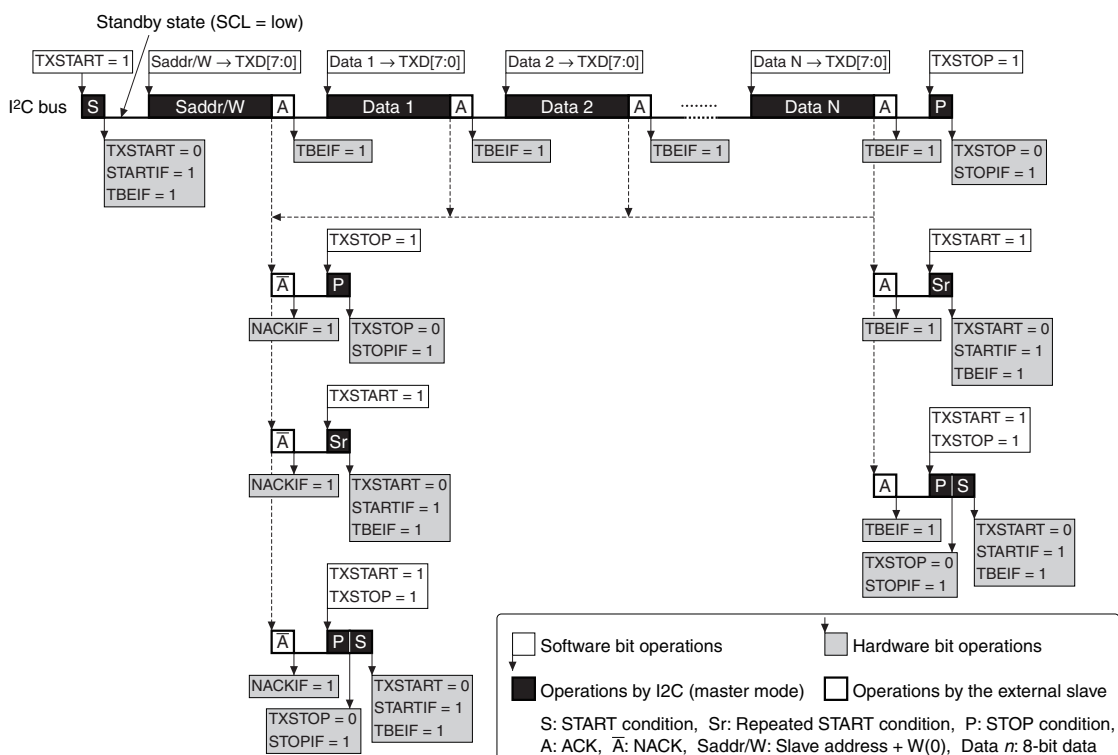


Figure 14.4.2.1 Example of Data Sending Operations in Master Mode

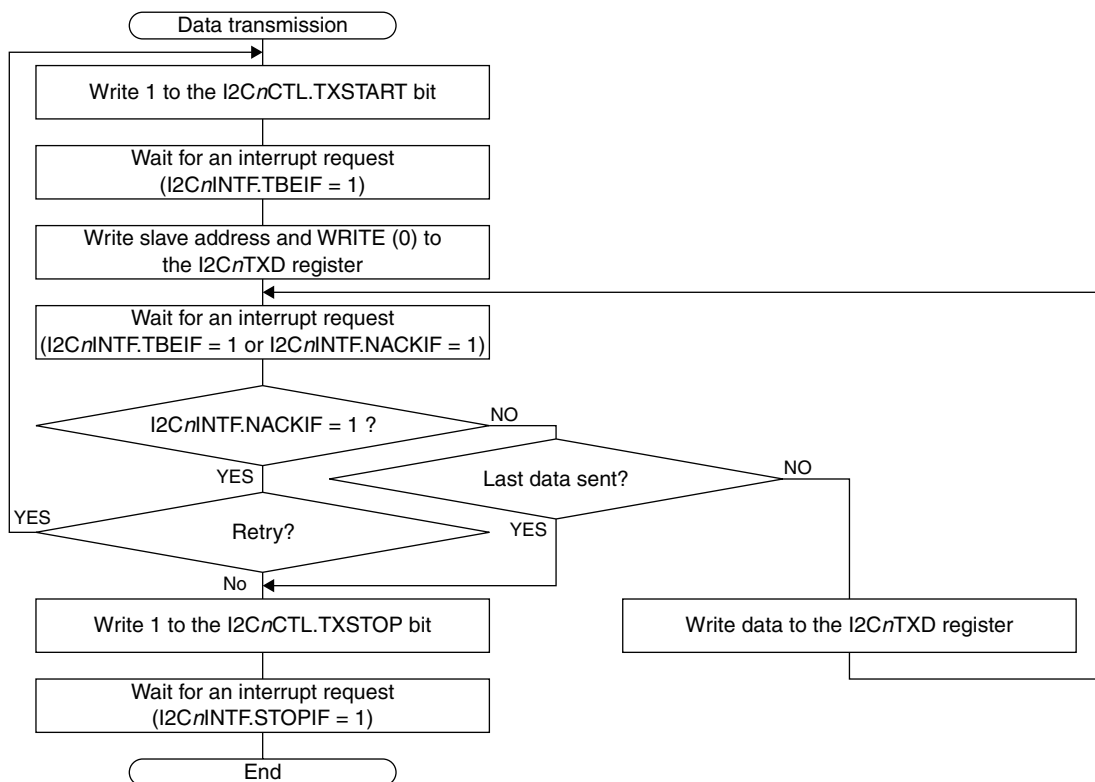


Figure 14.4.2.2 Master Mode Data Transmission Flowchart

14.4.3 Data Reception in Master Mode

A data receiving procedure in master mode and the I2C Ch.*n* operations are shown below. Figures 14.4.3.1 and 14.4.3.2 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.

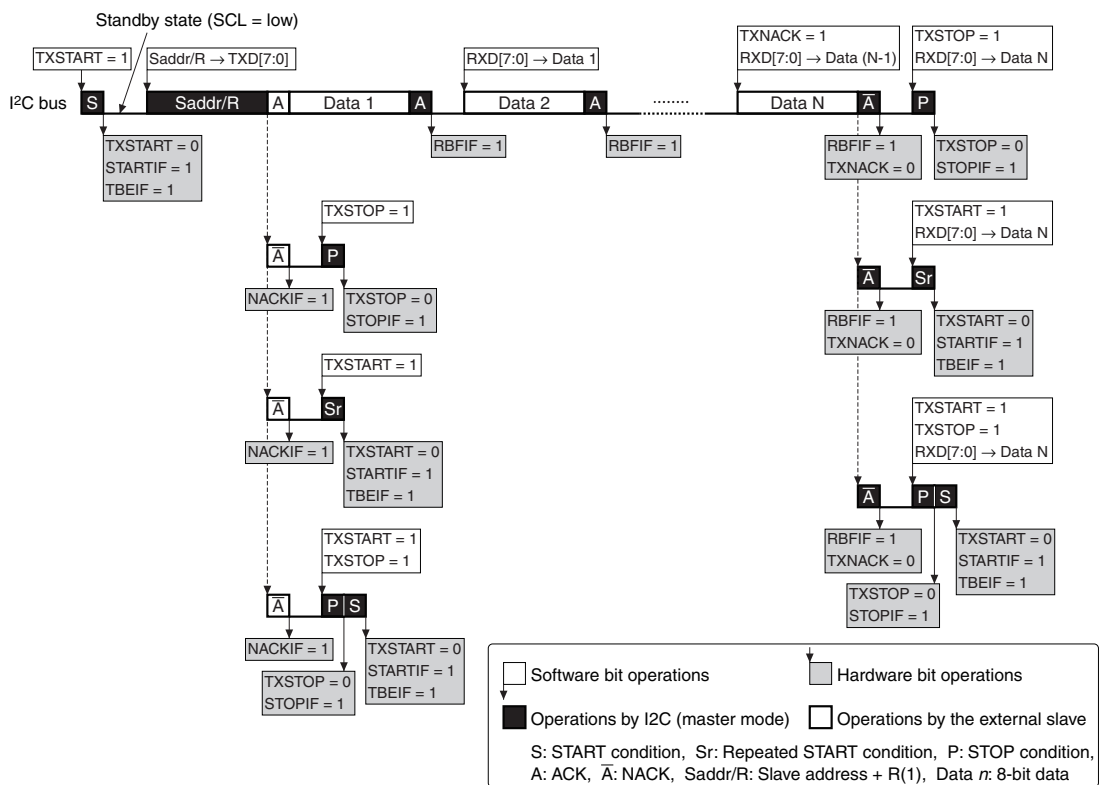


Figure 14.4.3.1 Example of Data Receiving Operations in Master Mode

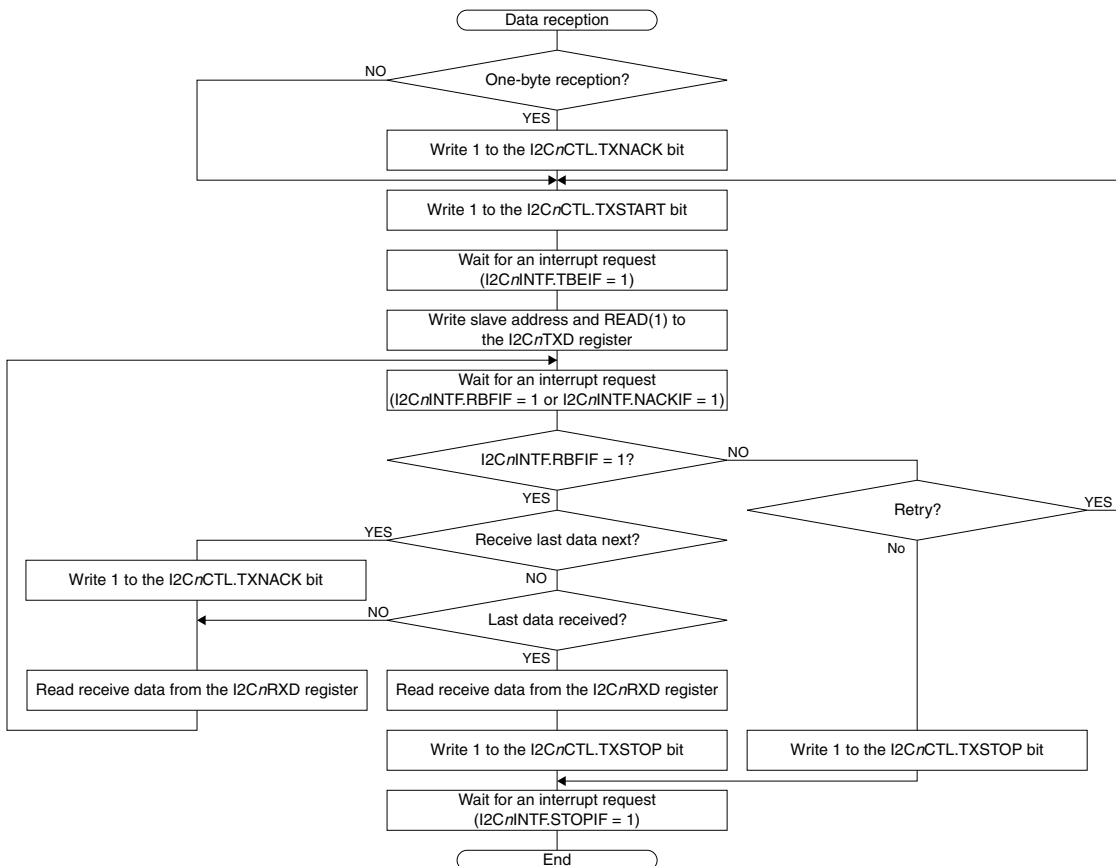
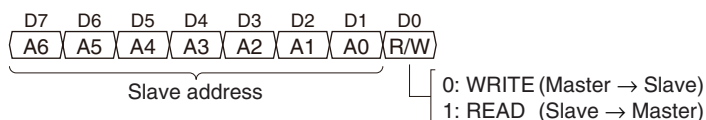


Figure 14.4.3.2 Master Mode Data Reception Flowchart

14.4.4 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.

7-bit address



10-bit address

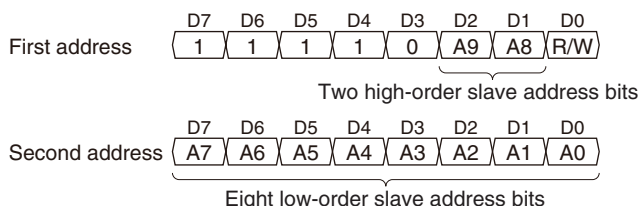


Figure 14.4.4.1 10-bit Address Configuration

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 14.4.4.2 shows an operation example.

Starting data transmission in 10-bit address mode

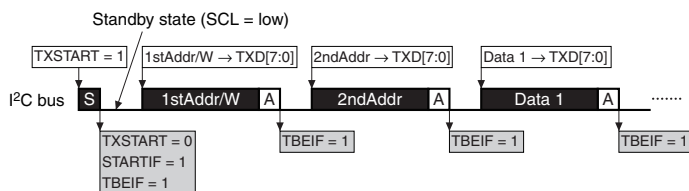
1. Issue a START condition by setting the I2CnCTL.TXSTART bit to 1.
2. Wait for a transmit buffer empty interrupt (I2CnINTF.TBEIF bit = 1) or a START condition interrupt (I2CnINTF.STARTIF bit = 1).
Clear the I2CnINTF.STARTIF bit by writing 1 after the interrupt has occurred.
3. Write the first address 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.

14 I²C (I2C)

At start of data transmission



At start of data reception

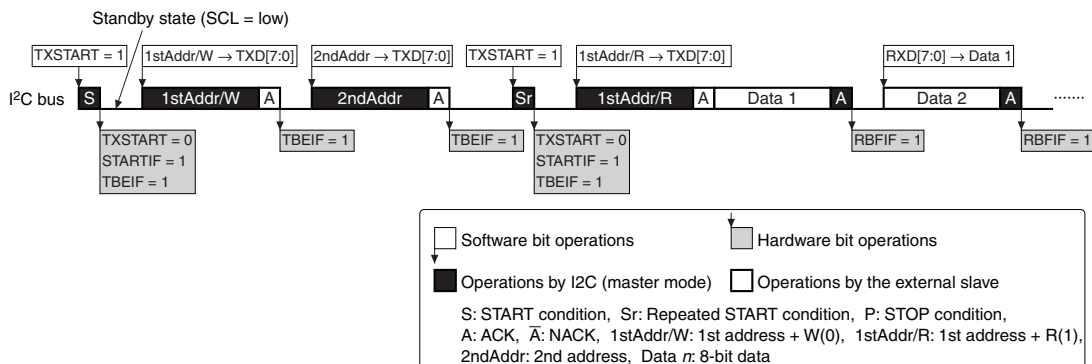


Figure 14.4.4.2 Example of Data Transfer Starting Operations in 10-bit Address Mode (Master Mode)

14.4.5 Data Transmission in Slave Mode

A data sending procedure in slave mode and the I2C Ch.*n* operations are shown below. Figures 14.4.5.1 and 14.4.5.2 show an operation example and a flowchart, respectively.

Data sending procedure

1. Wait for 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.
2. Check to see if the I2C*n*INTF.TR bit = 1 (transmission mode).
(Start a data receiving procedure if the I2C*n*INTF.TR bit = 0.)
3. Write transmit data to the I2C*n*TXD register.
4. Wait for a transmit buffer empty interrupt (I2C*n*INTF.TBEIF bit = 1), a NACK reception interrupt (I2C-*n*INTF.NACKIF bit = 1), or a STOP condition interrupt (I2C*n*INTF.STOPIF bit = 1).
 - i. Go to Step 3 when a transmit buffer empty interrupt has occurred.
 - ii. Go to Step 5 after clearing the I2C*n*INTF.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 (I2C*n*INTF.STOPIF bit = 1) or a START condition interrupt (I2C*n*INTF.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 I2C*n*INTF.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.

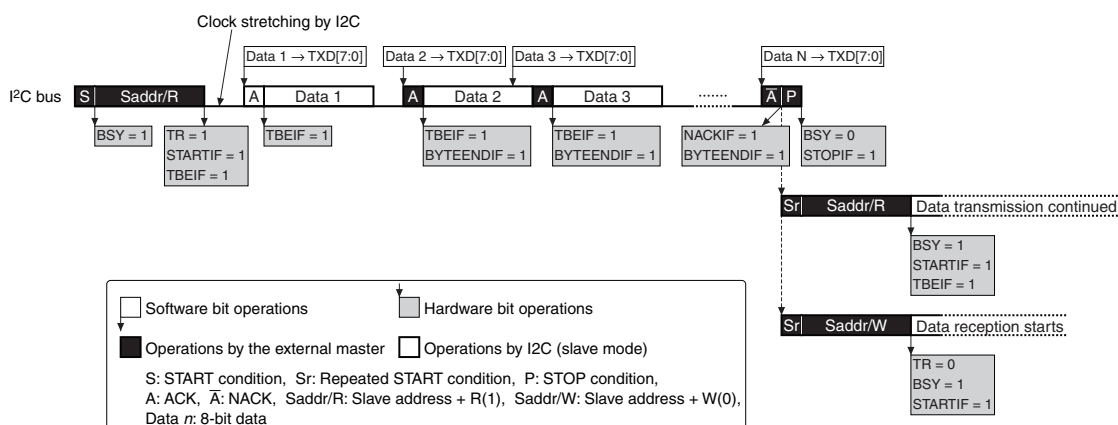
Sending subsequent data

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.



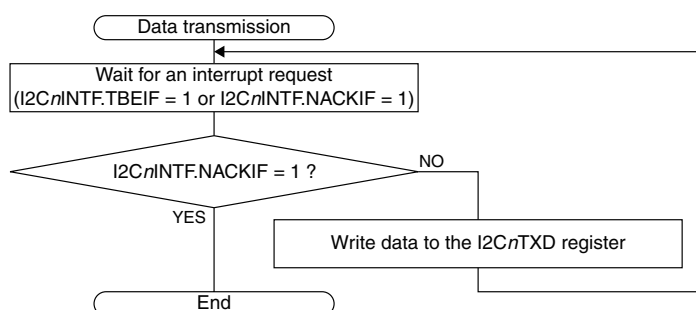


Figure 14.4.5.2 Slave Mode Data Transmission Flowchart

14.4.6 Data Reception in Slave Mode

A data receiving procedure in slave mode and the I2C Ch.n operations are shown below. Figures 14.4.6.1 and 14.4.6.2 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.

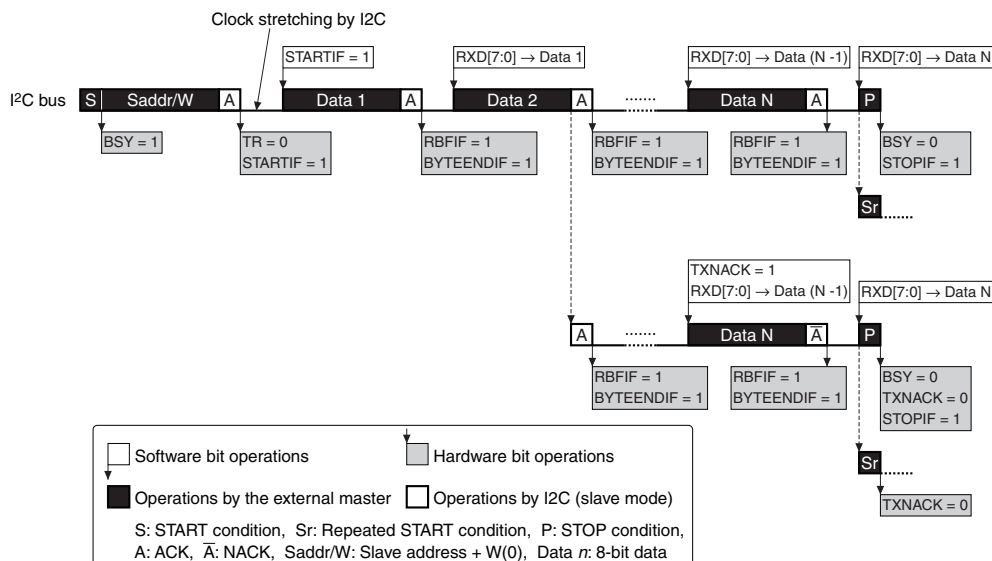


Figure 14.4.6.1 Example of Data Receiving Operations in Slave Mode

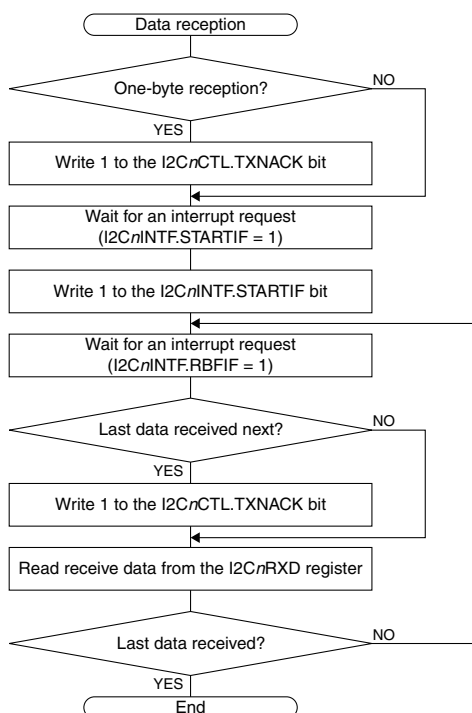


Figure 14.4.6.2 Slave Mode Data Reception Flowchart

14.4.7 Slave Operations in 10-bit Address Mode

The I2C Ch.*n* functions as a slave device in 10-bit address mode when the I2CnCTL.MST bit = 0 and the I2C-nMOD.OADR10 bit = 1.

The following shows the address receiving operations in 10-bit address mode. Figure 14.4.7.1 shows an operation example. See Figure 14.4.4.1 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 I2CnO-ADR.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 I2CnOADR.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.

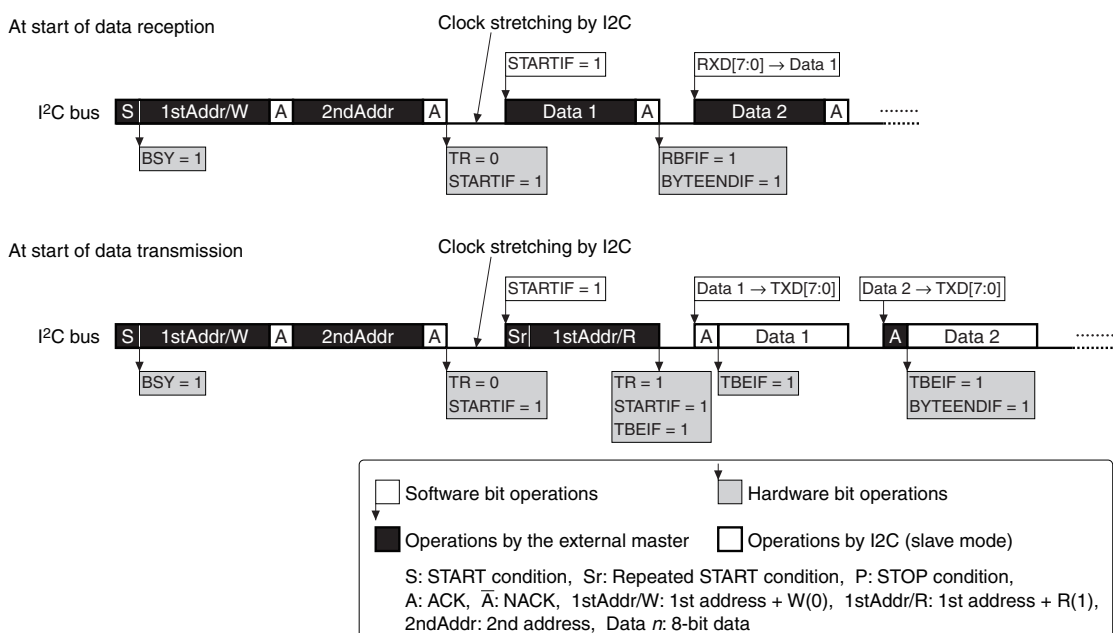


Figure 14.4.7.1 Example of Data Transfer Starting Operations in 10-bit Address Mode (Slave Mode)

14.4.8 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 SCLn 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 I2CnCTL.TXSTART bit to 0 and sets both the I2CnINTF.ERRIF and I2CnINTF.STARTIF bits to 1.

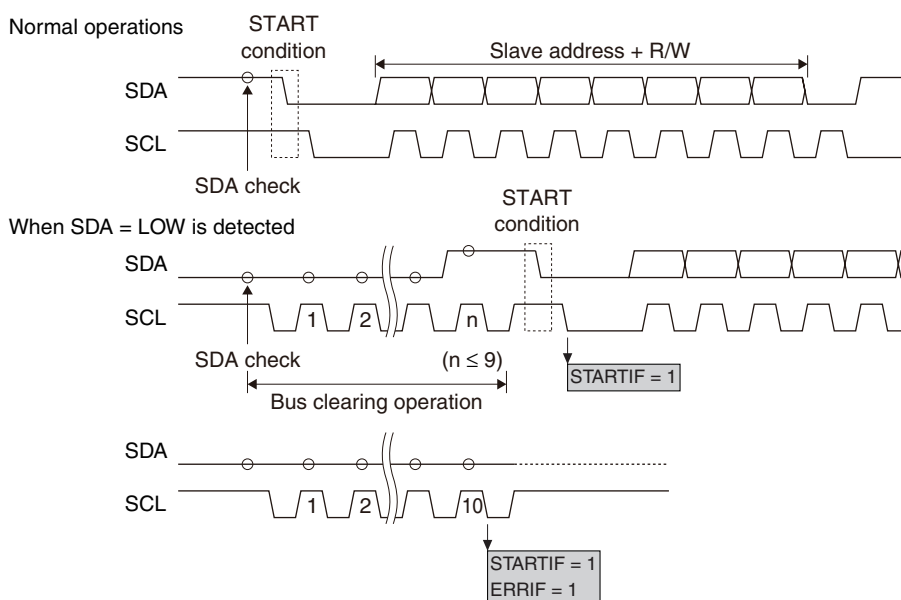


Figure 14.4.8.1 Automatic Bus Clearing Operation

14.4.9 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 14.4.9.1 Hardware Error Detection Function

No.	Error detecting period/timing	I ² C bus line monitored and error condition	Notification method
1	While the I2C Ch.n controls SDA to high for sending address, data, or a NACK	SDA = low	I2CnINTF.ERRIF = 1
2	<Master mode only> When 1 is written to the I2CnCTL.TX-START bit while the I2CnINTF.BSY bit = 0	SCL = low	I2CnINTF.ERRIF = 1 I2CnCTL.TXSTART = 0 I2CnINTF.STARTIF = 1
3	<Master mode only> When 1 is written to the I2CnCTL.TXSTOP bit while the I2CnINTF.BSY bit = 0	SCL = low	I2CnINTF.ERRIF = 1 I2CnCTL.TXSTOP = 0 I2CnINTF.STOPIF = 1
4	<Master mode only> When 1 is written to the I2CnCTL.TX-START bit while the I2CnINTF.BSY bit = 0 (Refer to “Automatic Bus Clearing Operation.”)	SDA Automatic bus clearing failure	I2CnINTF.ERRIF = 1 I2CnCTL.TXSTART = 0 I2CnINTF.STARTIF = 1

14.5 Interrupts

The I2C has a function to generate the interrupts shown in Table 14.5.1.

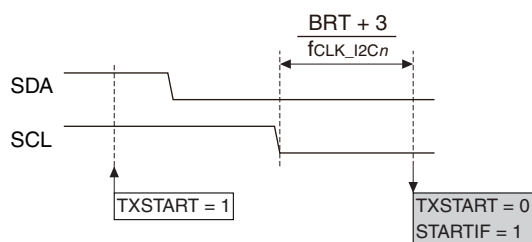
Table 14.5.1 I2C Interrupt Function

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. <i>n</i> 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

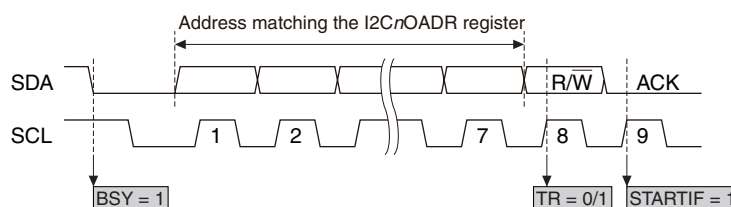
The I2C provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

(1) START condition interrupt

Master mode

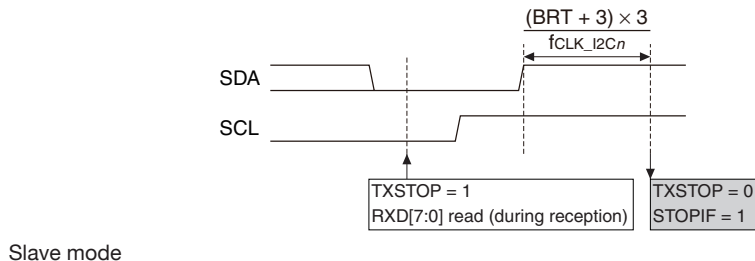


Slave mode



(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 14.5.1 START/STOP Condition Interrupt Timings

14.6 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 14.6.1 Clock Source and Division Ratio Settings

I2CnCLK. CLKDIV[1:0] bits	I2CnCLK.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 I2CnCLK register settings can be altered only when the I2CnCTL.MODEN bit = 0.

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

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	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	Cleared by writing 1.
	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 SDALLOW

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

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

15 16-bit PWM Timers (T16B)

15.1 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 15.1.1 shows the T16B configuration.

Table 15.1.1 T16B Channel Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
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.

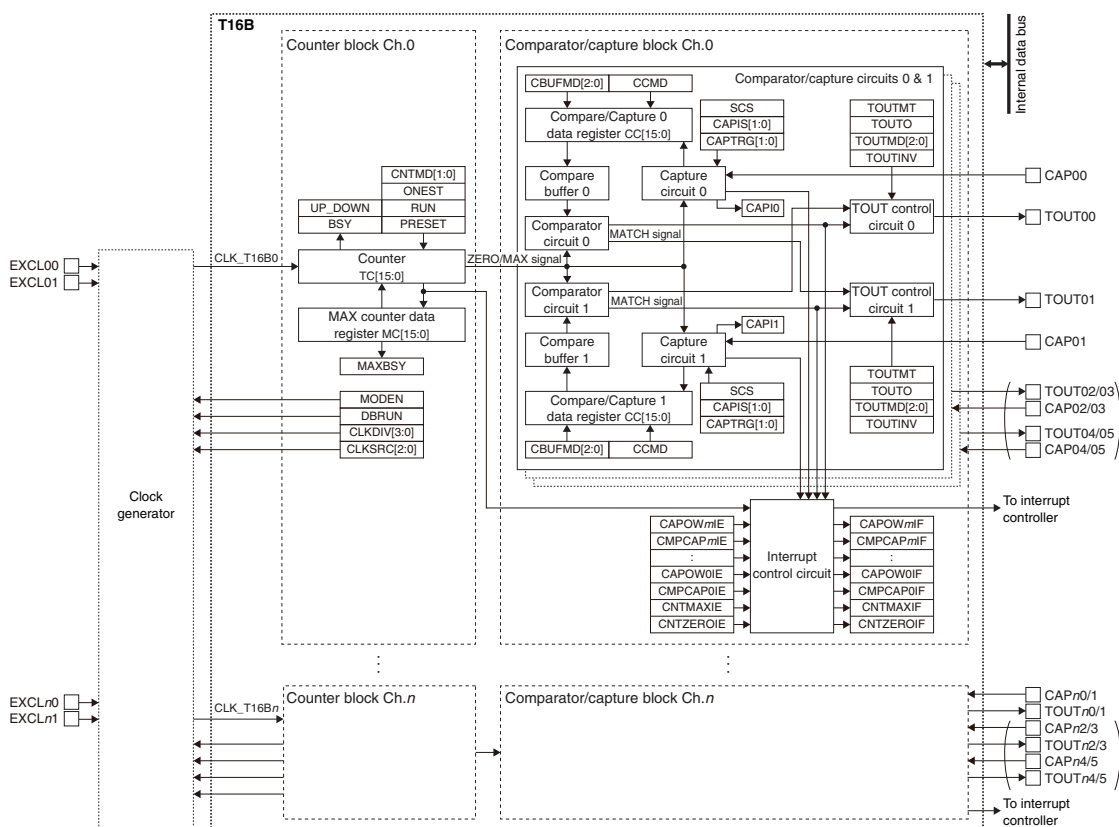


Figure 15.1.1 T16B Configuration

15.2 Input/Output Pins

Table 15.2.1 lists the T16B pins.

Table 15.2.1 List of T16B Pins

Pin name	I/O*	Initial status*	Function
EXCL n m	I	I (Hi-Z)	External clock input
TOUT n m/CAP n m	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.

15.3 Clock Settings

15.3.1 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 T16BnCLK register bits:
 - T16BnCLK.CLKSRC[2:0] bits (Clock source selection)
 - T16BnCLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)

15.3.2 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.

15.3.3 Clock Supply in DEBUG Mode

The CLK_T16B*n* supply during DEBUG mode should be controlled using the T16BnCLK.DBRUN bit.

The CLK_T16B*n* supply to T16B Ch.*n* is suspended when the CPU enters DEBUG mode if the T16BnCLK.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 T16BnCLK.DBRUN bit = 1, the CLK_T16B*n* supply is not suspended and T16B Ch.*n* will keep operating in DEBUG mode.

15.3.4 Event Counter Clock

When EXCL*nm* is selected as the clock source using the T16BnCLK.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.

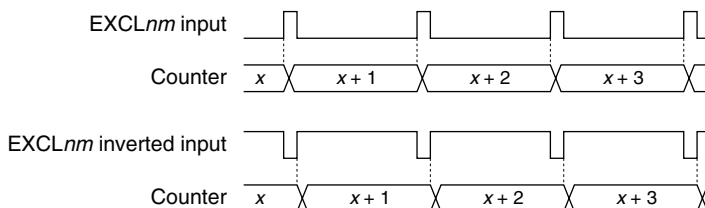


Figure 15.3.4.1 Count Timing (During Count Up Operation)

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.

15.4 Operations

15.4.1 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 T16BnCTL.MODEN bit to 1. (Enable T16B operations)
3. Set the following T16BnCCCTL0 and T16BnCCCTL1 register bits:
 - Set the T16BnCCCTLm.CCMD bit to 0. * (Set comparator mode)
 - T16BnCCCTLm.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.

 - T16BnCCCTLm.TOUTMT bit (Select waveform generation signal)
 - T16BnCCCTLm.TOUTMD[2:0] bits (Select TOUT signal generation mode)
 - T16BnCCCTLm.TOUTINV bit (Select TOUT signal polarity)
4. Set the T16BnMC register. (Set MAX counter data)
5. Set the T16BnCCR0 and T16BnCCR1 registers. (Set the counter comparison value)
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. 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. Set the T16BnMC register. (Set MAX counter data)
5. 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)
6. 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)

15.4.2 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 T16B n CTL.CNTMD[1:0] bits allow selection of up, down, and up/down mode. The T16B n CTL.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 (T16B n MC.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 T16B n CTL.MAXBSY bit is set to 0.
2. Write the MAX value to the T16B n MC.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 T16B n CTL.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 T16B n CTL.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 T16B n TC.TC[15:0] bits. However, since T16B operates on CLK_T16B 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.

Counter status check

The counter operating status can be checked using the T16B n CS.BSY bit. The T16B n CS.BSY bit is set to 1 while the counter is running or 0 while the counter is idle.

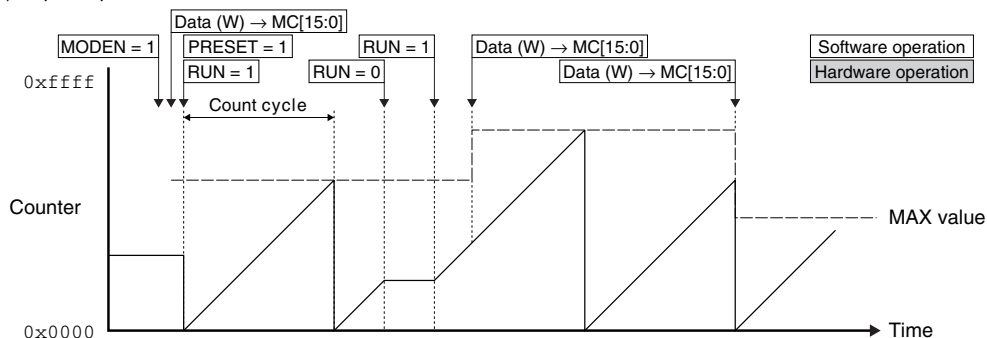
The current count direction can also be checked using the T16B n CS.UP_DOWN bit. The T16B n CS.UP_DOWN bit is set to 1 during count up operation or 0 during count down operation.

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 T16B n CTL.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



(2) One-shot up count mode

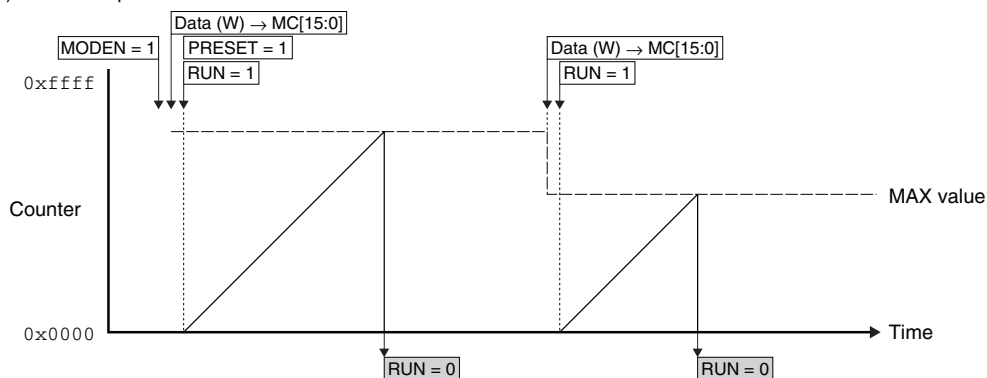


Figure 15.4.2.1 Operations in Repeat Up Count and One-shot Up Count Modes

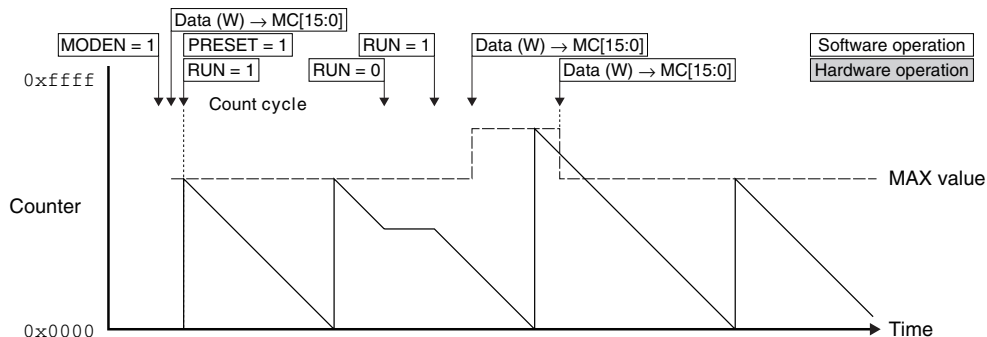
Operations in repeat down count and one-shot down count modes

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

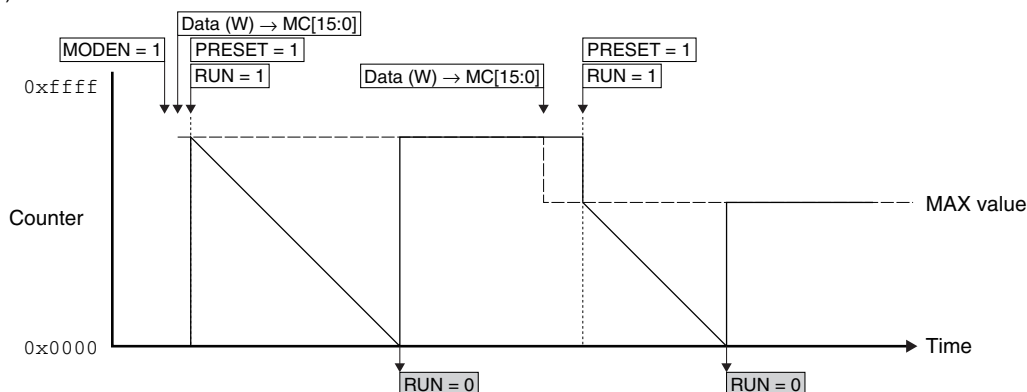


Figure 15.4.2.2 Operations in Repeat Down Count and One-shot Down Count Modes

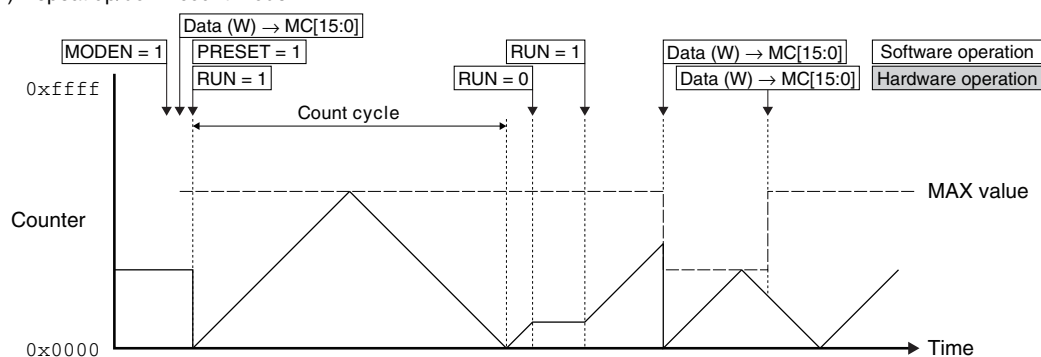
Operations in repeat up/down count and one-shot up/down count modes

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.

(1) Repeat up/down count mode



(2) One-shot up/down count mode

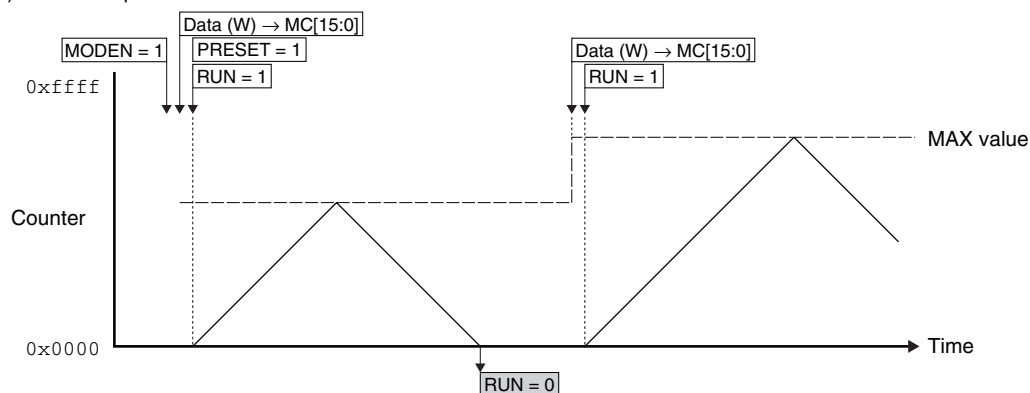


Figure 15.4.2.3 Operations in Repeat Up/Down Count and One-shot Up/Down Count Modes

15.4.3 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.

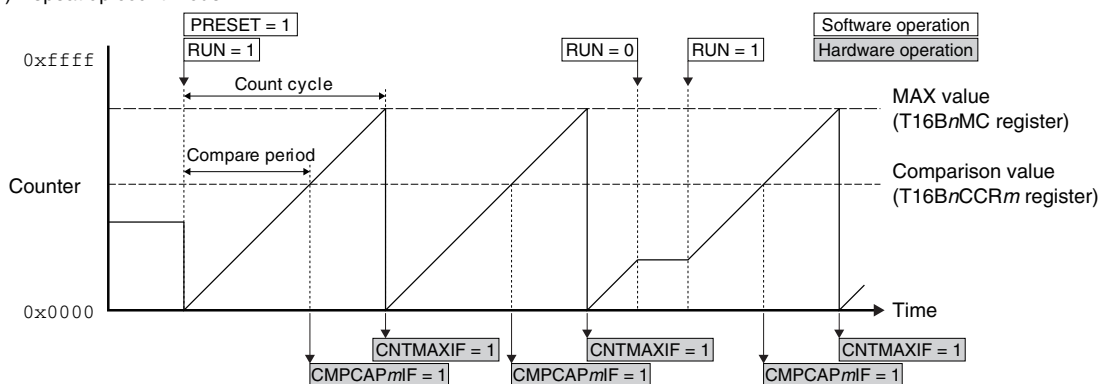
Operations in comparator mode

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.

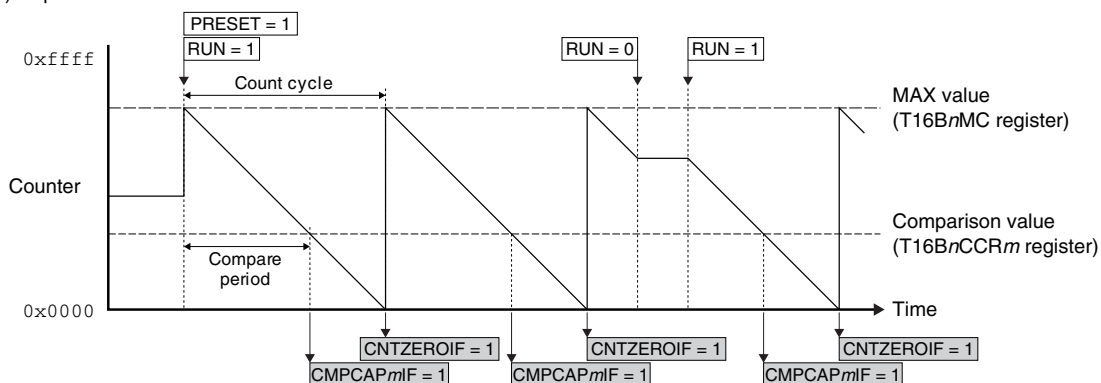
When the counter reaches the value set in the T16BnCCRm register during counting, the comparator asserts the MATCH signal and sets the T16BnINTF.COMPCAPmIF bit (compare interrupt flag) to 1.

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.

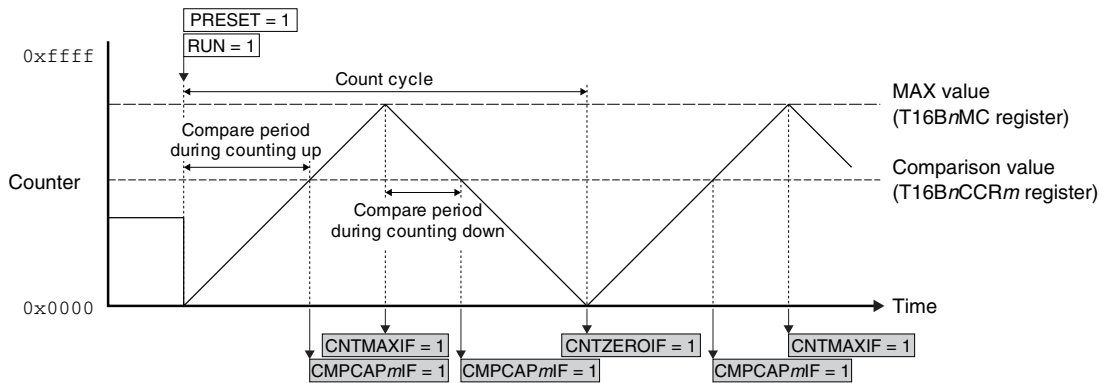
(1) Repeat up count mode



(2) Repeat down count mode



(3) Repeat up/down count mode



(Note that the T16BnINTF.CMPCAPmIF/CNTMAXIF/CNTZEROIF bit clearing operations via software are omitted from the figure.)

Figure 15.4.3.1 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

$$\text{Compare period} = \frac{(\text{CC} + 1)}{f_{\text{CLK T16B}}} [\text{s}] \quad \text{Count cycle} = \frac{(\text{MAX} + 1)}{f_{\text{CLK T16B}}} [\text{s}] \quad (\text{Eq. 15.1})$$

During counting down

$$\text{Compare period} = \frac{(\text{MAX} - \text{CC} + 1)}{f_{\text{CLK}} T_{16\text{B}}} [\text{s}] \quad \text{Count cycle} = \frac{(\text{MAX} + 1)}{f_{\text{CLK}} T_{16\text{B}}} [\text{s}] \quad (\text{Eq. 15.2})$$

Where

CC: T16BnCCR m register setting value (0 to 65,535)

MAX: T16BnMC register setting value (0 to 65,535)

fCLK_T16B: Count clock frequency [Hz]

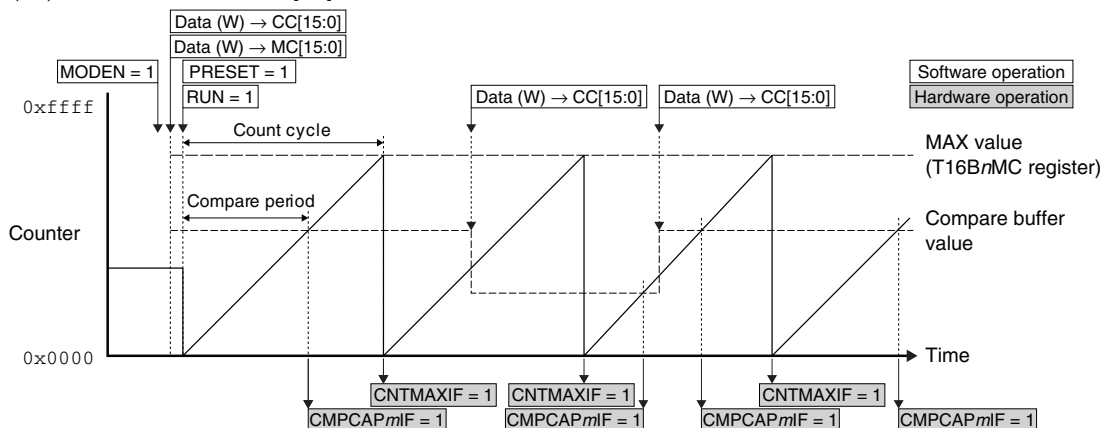
The comparator MATCH signal and counter MAX/ZERO signals are also used to generate a timer output waveform (TOUT). Refer to “TOUT Output Control” for more information.

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.

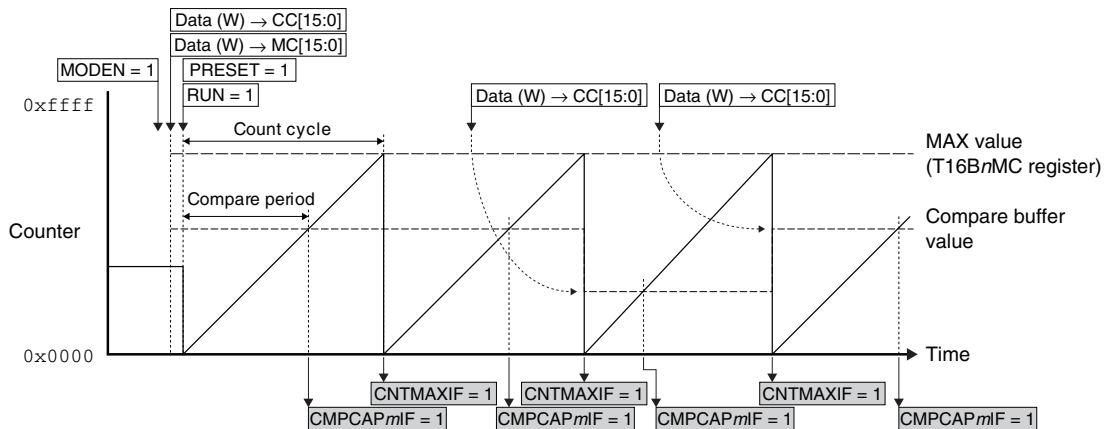
(1) Repeat up count mode

(1.1) T16BnCCCTLm.CBUFMD[2:0] bits = 0x0

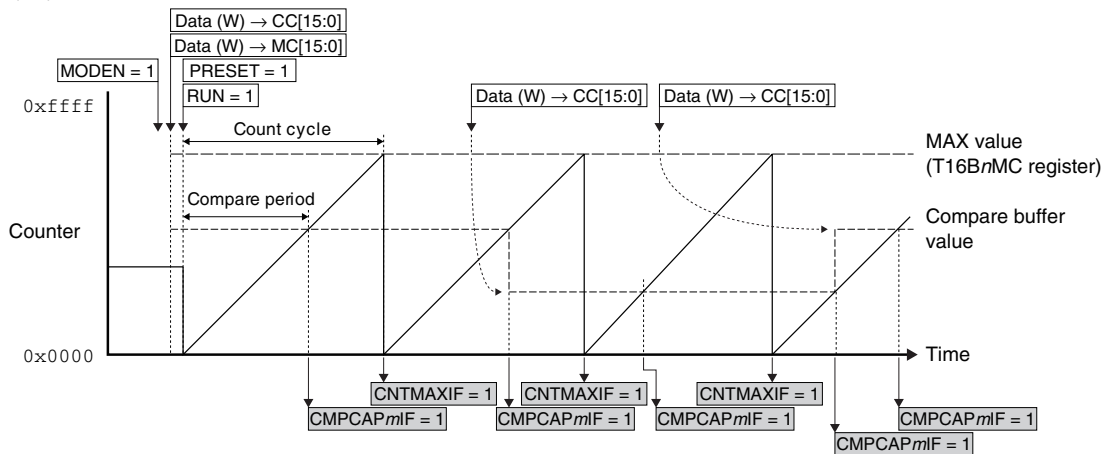


15 16-BIT PWM TIMERS (T16B)

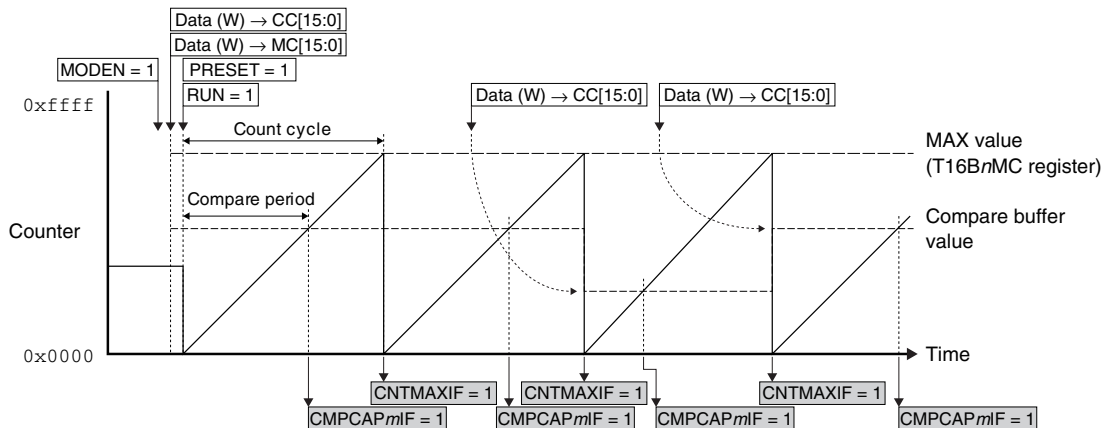
(1.2) T16BnCCCTLm.CBUFMD[2:0] bits = 0x1



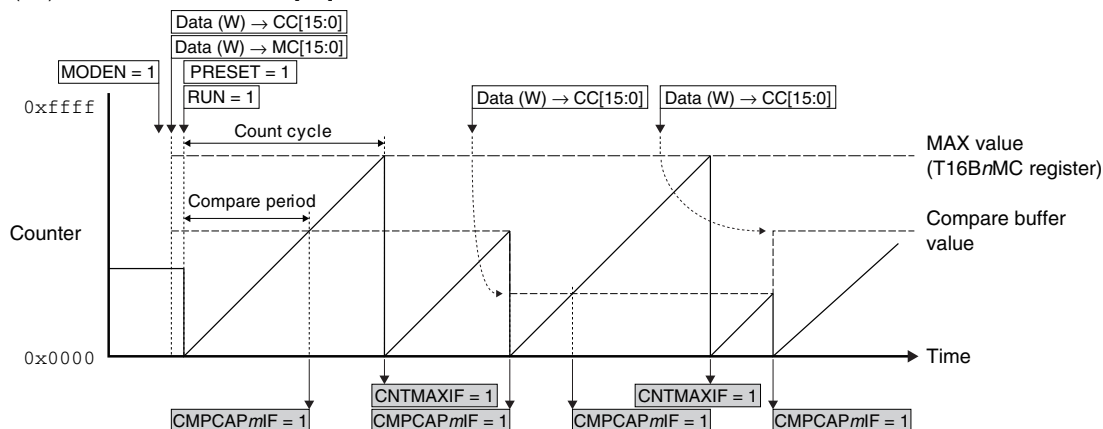
(1.3) T16BnCCCTLm.CBUFMD[2:0] bits = 0x2



(1.4) T16BnCCCTLm.CBUFMD[2:0] bits = 0x3

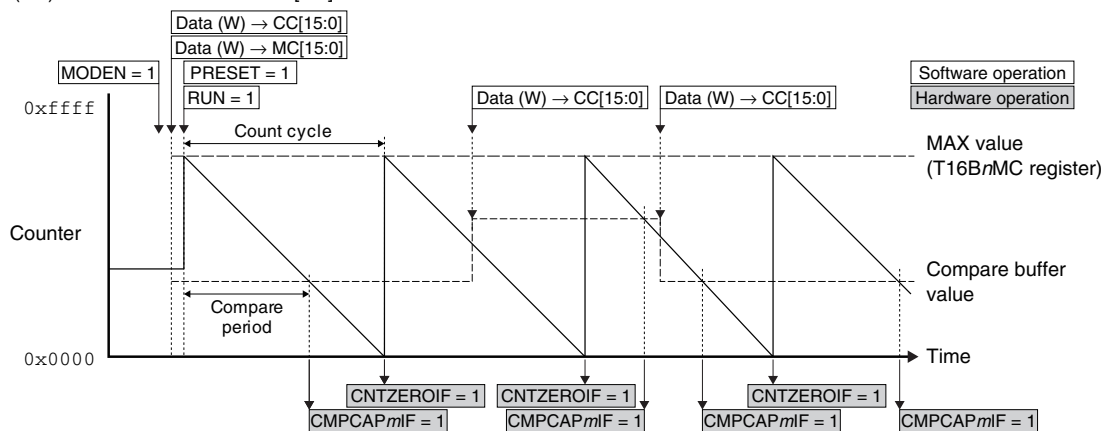


(1.5) T16BnCCCTLm.CBUFMD[2:0] bits = 0x4

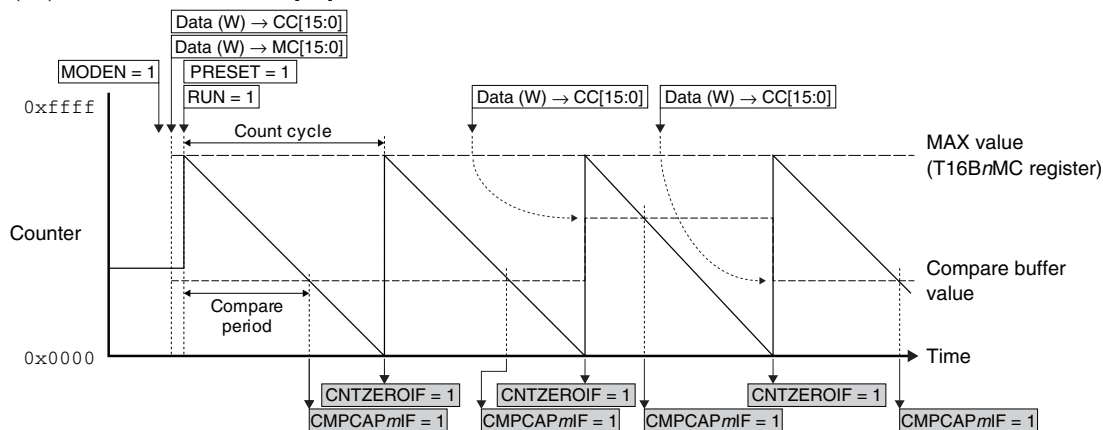


(2) Repeat down count mode

(2.1) T16BnCCCTLm.CBUFMD[2:0] bits = 0x0

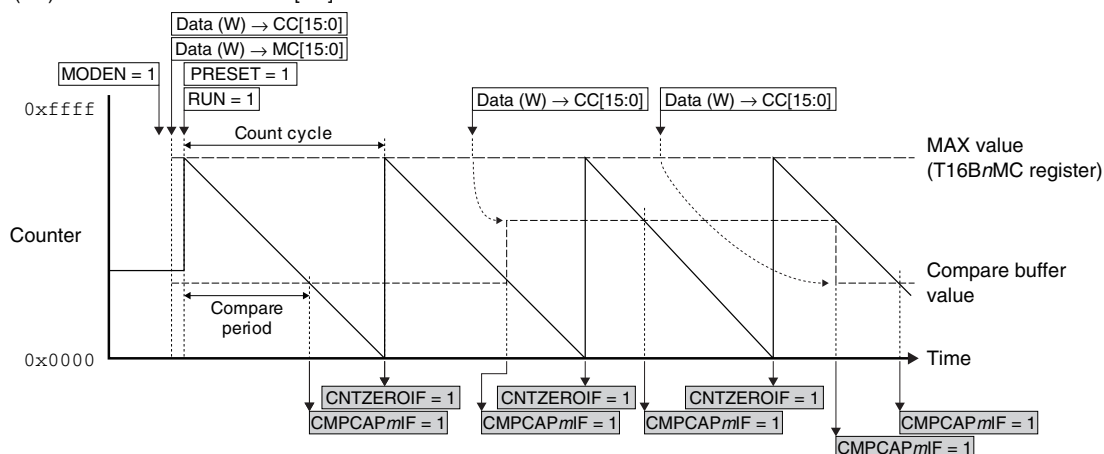


(2.2) T16BnCCCTLm.CBUFMD[2:0] bits = 0x1

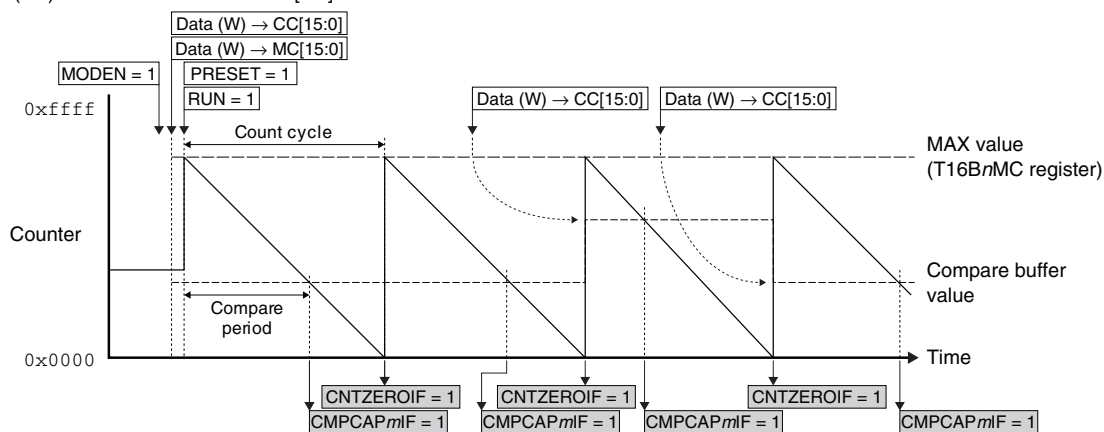


15 16-BIT PWM TIMERS (T16B)

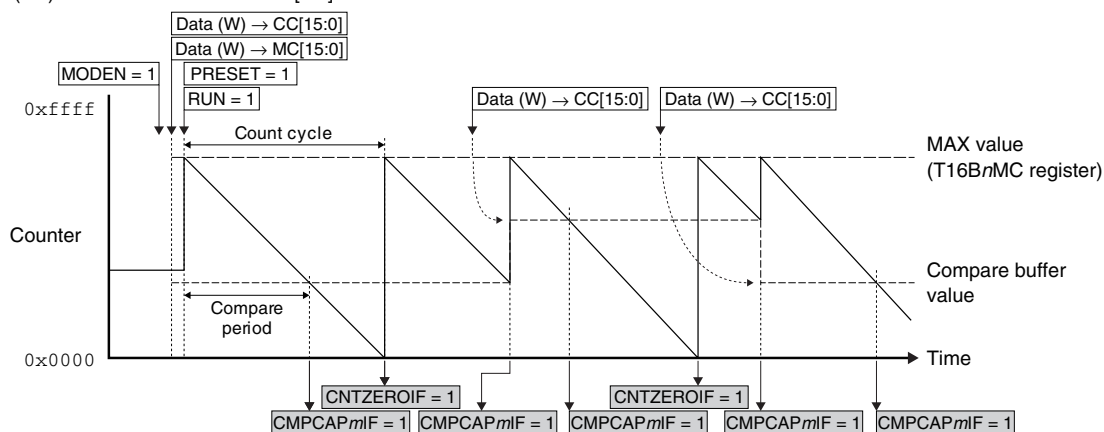
(2.3) T16BnCCCTLm.CBUFMD[2:0] bits = 0x2



(2.4) T16BnCCCTLm.CBUFMD[2:0] bits = 0x3

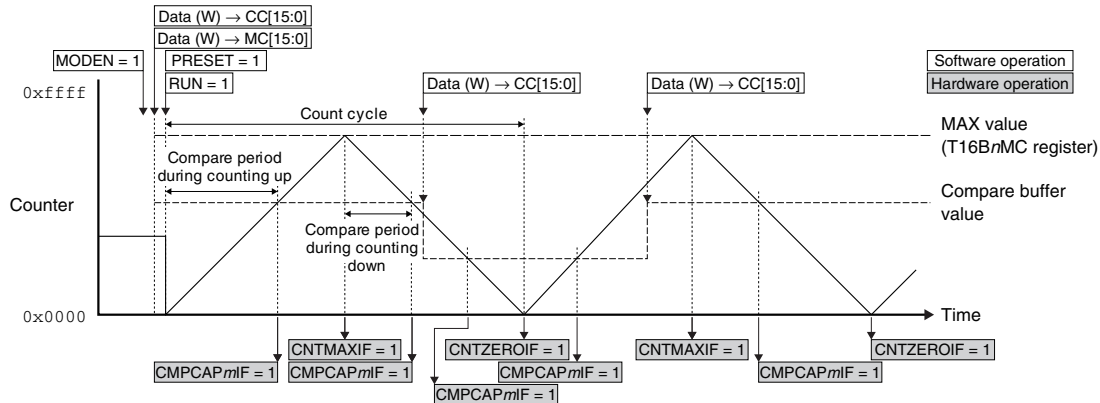


(2.5) T16BnCCCTLm.CBUFMD[2:0] bits = 0x4

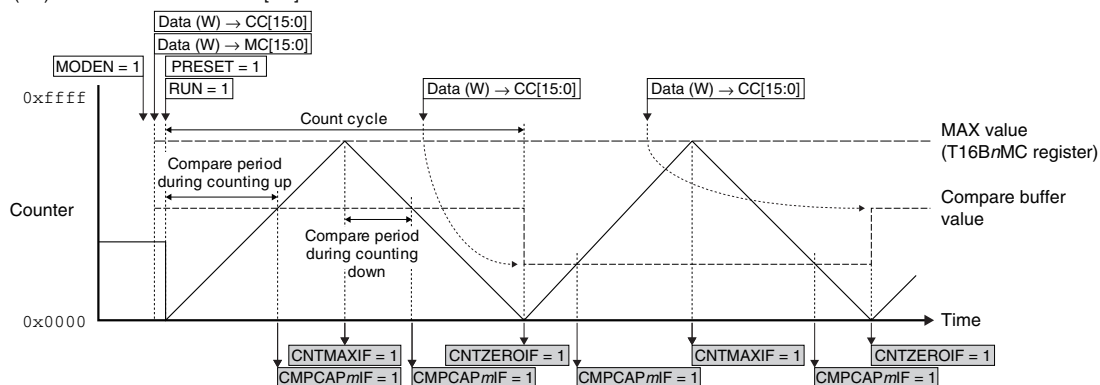


(3) Repeat up/down count mode

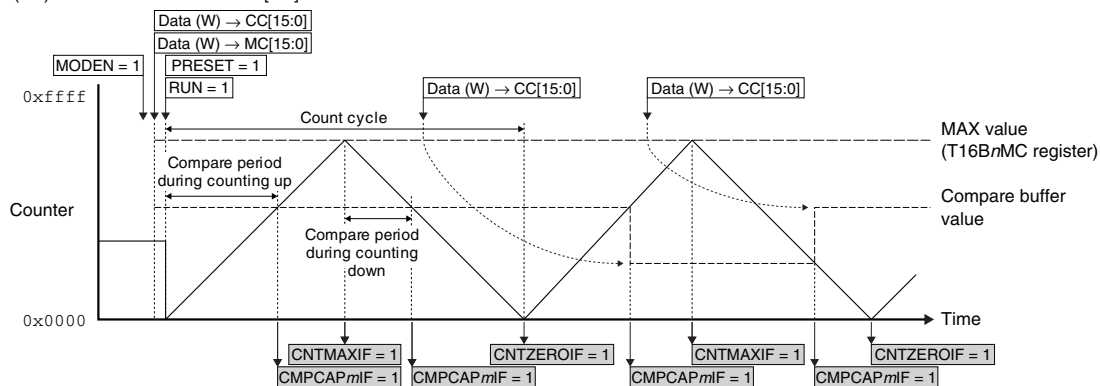
(3.1) T16BnCCCTLm.CBUFMD[2:0] bits = 0x0



(3.2) T16BnCCCTLm.CBUFMD[2:0] bits = 0x1



(3.3) T16BnCCCTLm.CBUFMD[2:0] bits = 0x2



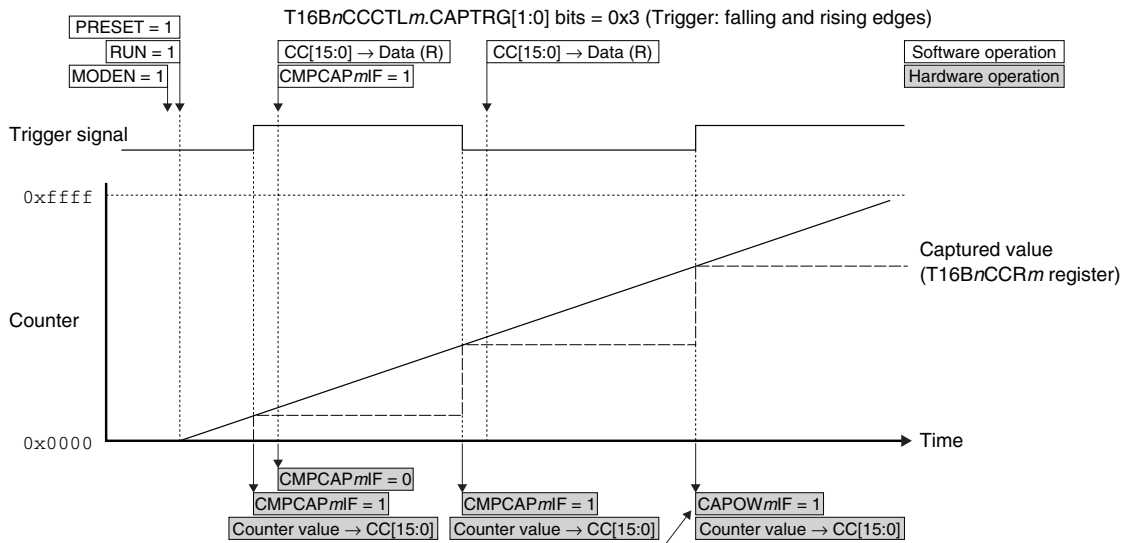


Figure 15.4.3.3 Operations in Capture Mode (Example in One-shot Up Count Mode)

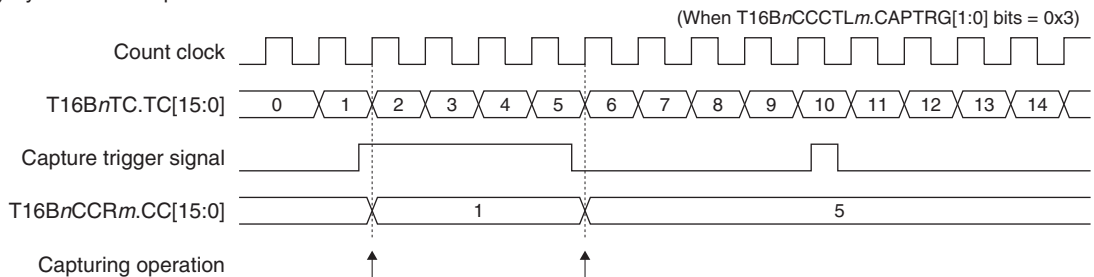
Synchronous capture mode/asynchronous capture mode

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

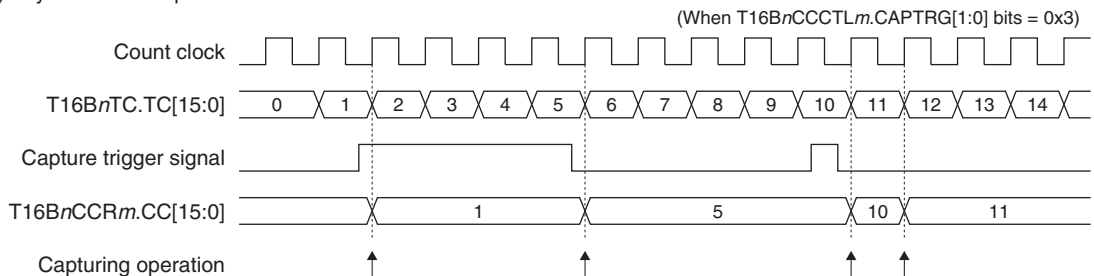


Figure 15.4.3.4 Synchronous Capture Mode/Asynchronous Capture Mode

15.4.4 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 15.4.4.1 shows the TOUT output circuits (circuits 0 and 1).

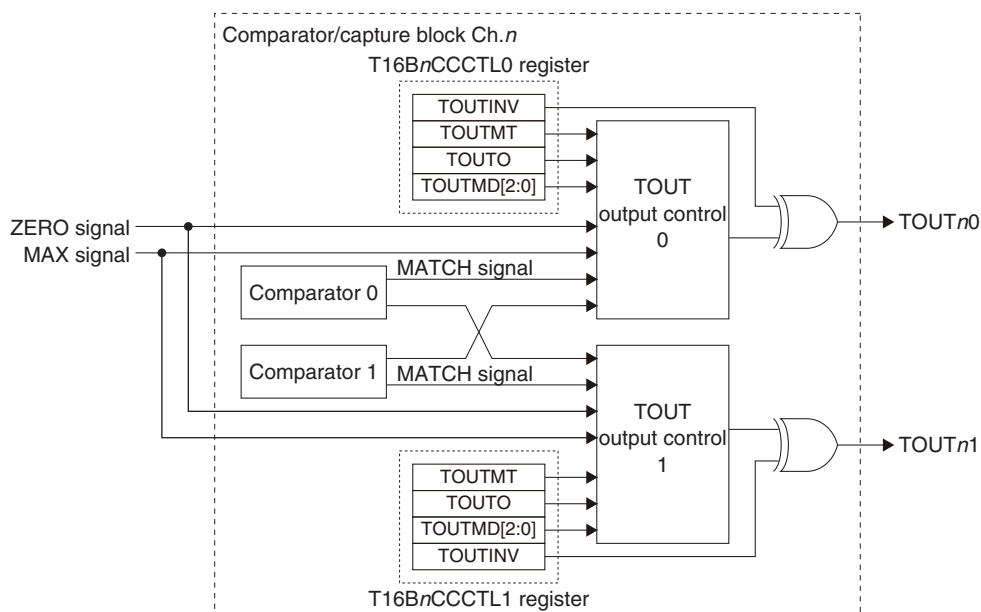


Figure 15.4.4.1 TOUT Output Circuits (Circuits 0 and 1)

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.

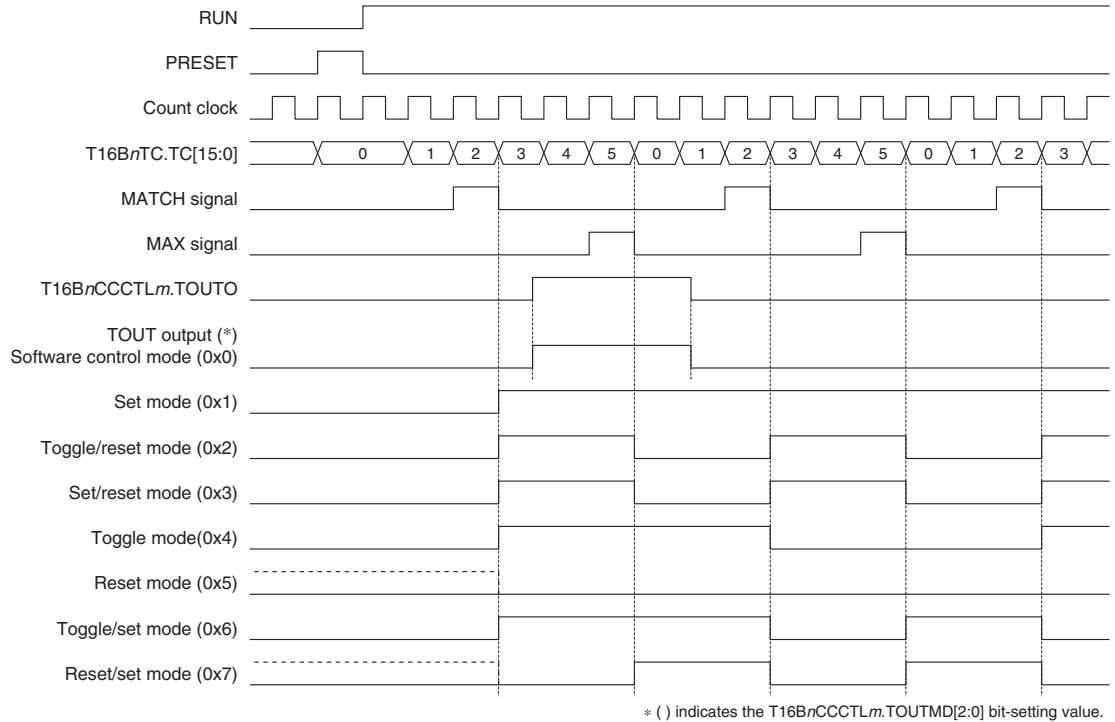
TOUT signal polarity

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.

Figures 15.4.4.2 and 15.4.4.3 show the TOUT output waveforms.

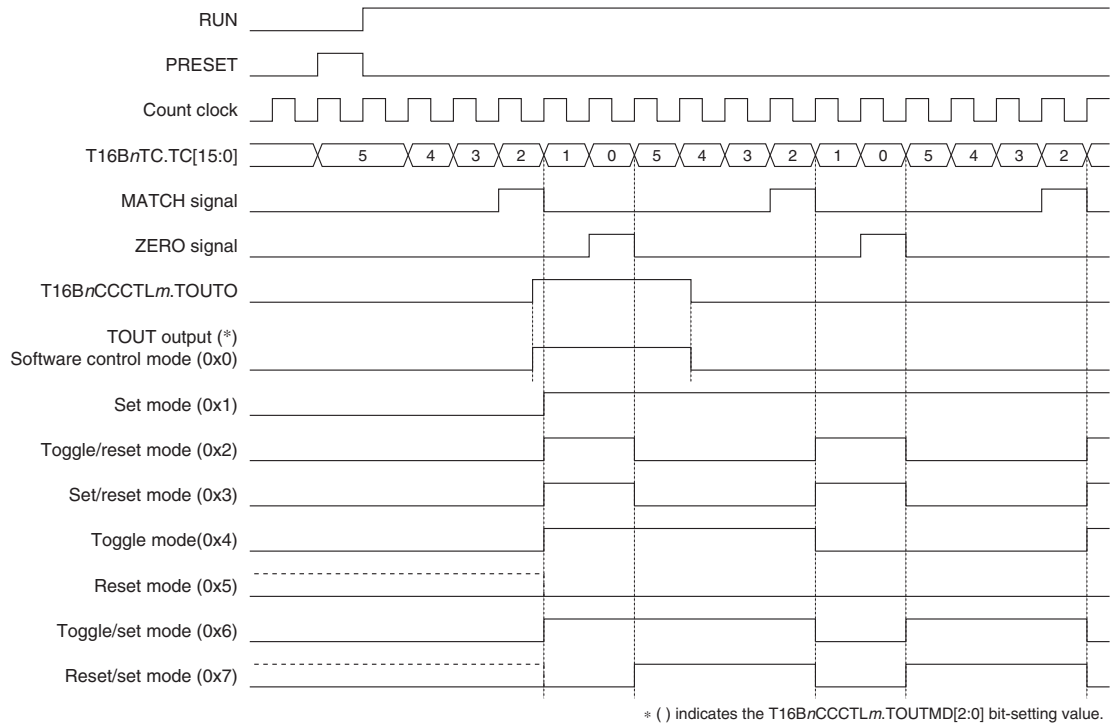
(1) Repeat up count mode

(MAX value = 5, Compare buffer value = 2, T16BnCCCTLm.TOUTINV bit = 0)



(2) Repeat down count mode

(MAX value = 5, Compare buffer value = 2, T16BnCCCTLm.TOUTINV bit = 0)



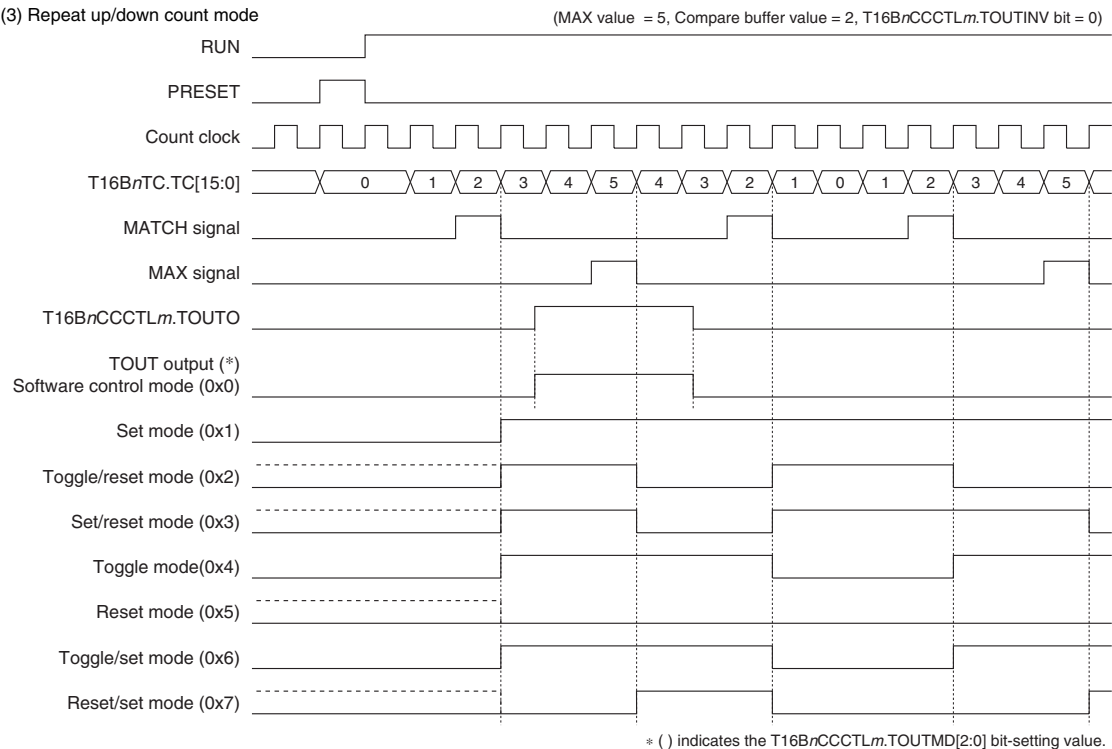
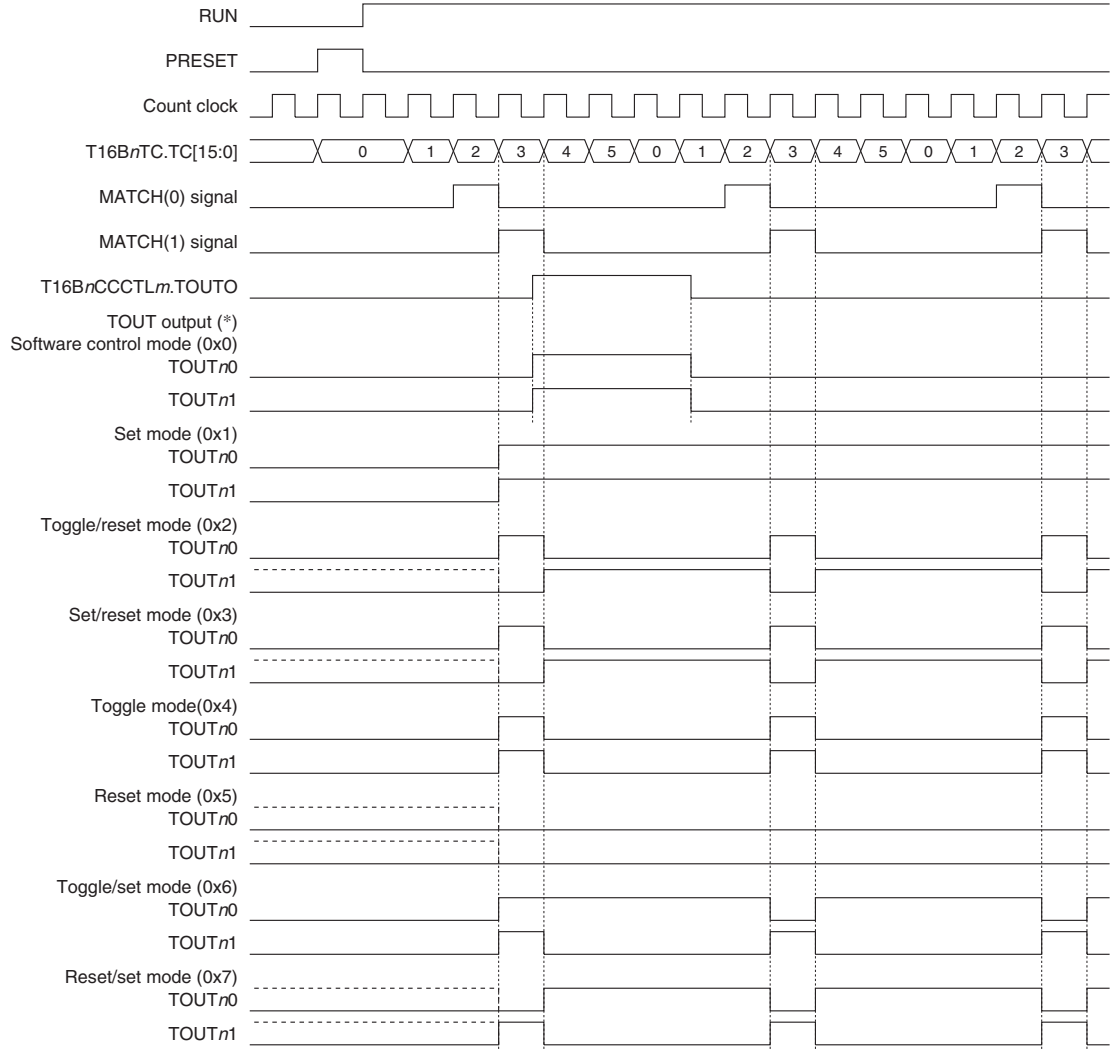


Figure 15.4.4.2 TOUT Output Waveform (T16BnCCCTLm.TOUTMT bit = 0)

(1) Repeat up count mode

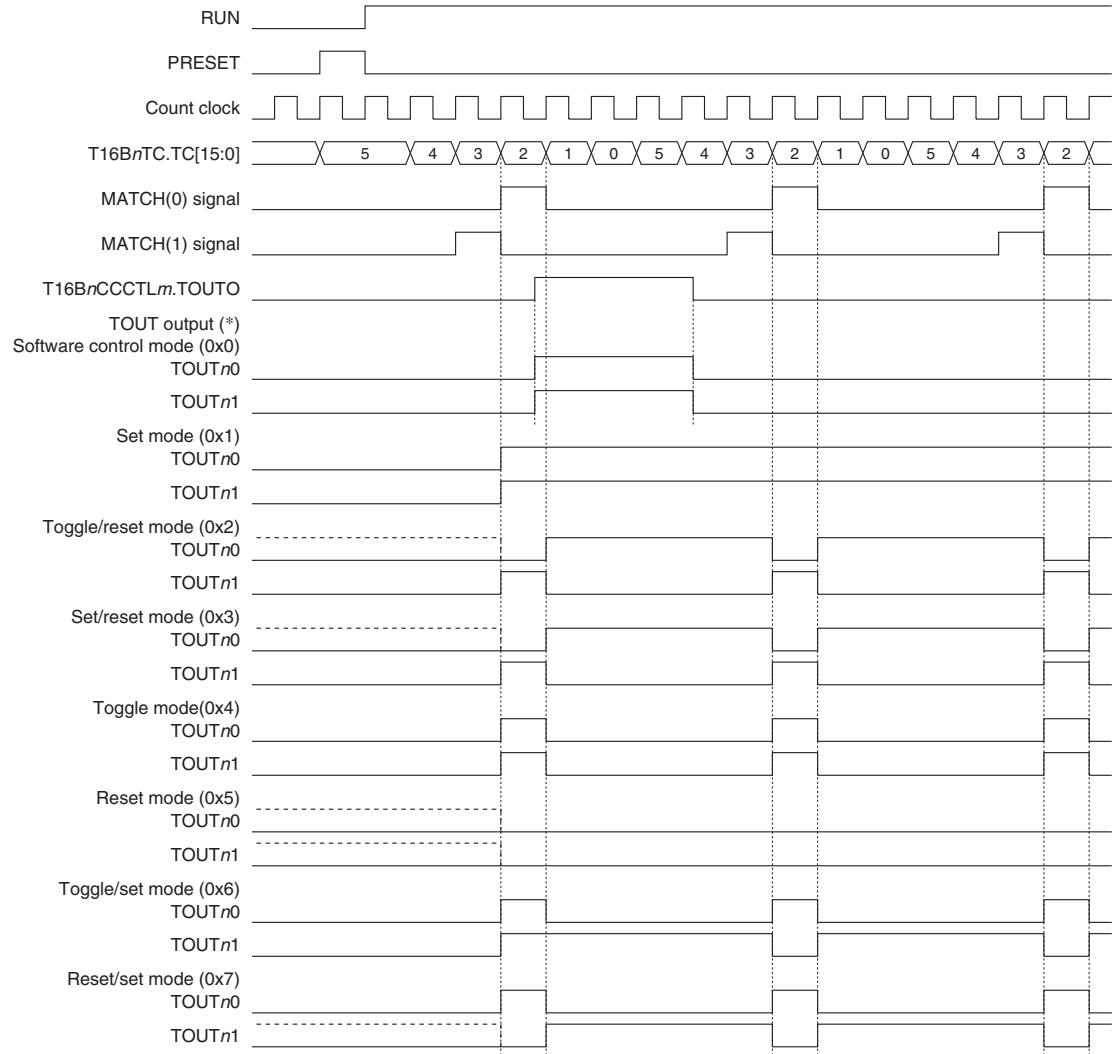
(MAX value = 5, Compare buffer (0) value = 2, Compare buffer (1) value = 3, T16BnCCCTLm.TOUTINV bit = 0)



* () indicates the T16BnCCCTLm.TOUTMD[2:0] bit-setting value.

15 16-BIT PWM TIMERS (T16B)

(2) Repeat down count mode (MAX value = 5, Compare buffer (0) value = 2, Compare buffer (1) value = 3, T16BnCCCTLm.TOUTINV bit = 0)



* () indicates the T16BnCCCTLm.TOUTMD[2:0] bit-setting value.

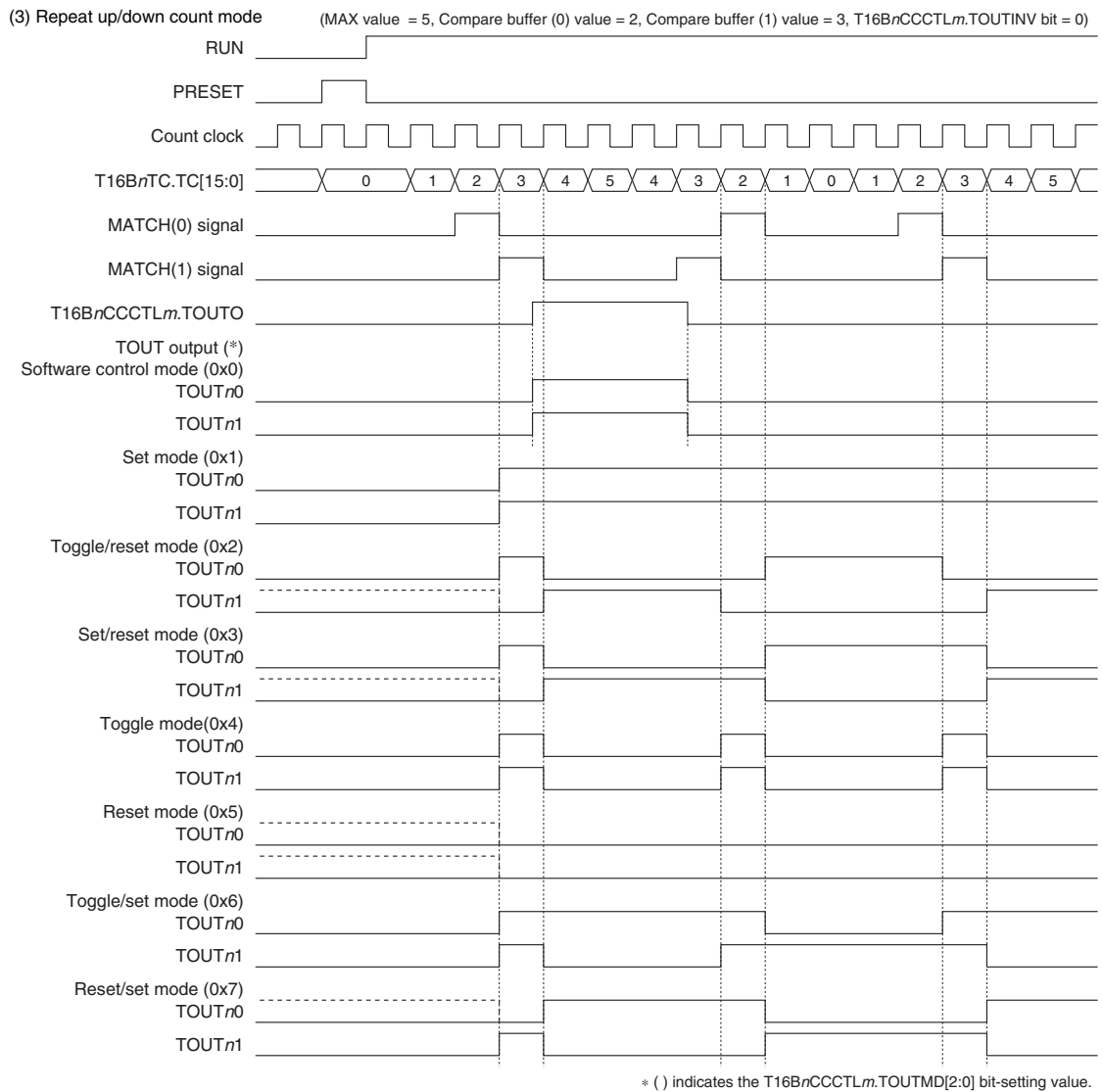


Figure 15.4.4.3 TOUT Output Waveform (T16BnCCCTL0.TOUTMT bit = 1, T16BnCCCTL1.TOUTMT bit = 0)

15.5 Interrupt

Each T16B channel has a function to generate the interrupt shown in Table 15.5.1.

Table 15.5.1 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.

15.6 Control Registers

T16B Ch.n Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16BnCLK	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 15.6.1 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					
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.

T16B Ch. 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 15.6.2).

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 15.6.2).

Table 15.6.2 Count Mode

T16BnCTL.CNTMD[1:0] bits	Count mode	
	T16BnCTL.ONEST bit = 1	T16BnCTL.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 T16BnCTL.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 T16BnCTL.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 T16BnMC register, is preset to the counter. However, the T16BnCTL.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 T16BnCTL.PRESET bit and the counting start operation using the T16BnCTL.RUN bit take effect only when the T16BnCTL.MODEN bit = 1.

T16B Ch.n Max Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16BnMC	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 T16BnMC.MC[15:0] bits (MAX value) during counting.
 - Make sure the T16BnCTL.MODEN bit is set to 1 before writing data to the T16BnMC.MC[15:0] bits. If the T16BnCTL.MODEN bit = 0 when writing to the T16BnMC.MC[15:0] bits, set the T16BnCTL.MODEN bit to 1 until the T16BnCS.BSY bit is set to 0 from 1.
 - Do not set the T16BnMC.MC[15:0] bits to 0x0000.

T16B Ch.n Timer Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16BnTC	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

The following shows the correspondence between the bit and the CAP*nm* pin:

T16B*n*CS.CAP15 bit: CAP*n*5 pin

T16B*n*CS.CAP14 bit: CAP*n*4 pin

T16B*n*CS.CAP13 bit: CAP*n*3 pin

T16B*n*CS.CAP12 bit: CAP*n*2 pin

T16B*n*CS.CAP11 bit: CAP*n*1 pin

T16B*n*CS.CAP10 bit: CAP*n*0 pin

Note: The configuration of the T16B*n*CS.CAP1*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 <i>n</i> 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 <i>n</i> 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

The following shows the correspondence between the bit and interrupt:

T16B*n*INTE.CAPOW5IE bit: Capture 5 overwrite interrupt

T16B*n*INTE.CMPCAP5IE bit: Compare/capture 5 interrupt

T16B*n*INTE.CAPOW4IE bit: Capture 4 overwrite interrupt

T16B*n*INTE.CMPCAP4IE bit: Compare/capture 4 interrupt

T16B*n*INTE.CAPOW3IE bit: Capture 3 overwrite interrupt

T16B*n*INTE.CMPCAP3IE bit: Compare/capture 3 interrupt

T16B*n*INTE.CAPOW2IE bit: Capture 2 overwrite interrupt

T16B*n*INTE.CMPCAP2IE bit: Compare/capture 2 interrupt

T16B*n*INTE.CAPOW1IE bit: Capture 1 overwrite interrupt

T16B*n*INTE.CMPCAP1IE bit: Compare/capture 1 interrupt

T16B*n*INTE.CAPOW0IE bit: Capture 0 overwrite interrupt

T16B*n*INTE.CMPCAP0IE bit: Compare/capture 0 interrupt

T16B*n*INTE.CNTMAXIE bit: Counter MAX interrupt

T16B*n*INTE.CNTZEROIE bit: Counter zero interrupt

Notes:

- The configuration of the T16B*n*INTE.CAPOW*m*IE and T16B*n*INTE.CMPCAP*m*IE 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
T16B _n CCCTL _m	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 T16B_nCCCTL_m.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 T16B_nCCR_m register to the compare buffer. The T16B_nCCCTL_m.CBUFMD[2:0] bits are control bits for comparator mode and are ineffective in capture mode.

Table 15.6.3 Timings to Load Comparison Value to Compare Buffer

T16B _n CCCTL _m . 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_T16B _n rising edge after writing to the T16B _n CCR _m register
	Down mode	
	Up/down mode	

Bits 11–10 CAPIS[1:0]

These bits select the trigger signal for capturing (see Table 15.6.4). The T16B_nCCCTL_m.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 T16B_nCCR_m register in capture mode (see Table 15.6.4). The T16B_nCCCTL_m.CAPTRG[1:0] bits are control bits for capture mode and are ineffective in comparator mode.

Table 15.6.4 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.

Bits 4–2 **TOUTMD[2:0]**

These bits configure how the TOUTnm signal waveform is changed by the comparator MATCH and counter MAX/ZERO signals.

The T16BnCCCTLm.TOUTMD[2:0] bits are control bits for comparator mode and are ineffective in capture mode.

Table 15.6.5 TOUT Generation Mode

T16BnCCCTLm. TOUTMD[2:0] bits	TOUT generation mode and operations			
	T16BnCCCTLm. TOUTMT bit	Count mode	Output signal	Change in the signal
0x7	Reset/set mode			
	0	Up count mode	TOUTnm	The signal becomes inactive by the MATCH signal and it becomes active by the MAX signal.
		Up/down count mode	TOUTnm	The signal becomes inactive by the MATCH signal and it becomes active by the ZERO signal.
	1	All count modes	TOUTnm	The signal becomes inactive by the MATCHm signal and it becomes active by the MATCHm+1 signal.
			TOUTnm+1	The signal becomes inactive by the MATCHm+1 signal and it becomes active by the MATCHm signal.
0x6	Toggle/set mode			
	0	Up count mode	TOUTnm	The signal is inverted by the MATCH signal and it becomes active by the MAX signal.
		Up/down count mode	TOUTnm	The signal is inverted by the MATCH signal and it becomes active by the ZERO signal.
	1	All count modes	TOUTnm	The signal is inverted by the MATCHm signal and it becomes active by the MATCHm+1 signal.
			TOUTnm+1	The signal is inverted by the MATCHm+1 signal and it becomes active by the MATCHm signal.
0x5	Reset mode			
	0	All count modes	TOUTnm	The signal becomes inactive by the MATCH signal.
	1	All count modes	TOUTnm	The signal becomes inactive by the MATCHm or MATCHm+1 signal.
TOUTnm+1			The signal becomes inactive by the MATCHm+1 or MATCHm signal.	

15 16-BIT PWM TIMERS (T16B)

T16BnCCCTLm. TOUTMD[2:0] bits	TOUT generation mode and operations			
	T16BnCCCTLm. TOUTMT bit	Count mode	Output signal	Change in the signal
0x4	Toggle mode			
	0	All count modes	TOUTnm	The signal is inverted by the MATCH signal.
	1	All count modes	TOUTnm	The signal is inverted by the MATCHm or MATCHm+1 signal.
			TOUTnm+1	The signal is inverted by the MATCHm+1 or MATCHm signal.
0x3	Set/reset mode			
	0	Up count mode	TOUTnm	The signal becomes active by the MATCH signal and it becomes inactive by the MAX signal.
		Down count mode	TOUTnm	The signal becomes active by the MATCH signal and it becomes inactive by the ZERO signal.
	1	All count modes	TOUTnm	The signal becomes active by the MATCHm signal and it becomes inactive by the MATCHm+1 signal.
			TOUTnm+1	The signal becomes active by the MATCHm+1 signal and it becomes inactive by the MATCHm signal.
0x2	Toggle/reset mode			
	0	Up count mode	TOUTnm	The signal is inverted by the MATCH signal and it becomes inactive by the MAX signal.
		Down count mode	TOUTnm	The signal is inverted by the MATCH signal and it becomes inactive by the ZERO signal.
	1	All count modes	TOUTnm	The signal is inverted by the MATCHm signal and it becomes inactive by the MATCHm+1 signal.
			TOUTnm+1	The signal is inverted by the MATCHm+1 signal and it becomes inactive by the MATCHm signal.
0x1	Set mode			
	0	All count modes	TOUTnm	The signal becomes active by the MATCH signal.
	1	All count modes	TOUTnm	The signal becomes active by the MATCHm or MATCHm+1 signal.
			TOUTnm+1	The signal becomes active by the MATCHm+1 or MATCHm signal.
0x0	Software control mode			
	*	All count modes	TOUTnm	The signal becomes active by setting the T16BnCCCTLm.TOUTO bit to 1 and it becomes inactive by setting to 0.

Bit 1 TOUTINV

This bit selects the TOUTnm signal polarity.

1 (R/W): Inverted (active low)

0 (R/W): Normal (active high)

The T16BnCCCTLm.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 (T16BnCCRm register = capture register)

0 (R/W): Comparator mode (T16BnCCRm register = compare data register)

T16B Ch.n Compare/Capture m Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16BnCCRm	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.

16 Sound Generator (SNDA)

16.1 Overview

SNDA is a sound generator that generates melodies and buzzer signals. The features of the SNDA 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 and slur 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 16.1.1 shows the SNDA configuration.

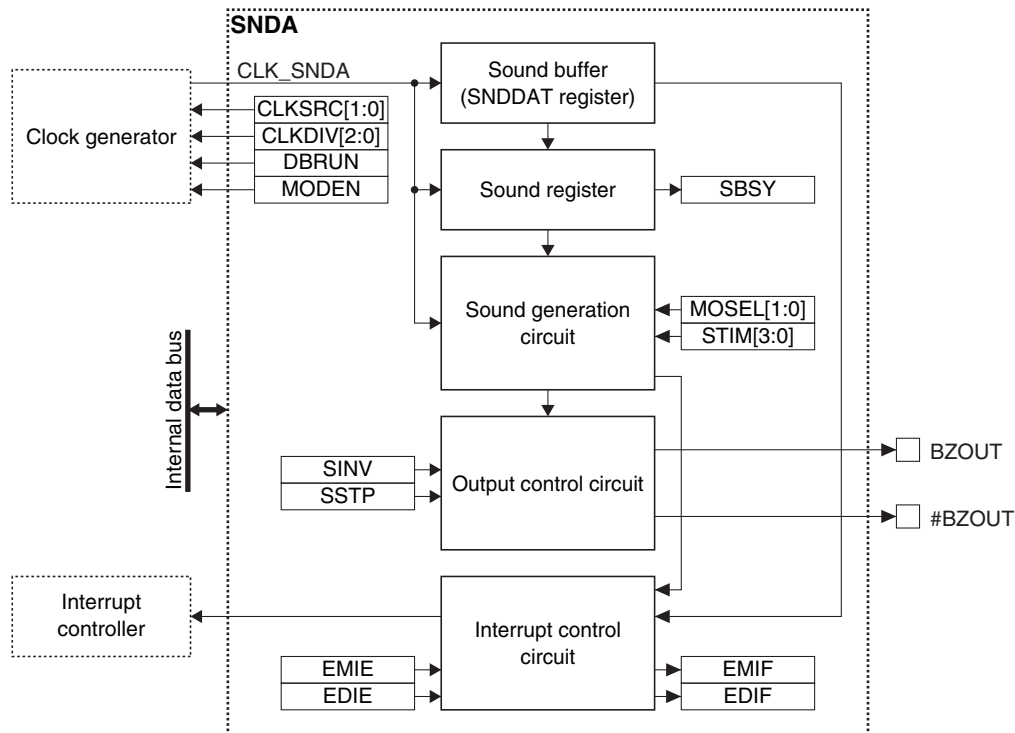


Figure 16.1.1 SNDA Configuration

16.2 Output Pins and External Connections

16.2.1 List of Output Pins

Table 16.2.1.1 lists the SNDA pins.

Table 16.2.1.1 List of SNDA 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 SNDA

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

16.2.2 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.

16.2.3 External Connections

Figures 16.2.2.1 and 16.2.2.2 show connection diagrams between SNDA and a piezoelectric buzzer.

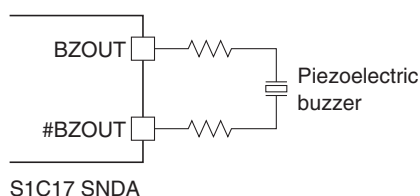


Figure 16.2.2.1 Connection between SNDA and Piezoelectric Buzzer (Direct Drive)

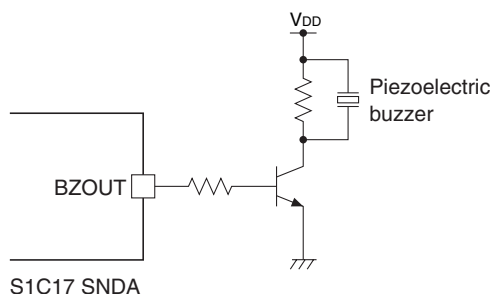


Figure 16.2.2.2 Connection between SNDA and Piezoelectric Buzzer (Single Pin Drive)

16.3 Clock Settings

16.3.1 SNDA Operating Clock

When using SNDA, the SNDA operating clock CLK_SNDA must be supplied to SNDA from the clock generator. The CLK_SNDA 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_SNDA frequency should be set to around 32,768 Hz.

16.3.2 Clock Supply in SLEEP Mode

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

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

16.3.3 Clock Supply in DEBUG Mode

The CLK_SNDA supply during DEBUG mode should be controlled using the SNDCLK.DBRUN bit.

The CLK_SNDA supply to SNDA is suspended when the CPU enters DEBUG mode if the SNDCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_SNDA supply resumes. Although SNDA stops operating when the CLK_SNDA supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the SNDCLK.DBRUN bit = 1, the CLK_SNDA supply is not suspended and SNDA will keep operating in DEBUG mode.

16.4 Operations

16.4.1 Initialization

SNDA should be initialized with the procedure shown below.

1. Assign the SNDA output function to the ports. (Refer to the “I/O Ports” chapter.)
2. Configure the SNDA operating clock.
3. Set the SNDCTL.MODEN bit to 1. (Enable SNDA operations)
4. Set the SNDSEL.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)

16.4.2 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 SNDA operations are shown below.

Normal buzzer output start/stop procedure

1. Set the SNDSEL.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)

Normal buzzer output operations

When data is written to the sound buffer (SNDDAT register), SNDA 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_SNDA 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 16.4.2.1 shows a buzzer output timing chart in normal buzzer mode.

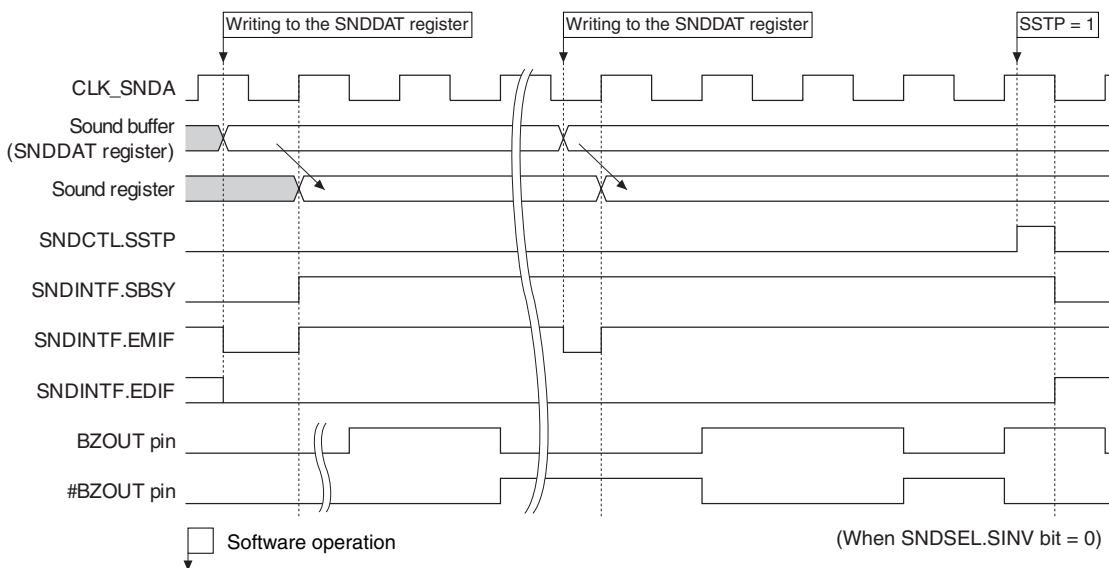


Figure 16.4.2.1 Buzzer Output Timing Chart in Normal Buzzer Mode

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.

$$\text{SNDDAT.SFRQ}[7:0] \text{ bits} = \frac{f_{\text{CLK_SNDA}}}{f_{\text{BZOUT}}} - 1 \quad (\text{Eq. 16.1})$$

$$\text{SNDDAT.SLEN}[5:0] \text{ bits} = \left(\frac{f_{\text{CLK_SNDA}}}{f_{\text{BZOUT}}} \times \frac{\text{DUTY}}{100} \right) - 1 \quad (\text{Eq. 16.2})$$

Where

$f_{\text{CLK_SNDA}}$: CLK_SNDA 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 16.4.2.1 Buzzer Frequency Settings (when fCLK_SND A = 32,768 Hz)

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	512.0	0x2f	682.7	0x1f	1,024.0	0x0f	2,048.0
0x3e	520.1	0x2e	697.2	0x1e	1,057.0	0x0e	2,184.5
0x3d	528.5	0x2d	712.3	0x1d	1,092.3	0x0d	2,340.6
0x3c	537.2	0x2c	728.2	0x1c	1,129.9	0x0c	2,520.6
0x3b	546.1	0x2b	744.7	0x1b	1,170.3	0x0b	2,730.7
0x3a	555.4	0x2a	762.0	0x1a	1,213.6	0x0a	2,978.9
0x39	565.0	0x29	780.2	0x19	1,260.3	0x09	3,276.8
0x38	574.9	0x28	799.2	0x18	1,310.7	0x08	3,640.9
0x37	585.1	0x27	819.2	0x17	1,365.3	0x07	4,096.0
0x36	595.8	0x26	840.2	0x16	1,424.7	0x06	4,681.1
0x35	606.8	0x25	862.3	0x15	1,489.5	0x05	5,461.3
0x34	618.3	0x24	885.6	0x14	1,560.4	0x04	6,553.6
0x33	630.2	0x23	910.2	0x13	1,638.4	0x03	8,192.0
0x32	642.5	0x22	936.2	0x12	1,724.6	0x02	10,922.7
0x31	655.4	0x21	963.8	0x11	1,820.4	0x01	16,384.0
0x30	668.7	0x20	993.0	0x10	1,927.5	0x00	Cannot be set

Table 16.4.2.2 Buzzer Duty Ratio Setting Examples (when fCLK_SND A = 32,768 Hz)

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	—	—	—	—	—	98.4
0x3d	—	—	—	—	—	96.9
0x3c	—	—	—	—	—	95.3
0x3b	—	—	—	—	—	93.8
0x3a	—	—	—	—	—	92.2
0x39	—	—	—	—	—	90.6
0x38	—	—	—	—	—	89.1
0x37	—	—	—	—	—	87.5
0x36	—	—	—	—	—	85.9
0x35	—	—	—	—	—	84.4
0x34	—	—	—	—	—	82.8
0x33	—	—	—	—	—	81.3
0x32	—	—	—	—	—	79.7
0x31	—	—	—	—	—	78.1
0x30	—	—	—	—	—	76.6
0x2f	—	—	—	—	—	75.0
0x2e	—	—	—	—	—	73.4
0x2d	—	—	—	—	—	71.9
0x2c	—	—	—	—	—	70.3
0x2b	—	—	—	—	—	68.8
0x2a	—	—	—	—	—	67.2
0x29	—	—	—	—	—	65.6
0x28	—	—	—	—	—	64.1
0x27	—	—	—	—	—	62.5
0x26	—	—	—	—	—	60.9
0x25	—	—	—	—	—	59.4
0x24	—	—	—	—	—	57.8
0x23	—	—	—	—	—	56.3
0x22	—	—	—	—	—	54.7
0x21	—	—	—	—	—	53.1
0x20	—	—	—	—	—	51.6
0x1f	—	—	—	—	—	50.0
0x1e	—	—	—	—	96.9	48.4
0x1d	—	—	—	—	93.8	46.9
0x1c	—	—	—	—	90.6	45.3
0x1b	—	—	—	—	87.5	43.8
0x1a	—	—	—	—	84.4	42.2
0x19	—	—	—	—	81.3	40.6
0x18	—	—	—	—	78.1	39.1
0x17	—	—	—	—	75.0	37.5
0x16	—	—	—	—	71.9	35.9
0x15	—	—	—	—	68.8	34.4
0x14	—	—	—	—	65.6	32.8
0x13	—	—	—	—	62.5	31.3
0x12	—	—	—	—	59.4	29.7

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
0x11	–	–	–	–	56.3	28.1
0x10	–	–	–	–	53.1	26.6
0x0f	–	–	–	–	50.0	25.0
0x0e	–	–	–	93.8	46.9	23.4
0x0d	–	–	–	87.5	43.8	21.9
0x0c	–	–	–	81.3	40.6	20.3
0x0b	–	–	–	75.0	37.5	18.8
0x0a	–	–	–	68.8	34.4	17.2
0x09	–	–	–	62.5	31.3	15.6
0x08	–	–	–	56.3	28.1	14.1
0x07	–	–	–	50.0	25.0	12.5
0x06	–	–	87.5	43.8	21.9	10.9
0x05	–	–	75.0	37.5	18.8	9.4
0x04	–	–	62.5	31.3	15.6	7.8
0x03	–	–	50.0	25.0	12.5	6.3
0x02	–	75.0	37.5	18.8	9.4	4.7
0x01	–	50.0	25.0	12.5	6.3	3.1
0x00	50.0	25.0	12.5	6.3	3.1	1.6

16.4.3 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 SNDA 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), SNDA 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_SNDA 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 16.4.3.1 shows a buzzer output timing chart in one-shot buzzer mode.

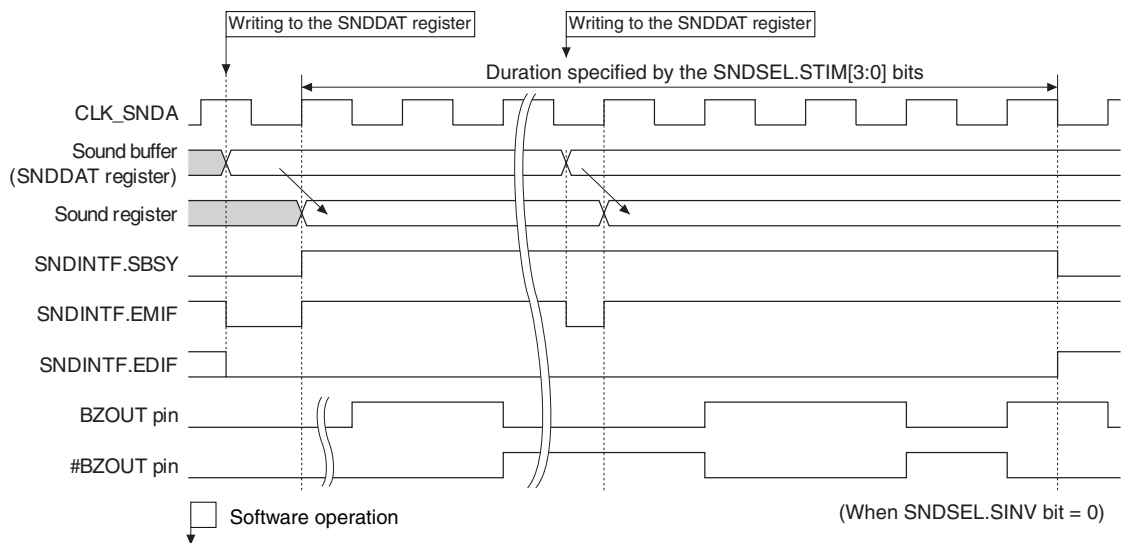


Figure 16.4.3.1 Buzzer Output Timing Chart in One-shot Buzzer Mode

16.4.4 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 SNDA operations are shown below.

Melody output start procedure

1. 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)
2. Write data to the following sound buffer (SNDDAT register) bits. (Start sound output)
 - SNDDAT.MDTI bit (Set tie/slur)
 - SNDDAT.MDRS bit (Set note/rest)
 - SNDDAT.SLEN[5:0] bits (Set duration)
 - SNDDAT.SFRQ[7:0] bits (Set scale)
3. Check to see if the SNDINTF.EMIF bit is set to 1 (an interrupt can be used).
4. Repeat Steps 2 and 3 until the end of the melody.

Melody output operations

When data is written to the sound buffer (SNDDAT register), SNDA 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 16.4.4.1 shows a melody mode operation timing chart.

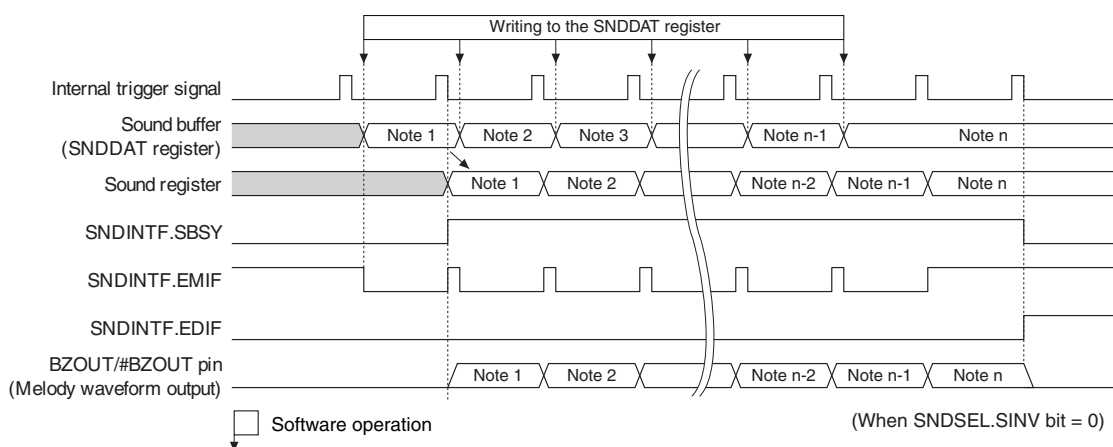


Figure 16.4.4.1 Melody Mode Operation Timing Chart

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 16.4.4.1 Note/Rest Specification (when $f_{CLK_SNDA} = 32,768$ Hz)

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 prohibited	

Tie/slur specification

A tie or slur takes effect by setting the SNDDAT.MDTI bit to 1 and the previous note and the current note are played continuously.

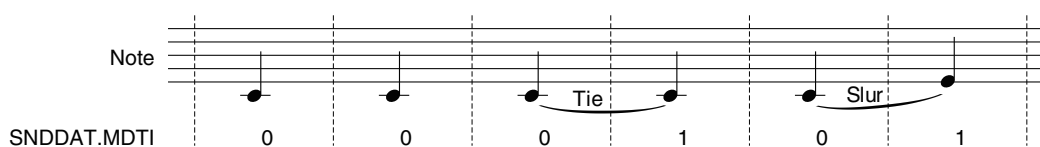


Figure 16.4.4.2 Tie and Slur

Scale specification

Scales can be specified using the SNDDAT.SFRQ[7:0] bits.

Table 16.4.4.2 Scale Specification (when $f_{CLK_SNDA} = 32,768$ Hz)

SNDDAT.SFRQ[7:0] bits	Scale	Frequency [Hz]
0xf8	C3	131.60
0xea	C#3	139.44
0xdd	D3	147.60
0xd1	D#3	156.04
0xc5	E3	165.49
0xba	F3	175.23
0xaf	F#3	186.18
0xa5	G3	197.40
0x9c	G#3	208.71
0x93	A3	221.41
0x8b	A#3	234.06

SNDDAT.SFRQ[7:0] bits	Scale	Frequency [Hz]
0x83	B3	248.24
0x7c	C4	262.14
0x75	C#4	277.69
0x6e	D4	295.21
0x68	D#4	312.08
0x62	E4	330.99
0x5c	F4	352.34
0x57	F#4	372.36
0x52	G4	394.80
0x4e	G#4	414.78
0x49	A4	442.81
0x45	A#4	468.11
0x41	B4	496.48
0x3d	C5	528.52
0x3a	C#5	555.39
0x37	D5	585.14
0x33	D#5	630.15
0x30	E5	668.73
0x2e	F5	697.19
0x2b	F#5	744.73
0x29	G5	780.19
0x26	G#5	840.21
0x24	A5	885.62
0x22	A#5	936.23
0x20	B5	992.97
0x1e	C6	1057.03

16.5 Interrupts

SNDA has a function to generate the interrupts shown in Table 16.5.1.

Table 16.5.1 SNDA Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Sound buffer empty	SNDFINT.EMIF	When data in the sound buffer (SNDDAT register) is transferred to the sound register or 1 is written to the SNDFCTL.SSTP bit	Writing to the SNDDAT register
Sound output completion	SNDFINT.EDIF	When a sound output has completed	Writing 1 or writing to the SNDDAT register

SNDA 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.

16.6 Control Registers

SNDA 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 SNDA 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

16 SOUND GENERATOR (SNDA)

Bits 6–4 CLKDIV[2:0]

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

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of SNDA.

Table 16.6.1 Clock Source and Division Ratio Settings

SNDCLK. CLKDIV[2:0] bits	SNDCLK.CLKSRC[1:0] bits			
	0x0 IOSC	0x1 OSC1	0x2 OSC3	0x3 EXOSC
0x7	Reserved	1/1	Reserved	1/1
0x6				
0x5	1/512		1/512	
0x4	1/256		1/256	
0x3	1/128		1/128	
0x2	1/64		1/64	
0x1	1/32		1/32	
0x0	1/16		1/16	

(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.

SNDA 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 16.6.2 Tempo/One-shot Buzzer Output Duration Selections (when $f_{CLK_SNDA} = 32,768$ Hz)

SNDSEL. STIM[3:0] bits	Tempo (= Quarter note/minute)	One-shot buzzer output duration [ms]
0xf	30	250.0
0xe	32	234.4
0xd	34.3	218.8
0xc	36.9	203.1
0xb	40	187.5
0xa	43.6	171.9
0x9	48	156.3
0x8	53.3	140.6
0x7	60	125.0
0x6	68.6	109.4
0x5	80	93.8
0x4	96	78.1
0x3	120	62.5
0x2	160	46.9
0x1	240	31.3
0x0	480	15.6

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 16.6.3 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

SNDA 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 SNDA operations.

1 (R/W): Enable SNDA operations (The operating clock is supplied.)

0 (R/W): Disable SNDA operations (The operating clock is stopped.)

SNDA 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 or slur (continuous play with the previous note) in melody mode.

1 (R/W): Enable tie/slur

0 (R/W): Disable tie/slur

This bit is ignored in normal buzzer mode/one-shot buzzer mode.

16 SOUND GENERATOR (SNDA)

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.

SNDA 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 Figures 16.4.2.1, 16.4.3.1, and 16.4.4.1.)

1 (R): Outputting

0 (R): Idle

Bits 7–2 Reserved

Bit 1 EMIF

Bit 0 EDIF

These bits indicate the SNDA 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

SNDA 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 SNDA 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

17 IR Remote Controller (REMC3)

17.1 Overview

The REMC3 circuit generates infrared remote control output signals. This circuit can also be applicable to an EL lamp drive circuit by adding a simple external circuit.

The features of the REMC3 are listed below.

- Outputs an infrared remote control signal.
- Includes a carrier generator.
- Flexible carrier signal generation and data pulse width modulation.
- Automatic data setting function for continuous data transmission.
- Output signal inverting function supporting various formats.
- EL lamp drive waveform can be generated for an application example.

Figure 17.1.1 shows the REMC3 configuration.

Table 17.1.1 REMC3 Channel Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
Number of channels	1 transmitter channel			

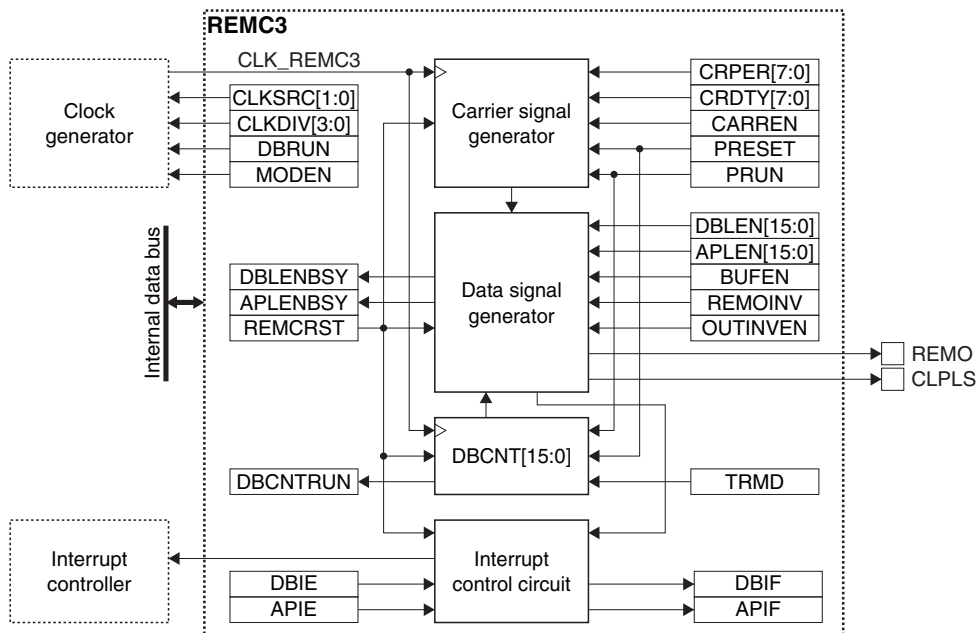


Figure 17.1.1 REMC3 Configuration

17.2 Input/Output Pins and External Connections

17.2.1 Output Pin

Table 17.2.1.1 shows the REMC3 pin.

Table 17.2.1.1 REMC3 Pin

Pin name	I/O*	Initial status*	Function
REMO	O	O (L)	IR remote controller transmit data output
CLPLS	O	O (L)	IR remote controller clear pulse output

* Indicates the status when the pin is configured for the REMC3.

If the port is shared with the REMC3 pin and other functions, the REMC3 output function must be assigned to the port before activating the REMC3. For more information, refer to the “I/O Ports” chapter.

17.2.2 External Connections

Figure 17.2.2.1 shows a connection example between the REMC3 and an external infrared module.

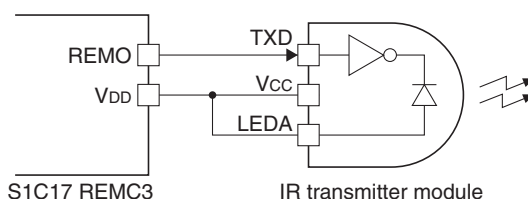


Figure 17.2.2.1 Connection Example Between REMC3 and External Infrared Module

17.3 Clock Settings

17.3.1 REMC3 Operating Clock

When using the REMC3, the REMC3 operating clock CLK_REMC3 must be supplied to the REMC3 from the clock generator. The CLK_REMC3 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 REMCLK register bits:
 - REMCLK.CLKSRC[1:0] bits (Clock source selection)
 - REMCLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)

17.3.2 Clock Supply in SLEEP Mode

When using REMC3 during SLEEP mode, the REMC3 operating clock CLK_REMC3 must be configured so that it will keep supplying by writing 0 to the CLGOSC.xxxxSLPC bit for the CLK_REMC3 clock source.

If the CLGOSC.xxxxSLPC bit for the CLK_REMC3 clock source is 1, the CLK_REMC3 clock source is deactivated during SLEEP mode and REMC3 stops with the register settings maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_REMC3 is supplied and the REMC3 operation resumes.

17.3.3 Clock Supply in DEBUG Mode

The CLK_REMC3 supply during DEBUG mode should be controlled using the REMCLK.DBRUN bit.

The CLK_REMC3 supply to the REMC3 is suspended when the CPU enters DEBUG mode if the REMCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_REMC3 supply resumes. Although the REMC3 stops operating when the CLK_REMC3 supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the REMCLK.DBRUN bit = 1, the CLK_REMC3 supply is not suspended and the REMC3 will keep operating in DEBUG mode.

17.4 Operations

17.4.1 Initialization

The REMC3 should be initialized with the procedure shown below.

1. Write 1 to the REMDBCTL.REMCNST bit. (Reset REMC3)
2. Configure the REMCLK.CLKSRC[1:0] and REMCLK.CLKDIV[3:0] bits. (Configure operating clock)
3. Assign the REMC3 output function to the port. (Refer to the “I/O Ports” chapter.)

4. Configure the following REMDBCTL register bits:
 - Set the REMDBCTL.MODEN bit to 1. (Enable count operation clock)
 - REMDBCTL.TRMD bit (Select repeat mode/one-shot mode)
 - Set the REMDBCTL.BUFEN bit to 1. (Enable compare buffer)
 - REMDBCTL.REMOINV bit (Configure inverse logic output signal)
5. Configure the following REMCARR register bits:
 - REMCARR.CRPER[7:0] bit (Set carrier signal cycle)
 - REMCARR.CRDY[7:0] bit (Set carrier signal duty)
6. Configure the following REMCCTL register bits:
 - REMCCTL.CARREN bit (Enable/disable carrier modulation)
 - REMCCTL.OUTINVEN bit (Configure output signal polarity)
7. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the REMINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the REMINTE register to 1. (Enable interrupts)

17.4.2 Data Transmission Procedures

Starting data transmission

The following shows a procedure to start data transmission.

1. Set the REMAPLEN.APLEN[15:0] bits. (Set data signal duty)
2. Set the REMDBLEN.DBLEN[15:0] bits. (Set data signal cycle)
3. Set the following REMDBCTL register bits:
 - Set the REMDBCTL.PRESET bit to 1. (Reset internal counters)
 - Set the REMDBCTL.PRUN bit to 1. (Start counting)

Continuous data transmission control

The following shows a procedure to send data continuously after starting data transmission (after Step 3 above).

1. Set the duty and cycle for the subsequent data to the REMAPLEN.APLEN[15:0] and REMDBLEN.DBLEN[15:0] bits, respectively, before a compare DB interrupt (REMINTF.DBIF bit = 1) occurs. (It is not necessary to rewrite settings when sending the same data with the current settings.)
2. Wait for a compare DB interrupt (REMINTF.DBIF bit = 1).
3. Repeat Steps 1 and 2 until the end of data.

Terminating data transmission

The following shows a procedure to terminate data transmission.

1. Wait for a compare DB interrupt (REMINTF.DBIF bit = 1).
2. Set the REMDBCTL.PRUN bit to 0. (Stop counting)
3. Set the REMDBCTL.MODEN bit to 0. (Disable count operation clock)

17.4.3 REMO Output Waveform

Carrier refers to infrared frequency in infrared remote control communication. Note, however, that carrier in this manual refers to sub-carrier used in infrared remote control communication, as REMC3 does not control infrared rays directly.

The REMC3 outputs the logical AND between the carrier signal output from the carrier generator and the data signal output from the data signal generator. Figure 17.4.3.1 shows an example of the output waveform.

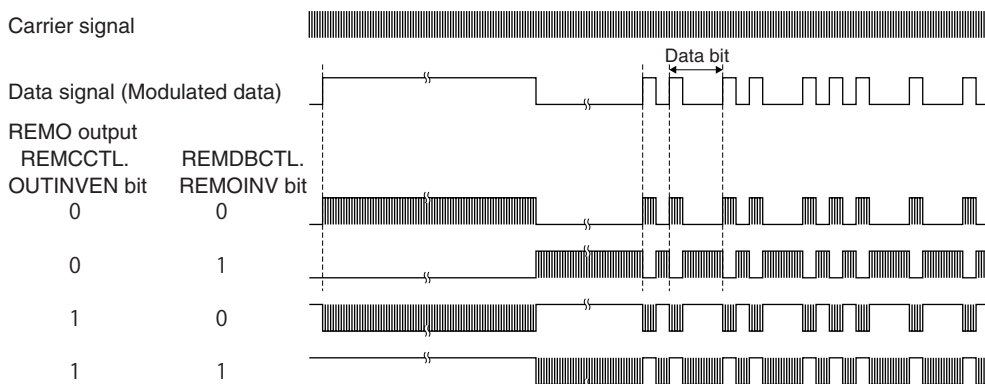


Figure 17.4.3.1 REMO Output Waveform Example

Carrier signal

The carrier signal is generated by comparing the values of the 8-bit counter for carrier generation that runs with CLK_REMC3 and the setting values of the REMCARR.CRDTY[7:0] and REMCARR.CRPER[7:0] bits. Figure 17.4.3.2 shows an example of the carrier signal generated.

Example) REMCARR.CRDTY[7:0] bits = 2, REMCARR.CRPER[7:0] bits = 8

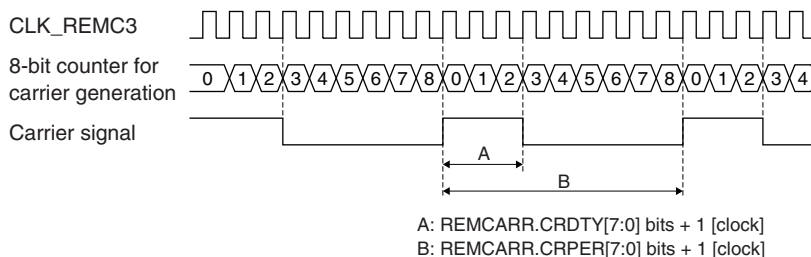


Figure 17.4.3.2 Example of Carrier Signal Generated

The carrier signal frequency and duty ratio can be calculated by the equations shown below.

$$\text{Carrier frequency} = \frac{f_{\text{CLK_REMC3}}}{\text{CRPER} + 1} \quad \text{Duty ratio} = \frac{\text{CRDTY} + 1}{\text{CRPER} + 1} \quad (\text{Eq. 17.1})$$

Where

$f_{\text{CLK_REMC3}}$: CLK_REMC3 frequency [Hz]

CRPER: REMCARR.CRPER[7:0] bit-setting value (1–255)

CRDTY: REMCARR.CRDTY[7:0] bit-setting value (0–254)

* REMCARR.CRDTY[7:0] bits < REMCARR.CRPER[7:0] bits

The 8-bit counter for carrier generation is reset by the REMDBCTL.PRESET bit and is started/stopped by the REMDBCTL.PRUN bit in conjunction with the 16-bit counter for data signal generation. When the counter value is matched with the REMCARR.CRDTY[7:0] bits, the carrier signal waveform is inverted. When the counter value is matched with the REMCARR.CRPER[7:0] bits, the carrier signal waveform is inverted and the counter is reset to 0x00.

Data signal

The data signal is generated by comparing the values of the 16-bit counter for data signal generation (REMDBCNT.DBCNT[15:0] bits) that runs with CLK_REMC3 and the setting values of the REMAPLEN.APLEN[15:0] and REMDBLEN.DBLEN[15:0] bits. Figure 17.4.3.3 shows an example of the data signal generated.

Example) REMAPLEN.APLEN[15:0] bits = 0x0bd0, REMDBLEN.DBLEN[15:0] bits = 0x11b8,
REMDBCTL.TRMD bit = 0 (repeat mode), REMDBCTL.REMOINV bit = 0 (signal logic non-inverted)

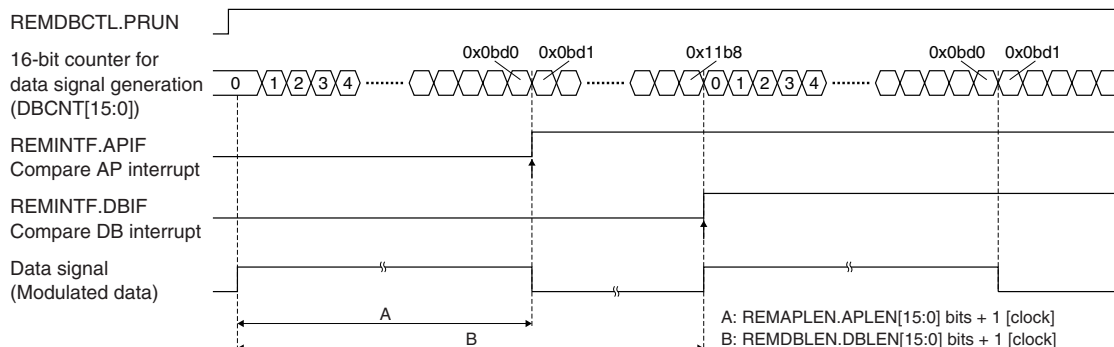


Figure 17.4.3.3 Example of Data Signal Generated

The data length and duty ratio of the pulse-width-modulated data signal can be calculated with the equations shown below.

$$\text{Data length} = \frac{\text{DBLEN} + 1}{f_{\text{CLK_REMC3}}} \quad \text{Duty ratio} = \frac{\text{APLEN} + 1}{\text{DBLEN} + 1} \quad (\text{Eq. 17.2})$$

Where

$f_{\text{CLK_REMC3}}$: CLK_REMC3 frequency [Hz]

DBLEN: REMDBLEN.DBLEN[15:0] bit-setting value (1–65,535)

APLEN: REMAPLEN.APLEN[15:0] bit-setting value (0–65,534)

* REMAPLEN.APLEN[15:0] bits < REMDBLEN.DBLEN[15:0] bits

The 16-bit counter for data signal generation is reset by the REMDBCTL.PRESET bit and is started/stopped by the REMDBCTL.PRUN bit. When the counter value is matched with the REMAPLEN.APLEN[15:0] bits (compare AP), the data signal waveform is inverted. When the counter value is matched with the REMDBLEN.DBLEN[15:0] bits (compare DB), the data signal waveform is inverted and the counter is reset to 0x0000.

A different interrupt can be generated when the counter value is matched with the REMDBLEN.DBLEN[15:0] and REMAPLEN.APLEN[15:0] bits, respectively.

Repeat mode and one-shot mode

When the 16-bit counter for data signal generation is set to repeat mode (REMDBCTL.TRMD bit = 0), the counter keeps operating until it is stopped using the REMDBCTL.PRUN bit. When the counter is set to one-shot mode (REMDBCTL.TRMD bit = 1), the counter stops automatically when the counter value is matched with the REMDBLEN.DBLEN[15:0] bit-setting value.

17.4.4 Continuous Data Transmission and Compare Buffers

Figure 17.4.4.1 shows an operation example of continuous data transmission with the compare buffer enabled.

17 IR REMOTE CONTROLLER (REMC3)

Example) REMDBCTL.TRMD bit = 0 (repeat mode), REMDBCTL.BUFEN bit = 1 (compare buffer enabled), REMDBCTL.REMOINV bit = 0 (signal logic non-inverted)

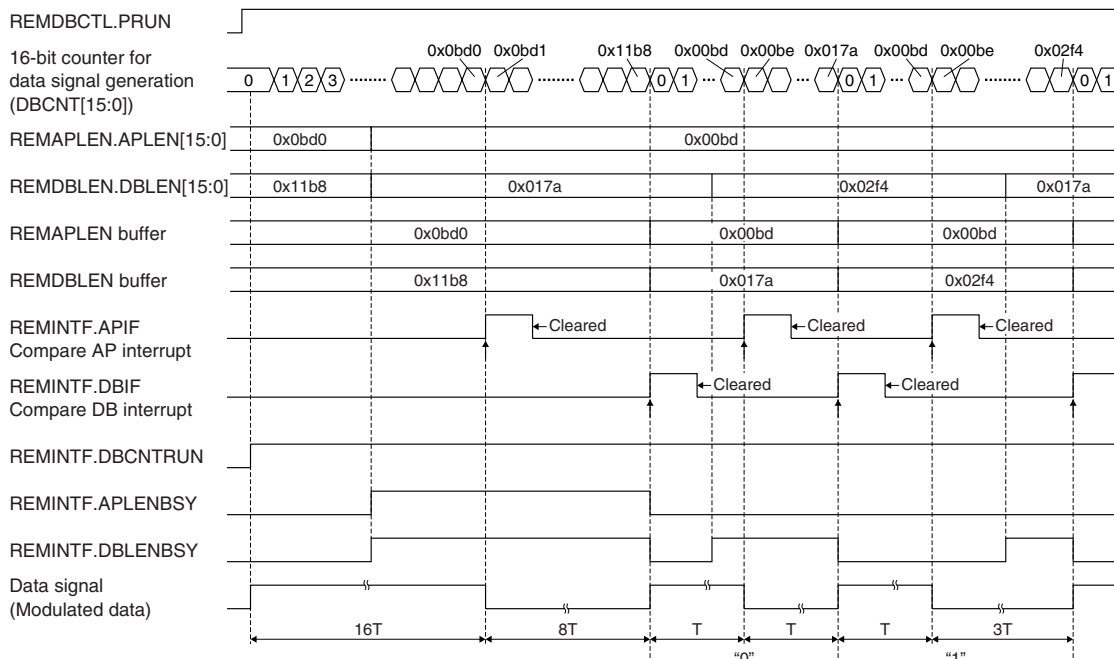


Figure 17.4.4.1 Continuous Data Transmission Example

When the compare buffer is disabled (REMDBCTL.BUFEN bit = 0), the 16-bit counter value is directly compared with the REMAPLEN.APLEN[15:0] and REMDBLEN.DBLEN[15:0] bit values. The comparison value is altered immediately after the REMAPLEN.APLEN[15:0] or REMDBLEN.DBLEN[15:0] bits are rewritten.

When the compare buffer is enabled (REMDBCTL.BUFEN bit = 1), the REMAPLEN.APLEN[15:0] and REMDBLEN.DBLEN[15:0] bit values are loaded into the compare buffers provided respectively (REMAPLEN buffer and REMDBLEN buffer) and the 16-bit counter value is compared with the compare buffers.

The comparison values are loaded into the compare buffers when the 16-bit counter is matched with the REMDBLEN buffer (when the count for the data length has completed). Therefore, the next transmit data can be set during the current data transmission. When the compare buffers are enabled, the buffer status flags (REMINTF.APLENBSY bit and REMINTF.DBLENBSY bit) become effective. The flag is set to 1 when the setting value is written to the register and cleared to 0 when the written value is transferred to the buffer.

17.5 Interrupts

The REMC3 has a function to generate the interrupts shown in Table 17.5.1.

Table 17.5.1 REMC3 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Compare AP	REMINTF.APIF	When the REMAPLEN register (or REMAPLEN buffer) value and the 16-bit counter for data signal generation are matched	Writing 1 to the interrupt flag or the REMDBCTL.REMCRST bit
Compare DB	REMINTF.DBIF	When the REMDBLEN register (or REMDBLEN buffer) value and the 16-bit counter for data signal generation are matched	Writing 1 to the interrupt flag or the REMDBCTL.REMCRST bit

The REMC3 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 Application Example: Driving EL Lamp

The REMC3 can be used to simply drive an EL lamp as an application example. Figures 17.6.1 and 17.6.2 show an example of an EL lamp drive circuit and an example of the drive waveform generated, respectively. For details of settings and an example of components, refer to the Application Note provided separately.

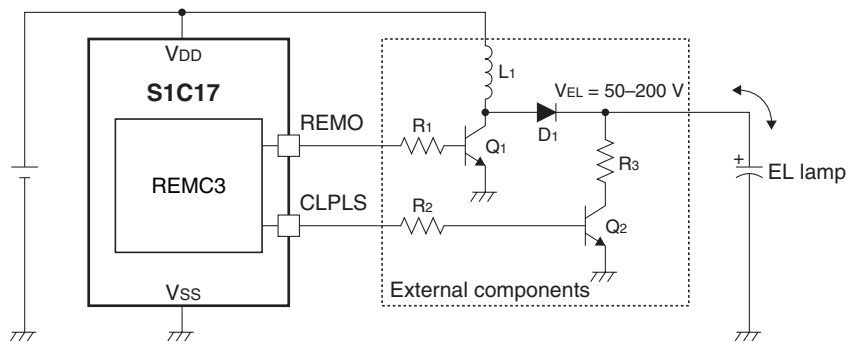


Figure 17.6.1 Example of EL Lamp Drive Circuit

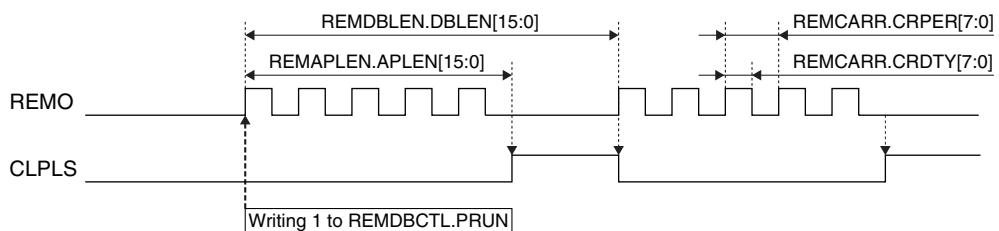


Figure 17.6.2 Example of Generated Drive Waveform

The REMO and CLPLS signals are output from the respective pins while the REMDBCTL.PRUN bit = 1. The difference between the setting values of the REMDBLEN.DBLEN[15:0] bits and REMAPLEN.APLEN[15:0] bits becomes the CLPLS pulse width (high period).

17.7 Control Registers

REMC3 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMCLK	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 REMC3 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 REMC3 operating clock.

Bits 3-2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the REMC3.

Table 17.7.1 Clock Source and Division Ratio Settings

REMCLK. CLKDIV[3:0] bits	REMCLK.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) The oscillation circuits/external input that are not supported in this IC cannot be selected as the clock source.

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

REMC3 Data Bit Counter Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMDBCTL	15–10	–	0x00	–	R	–
	9	PRESET	0	H0/S0	R/W	Cleared by writing 1 to the REMDBCTL.REMCRST bit.
	8	PRUN	0	H0/S0	R/W	
	7–5	–	0x0	–	R	–
	4	REMOINV	0	H0	R/W	
	3	BUFEN	0	H0	R/W	
	2	TRMD	0	H0	R/W	
	1	REMCRST	0	H0	W	
	0	MODEN	0	H0	R/W	

Bits 15–10 Reserved**Bit 9 PRESET**

This bit resets the internal counters (16-bit counter for data signal generation and 8-bit counter for carrier generation).

1 (W): Reset

0 (W): Ineffective

1 (R): Resetting in progress

0 (R): Resetting finished or normal operation

Before the counter can be reset using this bit, the REMDBCTL.MODEN bit must be set to 1.

This bit is cleared to 0 after the counter reset operation has finished or when 1 is written to the REMDBCTL.REMCRST bit.

Bit 8 PRUN

This bit starts/stops counting by the internal counters (16-bit counter for data signal generation and 8-bit counter for carrier generation).

1 (W): Start counting

0 (W): Stop counting

1 (R): Counting

0 (R): Idle

Before the counter can start counting by this bit, the REMDBCTL.MODEN bit must be set to 1. While the counter is running, writing 0 to the REMDBCTL.PRUN bit stops count operations. When the counter stops by occurrence of a compare DB in one-shot mode, this bit is automatically cleared to 0.

Bits 7–5 Reserved

Bit 4 REMOINV

This bit inverts the REMO output signal.

1 (R/W): Inverted

0 (R/W): Non-inverted

For more information, see Figure 17.4.3.1.

Bit 3 BUFEN

This bit enables or disables the compare buffers.

1 (R/W): Enable

0 (R/W): Disable

For more information, refer to “Continuous Data Transmission and Compare Buffers.”

Note: The REMDBCTL.BUFEN bit must be set to 0 when setting the data signal duty and cycle for the first time.

Bit 2 TRMD

This bit selects the operation mode of the 16-bit counter for data signal generation.

1 (R/W): One-shot mode

0 (R/W): Repeat mode

For more information, refer to “REMO Output Waveform, Data signal.”

Bit 1 REMCRST

This bit issues software reset to the REMC3.

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 REMC3 internal counters and interrupt flags. This bit is automatically cleared after the reset processing has finished.

Bit 0 MODEN

This bit enables the REMC3 operations.

1 (R/W): Enable REMC3 operations (The operating clock is supplied.)

0 (R/W): Disable REMC3 operations (The operating clock is stopped.)

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

REMC3 Data Bit Counter Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMDBCNT	15–0	DBCNT[15:0]	0x0000	H0/S0	R	Cleared by writing 1 to the REMDBCTL.REMCRST bit.

Bits 15–0 DBCNT[15:0]

The current value of the 16-bit counter for data signal generation can be read out through these bits.

REMC3 Data Bit Active Pulse Length Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMAPLEN	15–0	APLEN[15:0]	0x0000	H0	R/W	Writing enabled when REMDBCTL.MODEN bit = 1.

Bits 15–0 APLEN[15:0]

These bits set the active pulse length of the data signal (high period when the REMDBCTL.REMOINV bit = 0 or low period when the REMDBCTL.REMOINV bit = 1).

The REMO pin output is set to the active level from the 16-bit counter for data signal generation = 0x0000 and it is inverted to the inactive level when the counter exceeds the REMAPLEN.APLEN[15:0] bit-setting value. The data signal duty ratio is determined by this setting and the REMDBLEN.DBLEN[15:0] bit-setting. (See Figure 17.4.3.3.)

Before this register can be rewritten, the REMDBCTL.MODEN bit must be set to 1.

REMC3 Data Bit Length Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMDBLEN	15–0	DBLEN[15:0]	0x0000	H0	R/W	Writing enabled when REMDBCTL.MODEN bit = 1.

Bits 15–0 DBLEN[15:0]

These bits set the data length of the data signal (length of one cycle).

A data signal cycle begins with the 16-bit counter for data signal generation = 0x0000 and ends when the counter exceeds the REMDBLEN.DBLEN[15:0] bit-setting value. (See Figure 17.4.3.3.)

Before this register can be rewritten, the REMDBCTL.MODEN bit must be set to 1.

REMC3 Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMINTF	15–11	–	0x00	–	R	–
	10	DBCNTRUN	0	H0/S0	R	Cleared by writing 1 to the REMDBCTL.REMCRST bit.
	9	DBLENBSY	0	H0	R	Effective when the REMDBCTL.BUFEN bit = 1.
	8	APLENBSY	0	H0	R	Effective when the REMDBCTL.BUFEN bit = 1.
	7–2	–	0x00	–	R	–
	1	DBIF	0	H0/S0	R/W	Cleared by writing 1 to this bit or the REMDBCTL.REMCRST bit.
	0	APIF	0	H0/S0	R/W	Cleared by writing 1 to this bit or the REMDBCTL.REMCRST bit.

Bits 15–11 Reserved

Bit 10 DBCNTRUN

This bit indicates whether the 16-bit counter for data signal generation is running or not. (See Figure 17.4.4.1.)

1 (R): Running (Counting)

0 (R): Idle

Bit 9 DBLENBSY

This bit indicates whether the value written to the REMDBLEN.DBLEN[15:0] bits is transferred to the REMDBLEN buffer or not. (See Figure 17.4.4.1.)

1 (R): Transfer to the REMDBLEN buffer has not completed.

0 (R): Transfer to the REMDBLEN buffer has completed.

While this bit is set to 1, writing to the REMDBLEN.DBLEN[15:0] bits is ineffective.

Bit 8 APLENBSY

This bit indicates whether the value written to the REMAPLEN.APLEN[15:0] bits is transferred to the REMAPLEN buffer or not. (See Figure 17.4.4.1.)

1 (R): Transfer to the REMAPLEN buffer has not completed.

0 (R): Transfer to the REMAPLEN buffer has completed.

While this bit is set to 1, writing to the REMAPLEN.APLEN[15:0] bits is ineffective.

Bits 7–2 Reserved**Bit 1 DBIF****Bit 0 APIF**

These bits indicate the REMC3 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:

REMINTE.DBIF bit: Compare DB interrupt

REMINTE.APIF bit: Compare AP interrupt

These interrupt flags are also cleared to 0 when 1 is written to the REMDBCTL.REMCRST bit.

REMC3 Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMINTE	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	DBIE	0	H0	R/W	
	0	APIE	0	H0	R/W	

Bits 15–2 Reserved**Bit 1 DBIE****Bit 0 APIE**

These bits enable REMC3 interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

REMINTE.DBIE bit: Compare DB interrupt

REMINTE.APIE bit: Compare AP interrupt

REMC3 Carrier Waveform Register

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

Bits 15–8 CRDTY[7:0]

These bits set the high level period of the carrier signal.

The carrier signal is set to high level from the 8-bit counter for carrier generation = 0x00 and it is inverted to low level when the counter exceeds the REMCARR.CRDTY[7:0] bit-setting value. The carrier signal duty ratio is determined by this setting and the REMCARR.CRPER[7:0] bit-setting. (See Figure 17.4.3.2.)

Bits 7–0 CRPER[7:0]

These bits set the carrier signal cycle.

A carrier signal cycle begins with the 8-bit counter for carrier generation = 0x00 and ends when the counter exceeds the REMCARR.CRPER[7:0] bit-setting value. (See Figure 17.4.3.2.)

REMC3 Carrier Modulation Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMCCTL	15–9	–	0x00	–	R	–
	8	OUTINVEN	0	H0	R/W	
	7–1	–	0x00	–	R	
	0	CARREN	0	H0	R/W	

Bits 15–9 Reserved**Bit 8 OUTINVEN**

This bit inverts the REMO output polarity.

1 (R/W): Inverted

0 (R/W): Non-inverted

For more information, see Figure 17.4.3.1.

Bits 7–1 Reserved**Bit 0 CARREN**

This bit enables carrier modulation.

1 (R/W): Enable carrier modulation

0 (R/W): Disable carrier modulation (output data signal only)

Note: When carrier modulation is disabled, the REMDBCTL.REMOINV bit should be set to 0.

18 R/F Converter (RFC)

18.1 Overview

The RFC is a CR oscillation type A/D converter (R/F converter).

The features of the RFC are listed below.

- 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. (Note: See the table below.)
(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 18.1.1 shows the RFC configuration.

Table 18.1.1 RFC Channel Configuration of S1C17M20/M21/M22/M23/M24/M25

Item	S1C17M20/M23		S1C17M21/M24	S1C17M22/M25
	24-pin package	32-pin package		
Number of channels	-			2 channels (Ch.0 and Ch.1) * Can only be used in DC oscillation mode for resistive sensor measurements.

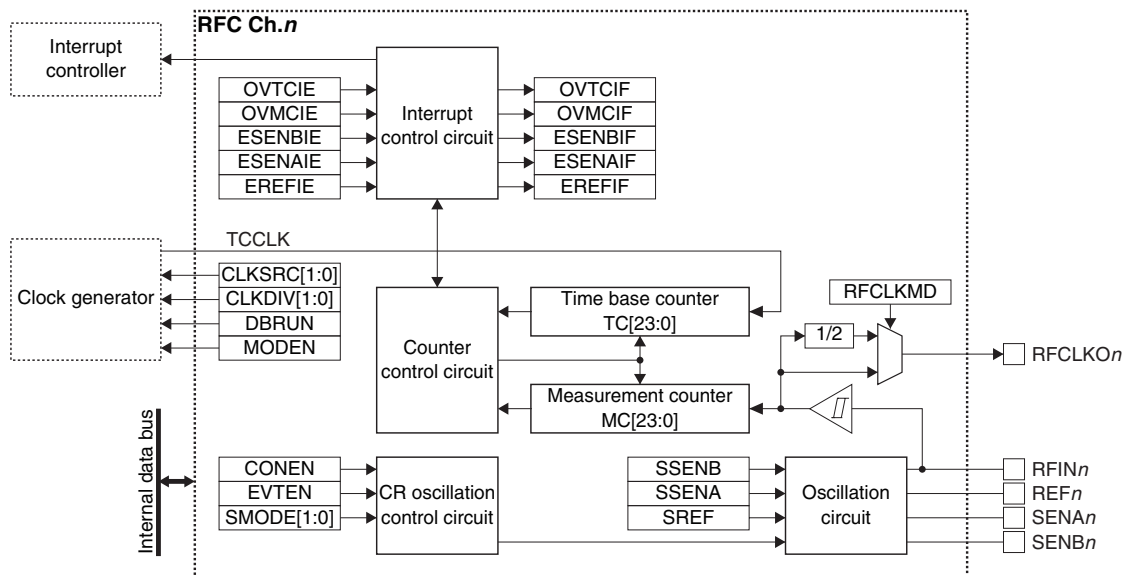


Figure 18.1.1 RFC Configuration

18.2 Input/Output Pins and External Connections

18.2.1 List of Input/Output Pins

Table 18.2.1.1 lists the RFC pins.

Table 18.2.1.1 List of RFC Pins

Pin name	I/O*	Initial status*	Function
SENB n	A	Hi-Z	Sensor B oscillation control pin
SENA n	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
RFCLKOn	O	Hi-Z	RFCLK monitoring output pin RFCLK is output to monitor the oscillation frequency.

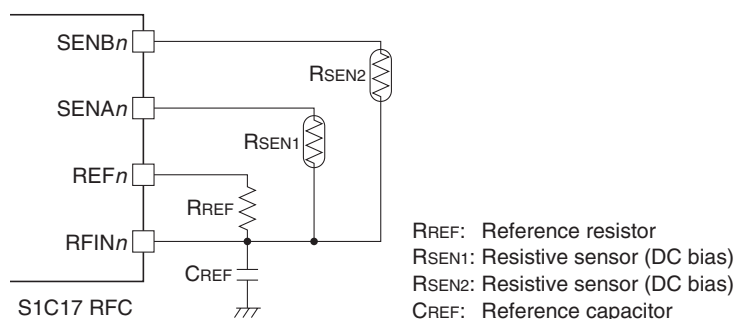
* Indicates the status when the pin is configured for the RFC.

If the port is shared with the RFC pin and other functions, the RFC input/output function must be assigned to the port before activating the RFC. 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.

18.2.2 External Connections

The figures below show connection examples between the RFC and external sensors. For the oscillation mode and external clock input mode, refer to “Operating Mode.”



* Leave the unused pin (SENA n or SENB n) open if one resistive sensor only is used.

Figure 18.2.2.1 Connection Example in Resistive Sensor DC Oscillation Mode

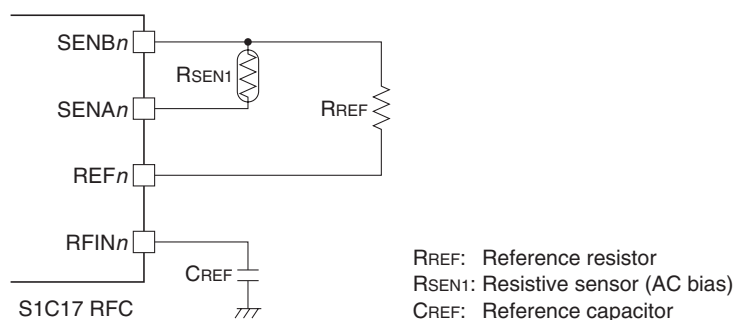
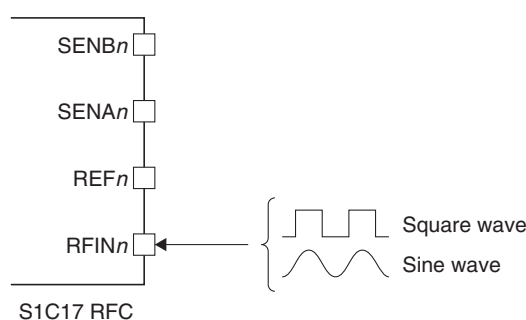


Figure 18.2.2.2 Connection Example in Resistive Sensor AC Oscillation Mode



* Leave the unused pins open.

Figure 18.2.2.3 External Clock Input in External Clock Input Mode

18.3 Clock Settings

18.3.1 RFC Operating Clock

When using the RFC, the RFC operating clock TCCLK must be supplied to the RFC 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 RFCnCLK register bits:
 - RFCnCLK.CLKSRC[1:0] bits (Clock source selection)
 - RFCnCLK.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.

18.3.2 Clock Supply in SLEEP Mode

When using RFC during SLEEP mode, the RFC 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.

18.3.3 Clock Supply in DEBUG Mode

The TCCLK supply during DEBUG mode should be controlled using the RFCnCLK.DBRUN bit.

The TCCLK supply to the RFC is suspended when the CPU enters DEBUG mode if the RFCnCLK.DBRUN bit = 0. After the CPU returns to normal mode, the TCCLK supply resumes. Although the RFC stops operating when the TCCLK supply is suspended, the output pin and registers retain the status before DEBUG mode was entered. If the RFCnCLK.DBRUN bit = 1, the TCCLK supply is not suspended and the RFC will keep operating in DEBUG mode.

18.4 Operations

18.4.1 Initialization

The RFC should be initialized with the procedure shown below.

1. Configure the RFCnCLK.CLKSRC[1:0] and RFCnCLK.CLKDIV[1:0] bits. (Configure operating clock)
2. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the RFCnINTF register. (Clear interrupt flags)
 - Set the interrupt enable bits in the RFCnINTE register to 1. (Enable interrupts)
3. Assign the RFC input/output function to the ports. (Refer to the “I/O Ports” chapter.)

4. Configure the following RFCnCTL register bits:

- RFCnCTL.EVTEN bit (Enable/disable external clock input mode)
- RFCnCTL.SMODE[1:0] bits (Select oscillation mode)
- Set the RFCnCTL.MODEN bit to 1. (Enable RFC operations)

18.4.2 Operating Modes

The RFC has two oscillation modes that use the RFC 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 RFCnCTL.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 RFC 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 RFC 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 RFCnCTL.EVTEN bit to 1. The measurement procedure is the same as when the internal oscillation circuit is used.

18.4.3 RFC Counters

The RFC 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.

18.4.4 Converting Operations and Control Procedure

An R/F conversion procedure and the RFC 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 RFC 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 RFC 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 RFC 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 RFC 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.

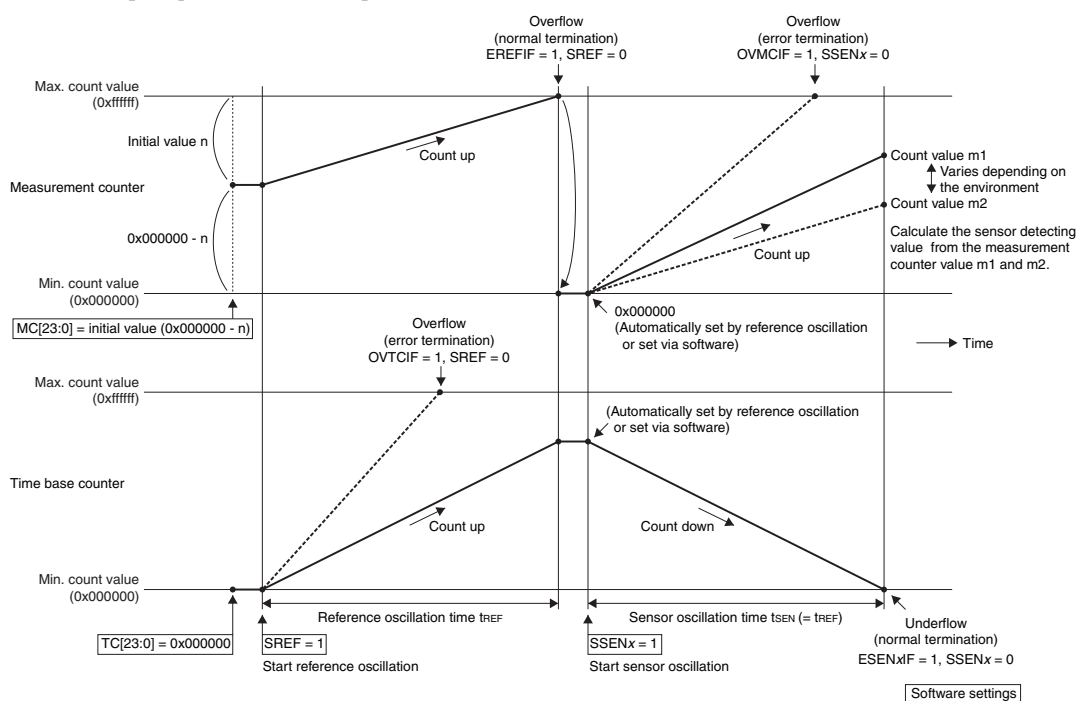


Figure 18.4.4.1 Counter Operations During Reference/Sensor Oscillation

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 18.4.4.1 lists the error factors. (n : measurement counter initial value, m : measurement counter value at the end of sensor oscillation)

Table 18.4.4.1 Error Factors

Error factor	Influence
External part tolerances	Large
Power supply voltage fluctuations	Large
Parasitic capacitance and resistance of the board	Middle
Temperature	Small
Unevenness of IC quality	Small

18.4.5 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 RFC 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.

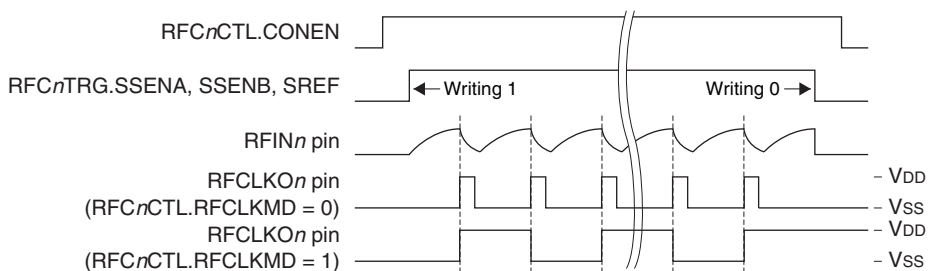


Figure 18.4.5.1 CR Oscillation Clock (RFCLK) Waveform

18.5 Interrupts

The RFC has a function to generate the interrupts shown in Table 18.5.1.

Table 18.5.1 RFC 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 RFC 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.

18.6 Control Registers

RFC 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	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the RFC 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 RFC operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the RFC.

Table 18.6.1 Clock Source and Division Ratio Settings

RFCnCLK. CLKDIV[1:0] bits	RFCnCLK.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 RFCnCLK register settings can be altered only when the RFCnCTL.MODEN bit = 0.

RFC Ch.n Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnCTL	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 RFCLKOn 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 $RFCnCTL.EVTEN$ bit is set to 1. The $RFINn$ 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 18.6.2 Oscillation Mode Selection

$RFCnCTL.SMODE[1:0]$ bits	Oscillation mode
0x3, 0x2	Reserved
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 RFC operations.

1 (R/W): Enable RFC operations (The operating clock is supplied.)

0 (R/W): Disable RFC 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.

RFC 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 (RFC 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.

RFC Ch.n Measurement Counter Low and High Registers

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnMCL	15–0	MC[15:0]	0x0000	H0	R/W	–
RFCnMCH	15–8	–	0x00	–	R	–
	7–0	MC[23:0]	0x00	H0	R/W	

Or

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnMCL	31–24	–	0x00	–	R	–
RFCnMCH	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 (RFCnMCL.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 (RFCnMCH.MC[23:16] bits) is written first.

RFC Ch.n Time Base Counter Low and High Registers

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnTCL	15–0	TC[15:0]	0x0000	H0	R/W	–
RFCnTCH	15–8	–	0x00	–	R	–
	7–0	TC[23:16]	0x00	H0	R/W	

Or

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnTCL	31–24	–	0x00	–	R	–
RFCnTCH	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 (RFCnTCL.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 (RFCnTCH.TC[23:16] bits) is written first.

RFC Ch.n Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnINTF	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	
	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 RFC 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:

RFCnINTF.OVTCIF bit: Time base counter overflow error interrupt

RFCnINTF.OVMCIF bit: Measurement counter overflow error interrupt

RFCnINTF.ESENBIF bit: Sensor B oscillation completion interrupt

RFCnINTF.ESENAIF bit: Sensor A oscillation completion interrupt

RFCnINTF.EREFIF bit: Reference oscillation completion interrupt

RFC Ch.n Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFCnINTE	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 RFC interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

RFCnINTE.OVTCIE bit: Time base counter overflow error interrupt

RFCnINTE.OVMCIE bit: Measurement counter overflow error interrupt

RFCnINTE.ESENBIE bit: Sensor B oscillation completion interrupt

RFCnINTE.ESENAIE bit: Sensor A oscillation completion interrupt

RFCnINTE.EREFIE bit: Reference oscillation completion interrupt

19.2 Input Pins and External Connections

19.2.1 List of Input Pins

Table 19.2.1.1 lists the ADC12A pins.

Table 19.2.1.1 List of ADC12A Pins

Pin name	I/O*	Initial status*	Function
ADIN n m	A	Hi-Z	Analog signal input
#ADTRG n	I	I	External trigger input
VREFAn	A	Hi-Z	Reference voltage input

* Indicates the status when the pin is configured for the ADC12A.

If the port is shared with the ADC12A pin and other functions, the ADC12A input function must be assigned to the port before activating the ADC12A. For more information, refer to the “I/O Ports” chapter.

19.2.2 External Connections

Figure 19.2.2.1 shows a connection diagram between the ADC12A and external devices.

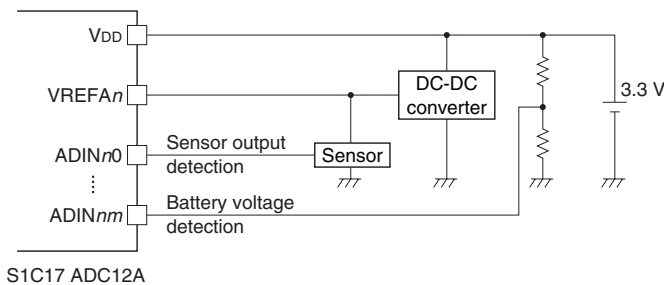


Figure 19.2.2.1 Connections between ADC12A and External Devices

19.3 Clock Settings

19.3.1 ADC12A Operating Clock

The 16-bit timer Ch.k operating clock CLK_T16_k is also used as the ADC12A operating clock. For more information on the CLK_T16_k settings and clock supply in SLEEP and DEBUG modes, refer to “Clock Settings” in the “16-bit Timers” chapter.

Note: When the CLK_T16_k supply stops during A/D conversion (e.g., when the CPU enters SLEEP or DEBUG mode), correct conversion results cannot be obtained even if the clock supply is resumed after that. In this case, perform A/D conversion again.

19.3.2 Sampling Time

The ADC12A includes a sample and hold circuit. The sampling time must be set so that it will satisfy the time required for acquiring input voltage (t_{ACQ} : acquisition time). Figure 19.3.2.1 shows an equivalent circuit of the analog input portion.

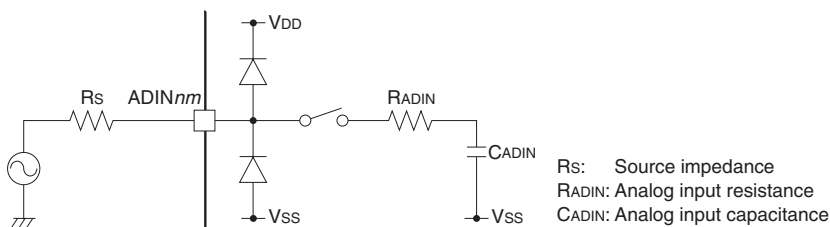


Figure 19.3.2.1 Equivalent Circuit of Analog Input Portion

For the R_{ADIN} and C_{ADIN} values in the equivalent circuit, refer to “12-bit A/D Converter Characteristics” in the “Electrical Characteristics” chapter. Based on these values, configure the ADC12A operating clock CLK_T16_k and the $ADC12_nTRG.SMPCLK[2:0]$ bits that set the sampling time so that these settings will satisfy the equations shown below.

$$t_{ACQ} = 8 \times (R_S + R_{ADIN}) \times C_{ADIN} \quad (\text{Eq. 19.1})$$

$$\frac{1}{f_{CLK_ADC}} \times SMPCLK > t_{ACQ} \quad (\text{Eq. 19.2})$$

Where

f_{CLK_ADC} : CLK_T16_k frequency [Hz]

$SMPCLK$: Sampling time = $ADC12_nTRG.SMPCLK[2:0]$ bit-setting (4 to 11 CLK_T16_k cycles)

The following shows the relationship between the sampling time and the maximum sampling rate.

$$\text{Maximum sampling rate [sps]} = \frac{f_{CLK_ADC}}{SMPCLK + 13} \quad (\text{Eq. 19.3})$$

19.4 Operations

19.4.1 Initialization

The ADC12A should be initialized with the procedure shown below.

1. Assign the ADC12A input function to the ports. (Refer to the “I/O Ports” chapter.)
2. Configure the 16-bit timer $Ch.k$ operating clock so that it will satisfy the sampling time.
3. Set the $ADC12_nCTL.MODEN$ bit to 1. (Enable ADC12A operations)
4. Configure the following $ADC12_nTRG$ register bits:
 - $ADC12_nTRG.SMPCLK[2:0]$ bits (Set sampling time)
 - $ADC12_nTRG.CNVTRG[1:0]$ bits (Select conversion start trigger source)
 - $ADC12_nTRG.CNVMD$ bit (Set conversion mode)
 - $ADC12_nTRG.STMD$ bit (Set data storing mode)
 - $ADC12_nTRG.STAAIN[2:0]$ bits (Set analog input pin to be A/D converted first)
 - $ADC12_nTRG.ENDAIN[2:0]$ bits (Set analog input pin to be A/D converted last)
5. Set the $ADC12_nCFG.VRANGE[1:0]$ bits. (Set operating voltage range according to V_{DD})
6. Set the following bits when using the interrupt:
 - Write 1 to the interrupt flags in the $ADC12_nINTF$ register. (Clear interrupt flags)
 - Set the interrupt enable bits in the $ADC12_nINTE$ register to 1. (Enable interrupts)

19.4.2 Conversion Start Trigger Source

The trigger source, which starts A/D conversion, can be selected from the three types shown below using the $ADC12_nTRG.CNVTRG[1:0]$ bits.

External trigger (#ADTRG n pin)

Writing 1 to the $ADC12_nCTL.ADST$ bit enables the ADC12A to accept trigger inputs. After that, the falling edge of the signal input to the #ADTRG n pin starts A/D conversion.

16-bit timer $Ch.k$ underflow trigger

Writing 1 to the $ADC12_nCTL.ADST$ bit enables the ADC12A to accept trigger inputs. After that, A/D conversion is started when an underflow occurs in the 16-bit timer $Ch.k$.

Software trigger

Writing 1 to the $ADC12_nCTL.ADST$ bit starts A/D conversion.

Trigger inputs can be accepted while the $ADC12_nCTL.BSYSTAT$ bit is set to 0 and are ignored while set to 1.

A/D conversion is actually started in sync with CLK_T16_k after a trigger is accepted.

Writing 0 to the $ADC12_nCTL.ADST$ bit stops A/D conversion after the one currently being executed has completed.

19.4.3 Conversion Mode and Analog Input Pin Settings

The ADC12A can be put into two conversion modes shown below using the ADC12_nTRG.CNVMD bit. Each mode allows setting of analog input pin range to be A/D converted. The analog input pin range can be set using the ADC12_nTRG.STAAIN[2:0] bits for specifying the first analog input pin and the ADC12_nTRG.ENDAIN[2:0] bits for specifying the last analog input pin. The analog input signals within the specified range are A/D converted successively in ascending order of the pin numbers.

One-time conversion mode

Once the ADC12A executes A/D conversion for all the analog input signals within the specified range, it is automatically stopped.

Continuous conversion mode

The ADC12A repeatedly executes A/D conversion within the specified range until 0 is written to the ADC12_nCTL.ADST bit.

19.4.4 A/D Conversion Operations and Control Procedures

The following shows A/D conversion control procedures and the ADC12A operations.

Control procedure in one-time conversion mode

1. Write 1 to the ADC12_nCTL.ADST bit.
2. Wait for an ADC12A interrupt.
 - i. If the ADC12_nINTF.ADmCIF bit = 1 (analog input signal *m* A/D conversion completion interrupt), clear the ADC12_nINTF.ADmCIF bit and then go to Step 3.
 - ii. If the ADC12_nINTF.ADmOVIF bit = 1 (analog input signal *m* A/D conversion result overwrite error interrupt), clear the ADC12_nINTF.ADmOVIF bit and terminate as an error or retry A/D conversion.
3. Read the A/D conversion result of the analog input *m* (ADC12_nADmD.ADmD[15:0] bits).
 - * The 12-bit conversion results are located at the low-order 12 bits or high-order 12-bits within the ADC12_nADmD.ADmD[15:0] bits according to the ADC12_nTRG.STMD bit setting.
4. Repeat Steps 2 and 3 until A/D conversion for all the analog input pins within the specified range is completed.
5. To forcefully terminate the A/D conversion being executed, write 0 to the ADC12_nCTL.ADST bit.
The ADC12A stops operating after the A/D conversion currently being executed has completed.
The ADC12_nCTL.ADST bit must be cleared by writing 0 even if A/D conversion is completed and automatically stopped.

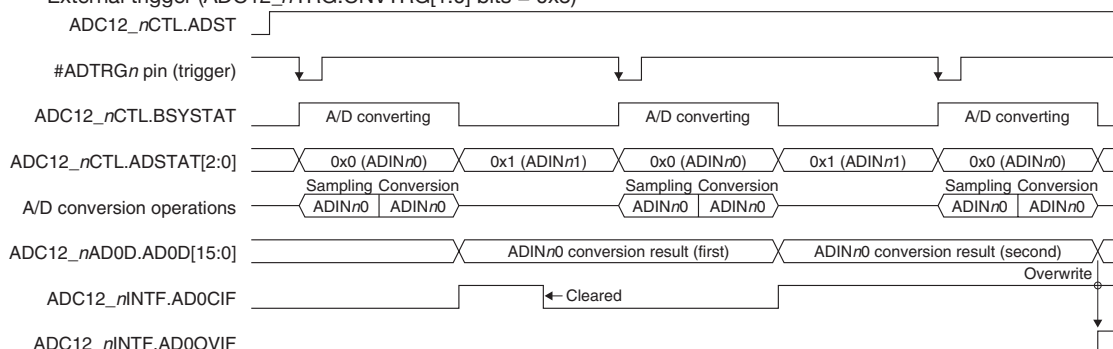
Control procedure in continuous conversion mode

1. Write 1 to the ADC12_nCTL.ADST bit.
2. Wait for an ADC12A interrupt.
 - i. If the ADC12_nINTF.ADmCIF bit = 1 (analog input signal *m* A/D conversion completion interrupt), clear the ADC12_nINTF.ADmCIF bit and then go to Step 3.
 - ii. If the ADC12_nINTF.ADmOVIF bit = 1 (analog input signal *m* A/D conversion result overwrite error interrupt), clear the ADC12_nINTF.ADmOVIF bit and terminate as an error or retry A/D conversion.
3. Read the A/D conversion result of the analog input *m* (ADC12_nADmD.ADmD[15:0] bits).
4. Repeat Steps 2 and 3 until terminating A/D conversion.
5. Write 0 to the ADC12_nCTL.ADST bit.
The ADC12A stops operating after the A/D conversion currently being executed has completed.

(1) One-time conversion mode (ADC12_nTRG.CNVMD bit = 0)

A/D conversion for ADIN_{n0} (ADC12_nTRG.STAAIN[2:0] bits = 0x0, ADC12_nTRG.ENDAIN[2:0] bits = 0x0)

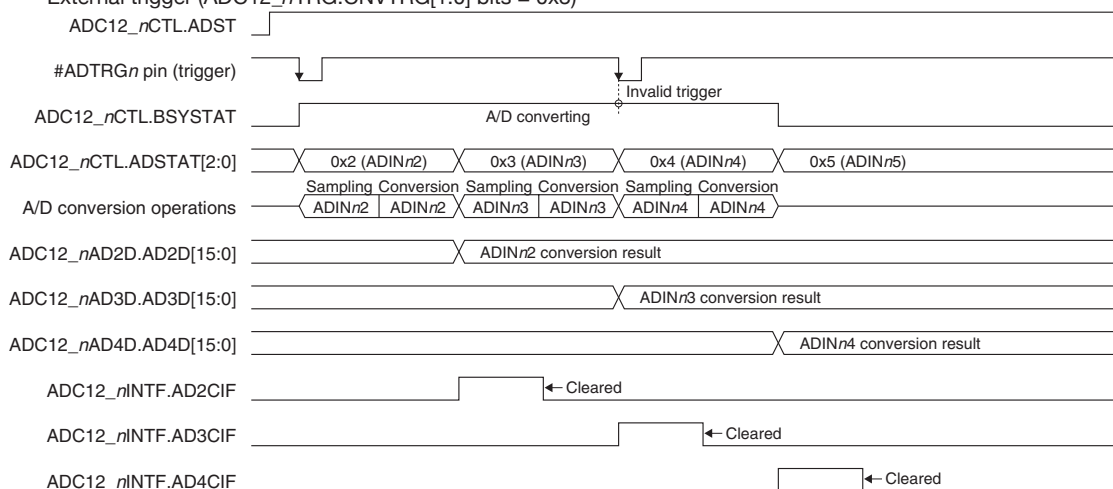
External trigger (ADC12_nTRG.CNVTRG[1:0] bits = 0x3)



(2) One-time conversion mode (ADC12_nTRG.CNVMD bit = 0)

A/D conversion for ADIN_{n2-4} (ADC12_nTRG.STAAIN[2:0] bits = 0x2, ADC12_nTRG.ENDAIN[2:0] bits = 0x4)

External trigger (ADC12_nTRG.CNVTRG[1:0] bits = 0x3)



(3) Continuous conversion mode (ADC12_nTRG.CNVMD bit = 1)

A/D conversion for ADIN_{n3-4} (ADC12_nTRG.STAAIN[2:0] bits = 0x3, ADC12_nTRG.ENDAIN[2:0] bits = 0x4)

Software trigger (ADC12_nTRG.CNVTRG[1:0] bits = 0x0)

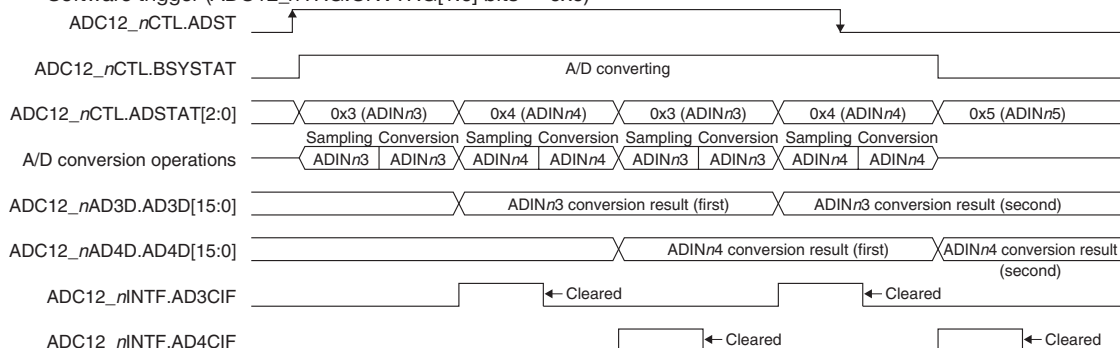


Figure 19.4.4.1 A/D Conversion Operations

19.5 Interrupts

The ADC12A has a function to generate the interrupts shown in Table 19.5.1.

Table 19.5.1 ADC12A Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Analog input signal <i>m</i> A/D conversion completion	ADC12_nINTF.ADmCIF	When an analog input signal <i>m</i> A/D conversion result is loaded to the ADC12_nADmD register	Writing 1
Analog input signal <i>m</i> A/D conversion result overwrite error	ADC12_nINTF.ADmOVIF	When a new A/D conversion result is loaded to the ADC12_nADmD register while the ADC12_nINTF.ADmCIF bit = 1	Writing 1

Note that the A/D conversion continues even if an A/D conversion result overwrite error has occurred. A/D conversion result overwrite errors are decided regardless of whether the ADC12_nADmD register has been read or not.

The ADC12A 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.

19.6 Control Registers

ADC12A Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12_nCTL	15	–	0	–	R	–
	14–12	ADSTAT[2:0]	0x0	H0	R	
	11	–	0	–	R	
	10	BSYSTAT	0	H0	R	
	9–8	–	0x0	–	R	
	7–2	–	0x00	–	R	
	1	ADST	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bit 15 **Reserved**

Bits 14–12 ADSTAT[2:0]

These bits indicate the analog input pin number *m* being A/D converted.

Table 19.6.1 Relationship Between Control Bit Value and Analog Input Pin

ADC12_nCTL.ADSTAT[2:0] bits ADC12_nTRG.STAAIN[2:0] bits ADC12_nTRG.ENDAIN[2:0] bits	Analog input pin
0x7	ADIN _{<i>n</i>7}
0x6	ADIN _{<i>n</i>6}
0x5	ADIN _{<i>n</i>5}
0x4	ADIN _{<i>n</i>4}
0x3	ADIN _{<i>n</i>3}
0x2	ADIN _{<i>n</i>2}
0x1	ADIN _{<i>n</i>1}
0x0	ADIN _{<i>n</i>0}

These bits indicate the last converted analog input pin number after A/D conversion is forcefully terminated by writing 0 to the ADC12_nCTL.ADST bit or automatically terminated in one-time conversion mode (ADC12_nTRG.CNVMD = 0). If A/D conversion is stopped after the maximum analog input pin number (different in each model) has been completed, these bits indicate ADIN_{*n*0}.

Bit 11 **Reserved**

Bit 10 BSYSSTAT

This bit indicates whether the ADC12A is executing A/D conversion or not.

1 (R/W): A/D converting

0 (R/W): Idle

Bits 9–2 Reserved**Bit 1 ADST**

This bit starts A/D conversion or enables to accept triggers.

1 (R/W): Start sampling and conversion (software trigger)/

Enable trigger acceptance (external trigger, 16-bit timer underflow trigger)

0 (R/W): Terminate conversion

This bit does not revert to 0 automatically after A/D conversion has completed. Write 0 to this bit once and write 1 again to start another A/D conversion. After 0 is written to this bit to forcefully terminate conversion, the ADC12A stops after the A/D conversion being executed is completed. Therefore, this bit cannot be used to determine whether the ADC12A is executing A/D conversion or not.

Note: The data written to the ADC12_nCTL.ADST bit must be retained for one or more CLK_T16_k clock cycles when 1 is written or two or more CLK_T16_k clock cycles when 0 is written.

Bit 0 MODEN

This bit enables the ADC12A operations.

1 (R/W): Enable ADC12A operations (The operating clock is supplied.)

0 (R/W): Disable ADC12A operations (The operating clock is stopped.)

Note: After 0 is written to the ADC12_nCTL.MODEN bit, the ADC12A executes a terminate processing. Before the clock source is deactivated, read the ADC12_nCTL.MODEN bit to make sure that it is set to 0.

ADC12A Ch.n Trigger/Analog Input Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12_nTRG	15–14	–	0x0	–	R	–
	13–11	ENDAIN[2:0]	0x0	H0	R/W	
	10–8	STAAIN[2:0]	0x0	H0	R/W	
	7	STMD	0	H0	R/W	
	6	CNVMD	0	H0	R/W	
	5–4	CNVTRG[1:0]	0x0	H0	R/W	
	3	–	0	–	R	
	2–0	SMPCLK[2:0]	0x7	H0	R/W	

Note: Make sure that the ADC12_nCTL.BSYSSTAT bit is set to 0 before altering the ADC12_nTRG register.

Bits 15–14 Reserved**Bits 13–11 ENDAIN[2:0]**

These bits set the analog input pin to be A/D converted last.

See Table 19.6.1 for the relationship between analog input pins and bit setting values.

Note: The analog input pin range to perform A/D conversion must be set as ADC12_nTRG.ENDAIN[2:0] bits \geq ADC12_nTRG.STAAIN[2:0] bits.

Bits 10–8 STAAIN[2:0]

These bits set the analog input pin to be A/D converted first.

See Table 19.6.1 for the relationship between analog input pins and bit setting values.

Bit 7 STMD

This bit selects the data alignment when the conversion results are loaded into the A/D conversion result registers (ADC12_nADmD.ADmD[15:0] bits).

1 (R/W): Left justify

0 (R/W): Right justify

All the A/D conversion result registers change their data alignment immediately after this bit is altered. This does not affect the conversion results.

ADC12_nADmD.ADmD[15:0] bits																			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Left justified (ADC12_nTRG.STMD bit = 1)				(MSB) 12-bit conversion result (LSB)								0 0 0 0							
Right justified (ADC12_nTRG.STMD bit = 0)				0 0 0 0				(MSB) 12-bit conversion result (LSB)											

Figure 19.6.1 Conversion Data Alignment

Bit 6 CNVMD

This bit sets the A/D conversion mode.

1 (R/W): Continuous conversion mode

0 (R/W): One-time conversion mode

Bits 5–4 CNVTRG[1:0]

These bits select a trigger source to start A/D conversion.

Table 19.6.2 Trigger Source Selection

ADC12_nTRG.CNVTRG[1:0] bits	Trigger source
0x3	#ADTRGn pin (external trigger)
0x2	Reserved
0x1	16-bit timer Ch.k underflow
0x0	ADC12_nCTL.ADST bit (software trigger)

Bit 3 Reserved**Bits 2–0 SMPCLK[2:0]**

These bits set the analog input signal sampling time.

Table 19.6.3 Sampling Time Settings

ADC12_nTRG.SMPCLK[2:0] bits	Sampling time (Number of CLK_T16_k cycles)
0x7	11 cycles
0x6	10 cycles
0x5	9 cycles
0x4	8 cycles
0x3	7 cycles
0x2	6 cycles
0x1	5 cycles
0x0	4 cycles

ADC12A Ch.n Configuration Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12_nCFG	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1–0	VRANGE[1:0]	0x0	H0	R/W	

Note: Make sure that the ADC12_nCTL.BSSTAT bit is set to 0 before altering the ADC12_nCFG register.

Bits 15–2 Reserved**Bits 1–0 VRANGE[1:0]**

These bits set the A/D converter operating voltage range.

Table 19.6.4 A/D Converter Operating Voltage Range Setting

ADC12_nCFG.VRANGE[1:0] bits	A/D converter operating voltage range
0x3	1.8 to 5.5 V
0x2	3.6 to 5.5 V
0x1	4.8 to 5.5 V
0x0	Conversion disabled

- Notes:**
- A/D conversion will not be performed if the ADC12_nCFG.VRANGE[1:0] bits = 0x0. Set these bits to the value according to the operating voltage to perform A/D conversion.
 - Be aware that ADC circuit current I_{ADC} flows if the ADC12_nCFG.VRANGE[1:0] bits are set to a value other than 0x0 when the ADC12_nCTL.BSYSTAT bit = 1.

ADC12A Ch.n Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12_nINTF	15	AD7OVIF	0	H0	R/W	Cleared by writing 1.
	14	AD6OVIF	0	H0	R/W	
	13	AD5OVIF	0	H0	R/W	
	12	AD4OVIF	0	H0	R/W	
	11	AD3OVIF	0	H0	R/W	
	10	AD2OVIF	0	H0	R/W	
	9	AD1OVIF	0	H0	R/W	
	8	AD0OVIF	0	H0	R/W	
	7	AD7CIF	0	H0	R/W	
	6	AD6CIF	0	H0	R/W	
	5	AD5CIF	0	H0	R/W	
	4	AD4CIF	0	H0	R/W	
	3	AD3CIF	0	H0	R/W	
	2	AD2CIF	0	H0	R/W	
	1	AD1CIF	0	H0	R/W	
	0	AD0CIF	0	H0	R/W	

Bits 15–8 ADmOVIF

Bits 7–0 ADmCIF

These bits indicate the ADC12A 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:

ADC12_nINTF.ADmOVIF bit: Analog input signal m A/D conversion result overwrite error interrupt

ADC12_nINTF.ADmCIF bit: Analog input signal m A/D conversion completion interrupt

ADC12A Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12_ <i>n</i> INTE	15	AD7OVIE	0	H0	R/W	–
	14	AD6OVIE	0	H0	R/W	
	13	AD5OVIE	0	H0	R/W	
	12	AD4OVIE	0	H0	R/W	
	11	AD3OVIE	0	H0	R/W	
	10	AD2OVIE	0	H0	R/W	
	9	AD1OVIE	0	H0	R/W	
	8	AD0OVIE	0	H0	R/W	
	7	AD7CIE	0	H0	R/W	
	6	AD6CIE	0	H0	R/W	
	5	AD5CIE	0	H0	R/W	
	4	AD4CIE	0	H0	R/W	
	3	AD3CIE	0	H0	R/W	
	2	AD2CIE	0	H0	R/W	
	1	AD1CIE	0	H0	R/W	
	0	AD0CIE	0	H0	R/W	

Bits 15–8 AD*m*OVIE**Bits 7–0 AD*m*CIE**

These bits enable ADC12A interrupts.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:

ADC12_*n*INTE.AD*m*OVIE bit: Analog input signal *m* A/D conversion result overwrite error interrupt

ADC12_*n*INTE.AD*m*CIE bit: Analog input signal *m* A/D conversion completion interrupt

ADC12A Ch.*n* Result Register *m*

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12_ <i>n</i> AD <i>m</i> D	15–0	AD <i>m</i> D[15:0]	0x0000	H0	R	–

Bits 15–0 AD*m*D[15:0]

These bits are the A/D conversion results of the analog input signal *m*.

20 Multiplier/Divider (COPRO2)

20.1 Overview

COPRO2 is the coprocessor that provides multiplier/divider functions. The features of COPRO2 are listed below.

- **Multiplication:** Supports signed/unsigned multiplications.
(16 bits × 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 × 16 bits + 32 bits = 32 bits)
Can be executed in 1 cycle.
- **Division:** Supports signed/unsigned divisions.
(32 bits ÷ 32 bits = 32 bits with 32-bit reminder)
Can be executed in 17 to 20 cycles.
Overflow detection and division by zero processing are not supported.

Figure 20.1.1 shows the COPRO2 configuration.

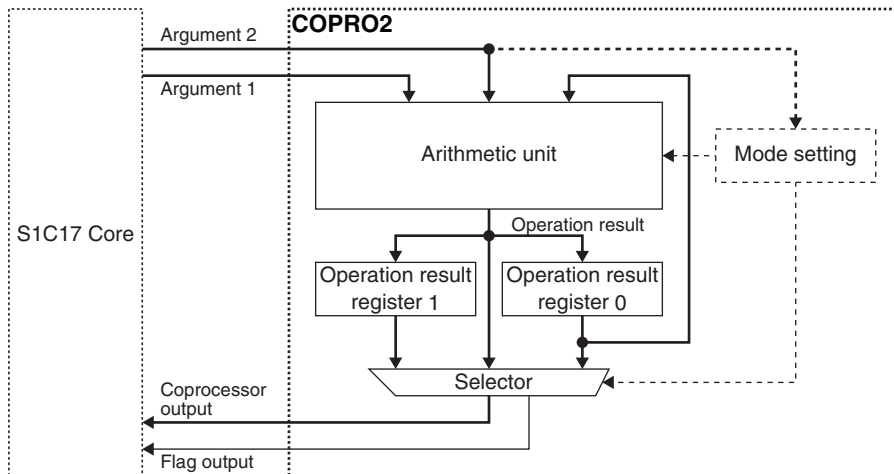


Figure 20.1.1 COPRO2 Configuration

20.2 Operation Mode and Output Mode

COPRO2 operates according to the operation mode specified by the application program. As listed in Table 20.2.1, 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)
```

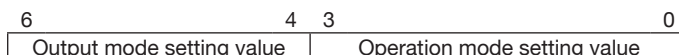


Figure 20.2.1 Mode Setting Register

Table 20.2.1 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

20.3 Multiplication

The multiplication function performs “ A (32 bits) = B (16 bits) \times 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 “1d.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 = $A[31:16]$).

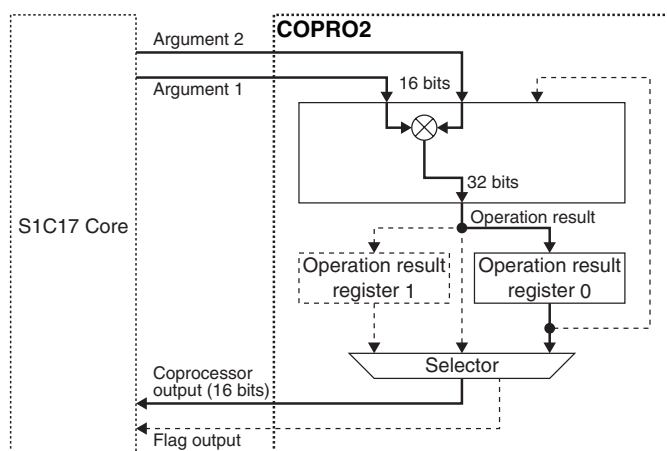


Figure 20.3.1 Data Path in Multiplication Mode

Table 20.3.1 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.
```

20.4 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).

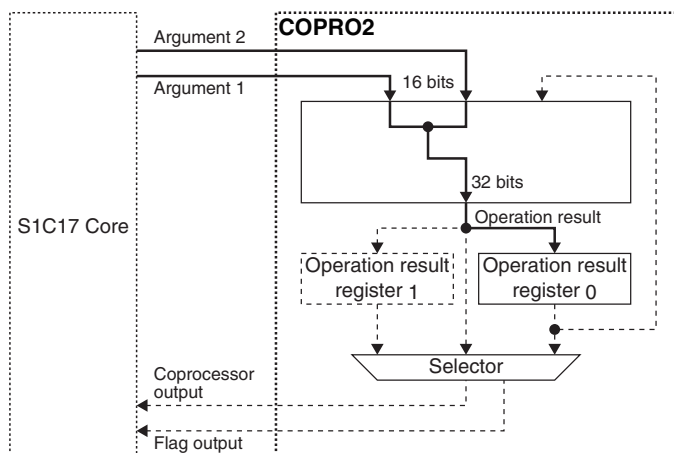


Figure 20.4.1 Data Path in Initialize Mode 2

Table 20.4.1 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)	res0[31:16] ← %rd	
	ld.cf %rd,imm7	res0[15:0] ← imm7/16	

res0: operation result register 0

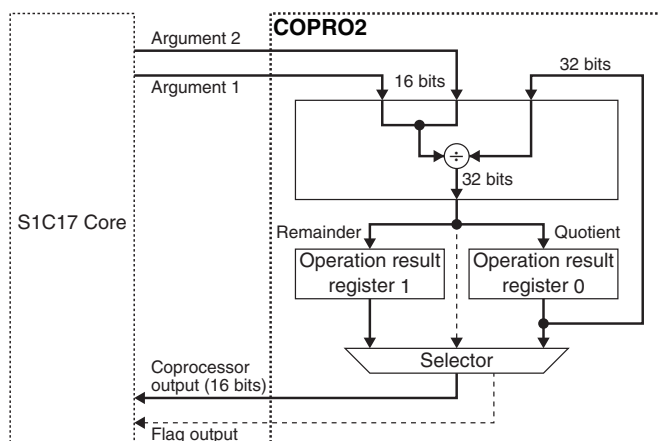


Figure 20.4.2 Data Path in Division Mode

Table 20.4.2 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.
```

20.5 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 = $A[31:16]$).

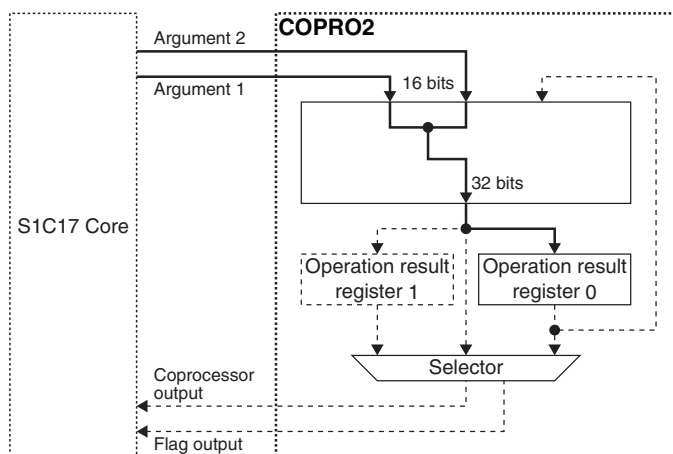


Figure 20.5.1 Data Path in Initialize Mode

Table 20.5.1 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

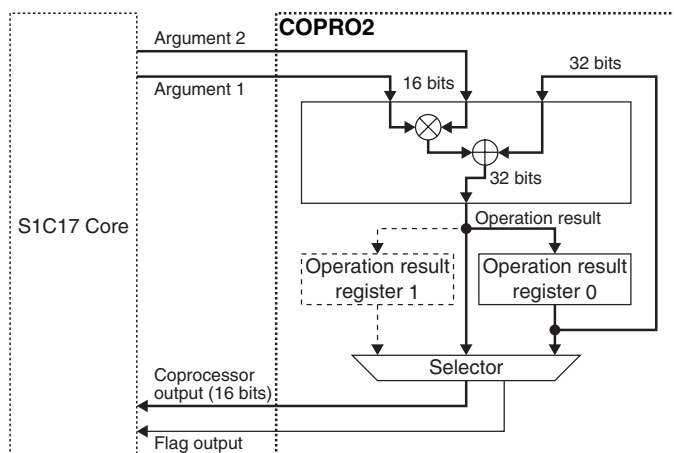


Figure 20.5.2 Data Path in MAC Mode

Table 20.5.2 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 Otherwise psr (CVZN) ← 0b0000	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]		
	(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 20.5.3 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.

20.6 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.

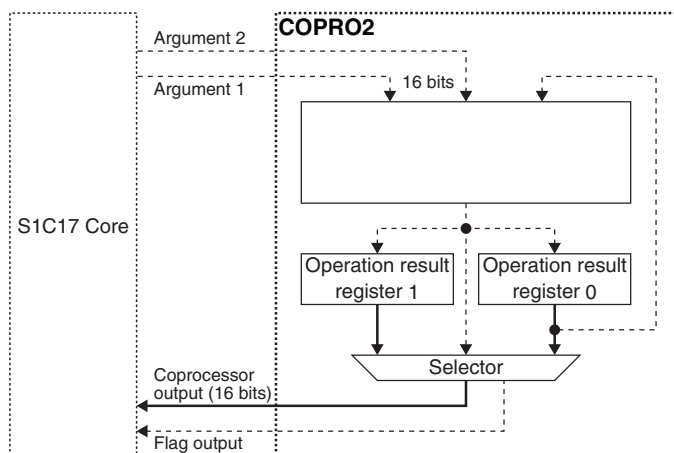


Figure 20.6.1 Data Path in Operation Result Read Mode

Table 20.6.1 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

21 Electrical Characteristics

21.1 Absolute Maximum Ratings

(V _{SS} = 0 V)				
Item	Symbol	Condition	Rated value	Unit
Power supply voltage	V _{DD}		-0.3 to 7.0	V
Flash programming voltage	V _{PP}		-0.3 to 8.0	V
Input voltage	V _I	P00-07, P10-17, P31-37, P40-42, PD0-D1, PD3-D4	-0.3 to 7.0	V
		P20-27, P30, #RESET	-0.3 to V _{DD} + 0.5	V
Output voltage	V _O	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D4	-0.3 to V _{DD} + 0.5	V
High level output current	I _{OH}	1 pin	-10	mA
		Total of all pins	-20	mA
Low level output current	I _{OL}	1 pin	10	mA
		Total of all pins	20	mA
Operating temperature	T _a		-40 to 85	°C
Storage temperature	T _{stg}		-65 to 125	°C

21.2 Recommended Operating Conditions

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Power supply voltage	V _{DD}	For normal operation	1.8	–	5.5	V
		For Flash	2.4	–	5.5	V
		When V _{PP} is supplied externally When V _{PP} is generated internally	2.4	–	5.5	V
Flash programming voltage	V _{PP}		7.3	7.5	7.7	V
OSC1 oscillator oscillation frequency	f _{OSC1}	Crystal oscillator	–	32.768	–	kHz
OSC3 oscillator oscillation frequency	f _{OSC3}	Crystal/ceramic oscillator	1	–	21	MHz
EXOSC external clock frequency	f _{EXOSC}	When supplied from an external oscillator	0.016	–	21	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
Gate capacitor for OSC1 oscillator	C _{G1}	When the crystal oscillator is used *1	0	–	25	pF
Drain capacitor for OSC1 oscillator	C _{D1}	When the crystal oscillator is used *1	–	0	–	pF
Gate capacitor for OSC3 oscillator	C _{G3}	When the crystal/ceramic oscillator is used *1	0	–	100	pF
Drain capacitor for OSC3 oscillator	C _{D3}	When the crystal/ceramic oscillator is used *1	0	–	100	pF
DSIO pull-up resistor	R _{DBG}	*2	–	10	–	kΩ
Capacitor between V _{SS} and V _{PP}	C _{VPP}		–	0.1	–	μF
Capacitor between V _{SS} and V _{REFA}	C _{VREFA}		–	0.1	–	μF

*1 The component values should be determined after performing matching evaluation of the resonator mounted on the printed circuit board actually used.

*2 R_{DBG} is not required when using the DSIO pin as a general-purpose I/O port.

21.3 Current Consumption

Unless otherwise specified: $V_{DD} = 1.8$ to 5.5 V, $V_{SS} = 0$ V, $T_a = 25^\circ\text{C}$, EXOSC = OFF, PWGVD1CTL.REGMOD[1:0] bits = 0x0 (automatic mode), FLASHCWAIT.RDWAIT[1:0] bits = 0x1 (2 cycles)

Item	Symbol	Condition	T_a	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I _{SLEEP}	IOSC = OFF, OSC1 = OFF, OSC3 = OFF	25°C	–	0.36	2	μA
			85°C	–	1.6	10	μA
Current consumption in HALT mode	I _{HALT1}	IOSC = ON, OSC1 = 32.768 kHz*1, OSC3 = OFF	–	–	40	60	μA
	I _{HALT2}	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = OFF	–	–	0.7	1.8	μA
		IOSC = OFF, OSC1 = 32 kHz*2, OSC3 = OFF	–	–	1.5	6.5	μA
	I _{HALT3}	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = 1 MHz (ceramic oscillator)*3	–	–	40	60	μA
		IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = 12 MHz (internal oscillator)*4	–	–	410	710	μA
Current consumption in RUN mode	I _{RUN1} *5	IOSC = ON, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = IOSC	–	–	120	200	μA
		IOSC = ON, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = IOSC FLASHCWAIT.RDWAIT[1:0] bits = 0x0 (1 cycle)	–	–	145	300	μA
		IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = OSC1	–	–	5	9	μA
	I _{RUN2} *5	IOSC = OFF, OSC1 = 32 kHz*2, OSC3 = OFF, SYSCLK = OSC1	–	–	5.5	12	μA
		IOSC = OFF, OSC1 = 32 kHz*2, OSC3 = OFF, SYSCLK = OSC1	–	–	5.5	12	μA
	I _{RUN3} *5	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = 1 MHz (ceramic oscillator)*3, SYSCLK = OSC3	–	–	160	320	μA
		IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = 12 MHz (internal oscillator)*4, SYSCLK = OSC3	–	–	1,800	2,800	μA

*1 OSC1 oscillator: CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N[1:0] bits = 0x0, CLGOSC1.CGI1[2:0] bits = 0x0, CLGOSC1.OSDEN bit = 0, C_{G1} = C_{D1} = 0 pF, Crystal resonator = C-002RX (manufactured by Seiko Epson Corporation, R₁ = 50 kΩ (Max.), C_L = 7 pF)

*2 OSC1 oscillator: CLGOSC1.OSC1SELCR bit = 1

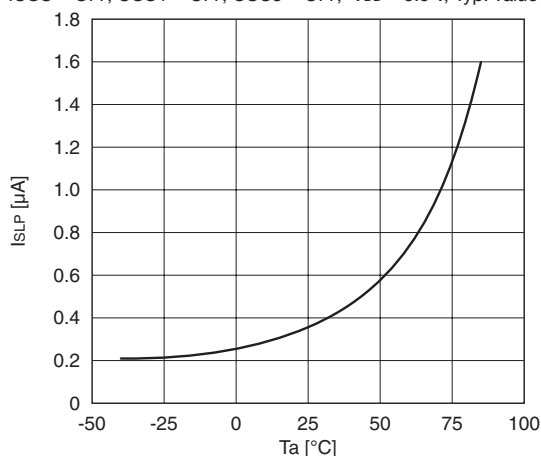
*3 OSC3 oscillator: CLGOSC3.OSC3MD bit = 1, CLGOSC3.OSC3INV[1:0] bits = 0x0, C_{G3} = C_{D3} = 100 pF, ceramic resonator = CS-BLA_J (manufactured by Murata Manufacturing Co., Ltd., 1 MHz)

*4 OSC3 oscillator: CLGOSC3.OSC3MD bit = 0, CLGOSC3.OSC3FQ[1:0] bits = 0x0

*5 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.

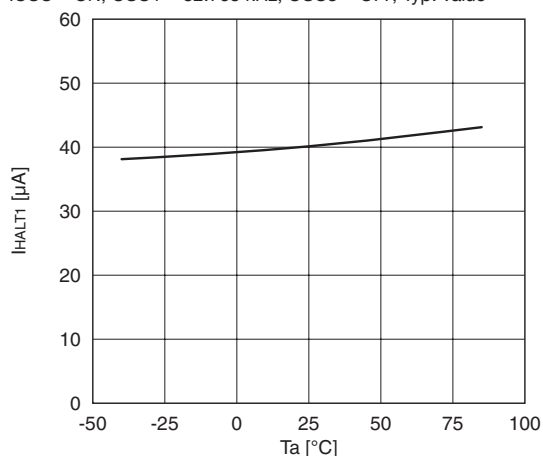
Current consumption-temperature characteristic in SLEEP mode

IOSC = OFF, OSC1 = OFF, OSC3 = OFF, $V_{DD} = 5.5$ V, Typ. value



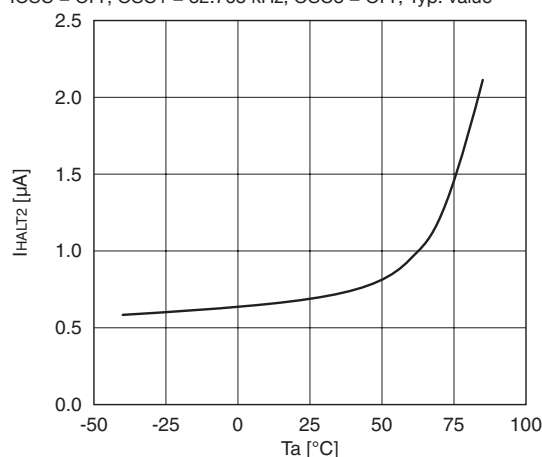
Current consumption-temperature characteristic in HALT mode (IOSC operation)

IOSC = ON, OSC1 = 32.768 kHz, OSC3 = OFF, Typ. value

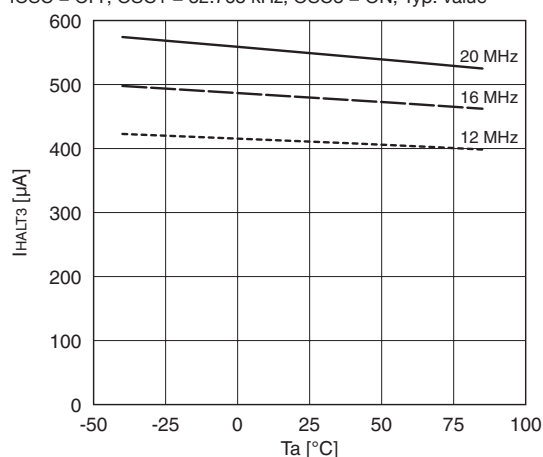


Current consumption-temperature characteristic in HALT mode (OSC1 operation)

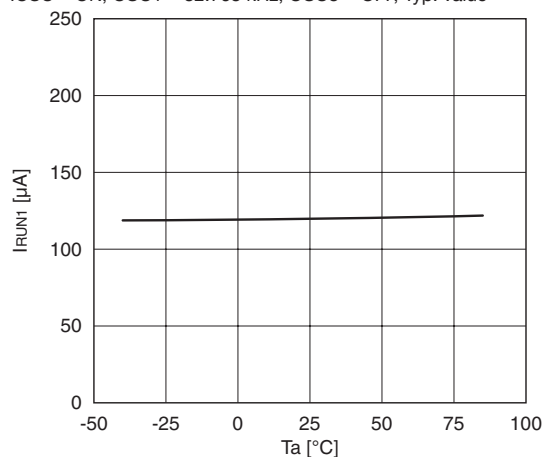
IOSC = OFF, OSC1 = 32.768 kHz, OSC3 = OFF, Typ. value


Current consumption-temperature characteristic in HALT mode (OSC3 operation)

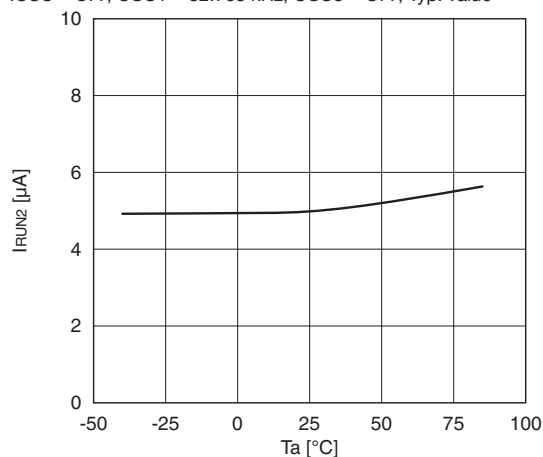
IOSC = OFF, OSC1 = 32.768 kHz, OSC3 = ON, Typ. value

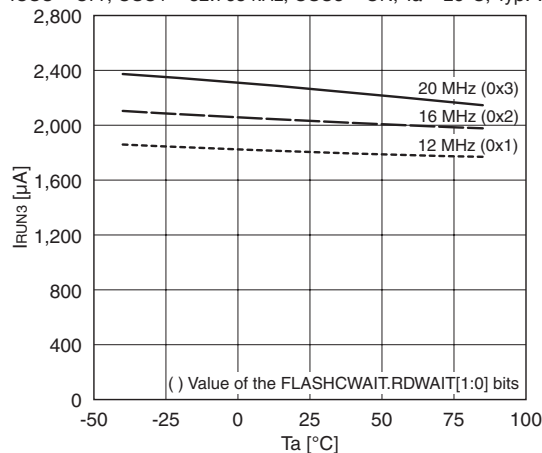

Current consumption-temperature characteristic in RUN mode (IOSC operation)

IOSC = ON, OSC1 = 32.768 kHz, OSC3 = OFF, Typ. value


Current consumption-temperature characteristic in RUN mode (OSC1 operation)

IOSC = OFF, OSC1 = 32.768 kHz, OSC3 = OFF, Typ. value


Current consumption-temperature characteristic in RUN mode (OSC3 operation)

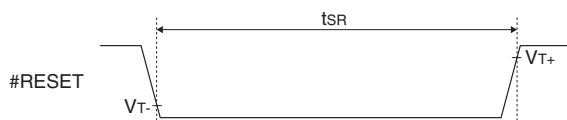
 IOSC = OFF, OSC1 = 32.768 kHz, OSC3 = ON, $T_a = 25^{\circ}C$, Typ. value


21.4 System Reset Controller (SRC) Characteristics

#RESET pin characteristics

Unless otherwise specified: $V_{DD} = 1.8$ to 5.5 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

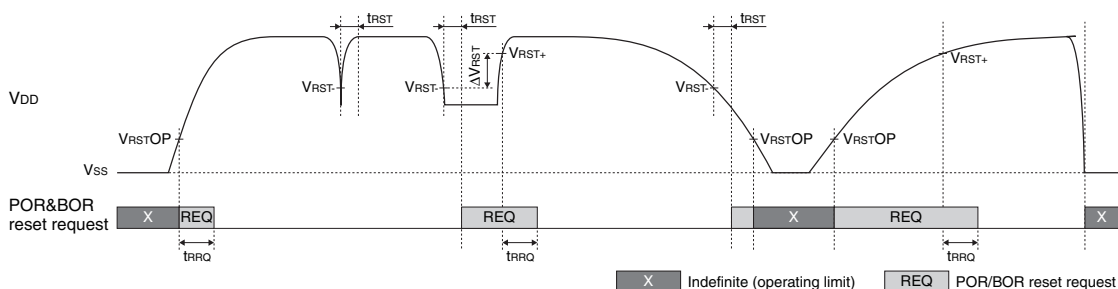
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}		$0.5 \times V_{DD}$	—	$0.8 \times V_{DD}$	V
Low level Schmitt input threshold voltage	V_{T-}		$0.2 \times V_{DD}$	—	$0.5 \times V_{DD}$	V
Schmitt input hysteresis voltage	ΔV_T		180	—	—	mV
Input pull-up resistance	R_{IN}		100	230	500	k Ω
Pin capacitance	C_{IN}		—	—	15	pF
Reset Low pulse width	t_{SR}		5	—	—	μs



POR/BOR characteristics

Unless otherwise specified: $V_{DD} = 1.8$ to 5.5 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
POR/BOR canceling voltage	V_{RST+}		1.41	—	1.75	V
POR/BOR detection voltage	V_{RST-}		1.25	—	1.55	V
POR/BOR hysteresis voltage	ΔV_{RST}		40	60	—	mV
POR/BOR detection response time	t_{RST}		—	—	20	μs
POR/BOR operating limit voltage	V_{RSTOP}		—	0.5	0.95	V
POR/BOR 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: $V_{DD} = 1.8$ to 5.5 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.

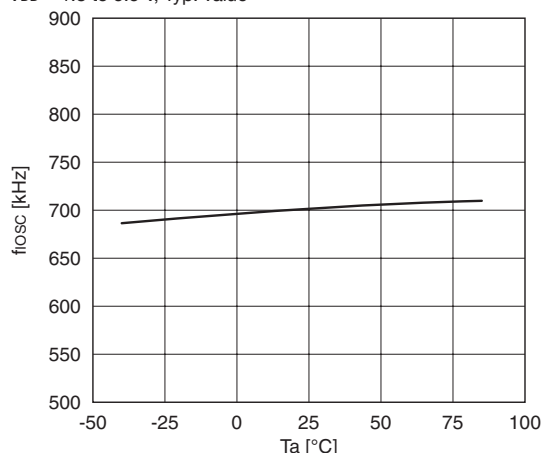
21.5 Clock Generator (CLG) Characteristics

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: $V_{DD} = 1.8$ to 5.5 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

Item	Symbol	Condition	T_a	Min.	Typ.	Max.	Unit
Oscillation start time	t_{stal}			—	—	3	μs
Oscillation frequency	f_{osc}		25°C	679	700	721	kHz
			-40 to 85°C	651	700	749	kHz

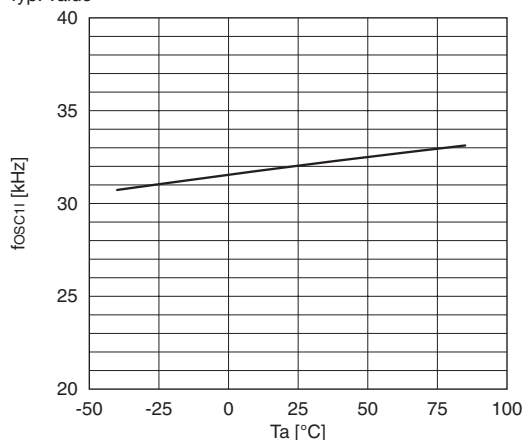
IOSC oscillation frequency-temperature characteristicV_{DD} = 1.8 to 5.5 V, Typ. value**OSC1 oscillator circuit characteristics**Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Crystal oscillator oscillation start time*1	t _{sta1C}	CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N[1:0] bits = 0x1, CLGOSC1.INV1B[1:0] bits = 0x2, CLGOSC1.OSC1BUP bit = 1	–	–	3	s
Crystal oscillator internal gate capacitance	C _{GI1C}	CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x0	–	12	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x1	–	14	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x2	–	16	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x3	–	18	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x4	–	19	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x5	–	21	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x6	–	23	–	pF
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.CGI1[2:0] bits = 0x7	–	24	–	pF
Crystal oscillator internal drain capacitance	C _{DI1C}	CLGOSC1.OSC1SELCR bit = 0,	–	6	–	pF
Crystal oscillator oscillator circuit current - oscillation inverter drivability ratio *1	I _{OSC1C}	CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N/INV1B[1:0] bits = 0x0	–	70	–	%
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N/INV1B[1:0] bits = 0x1 (reference)	–	100	–	%
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N/INV1B[1:0] bits = 0x2	–	130	–	%
		CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.INV1N/INV1B[1:0] bits = 0x3	–	300	–	%
Crystal oscillator oscillation stop detector current	I _{OSD1C}	CLGOSC1.OSC1SELCR bit = 0, CLGOSC1.OSDEN bit = 1	–	0.025	0.1	μA
Internal oscillator oscillation start time	t _{sta1I}	CLGOSC1.OSC1SELCR bit = 1	–	–	100	μs
Internal oscillator oscillation frequency	f _{OSC1I}	CLGOSC1.OSC1SELCR bit = 1	31.04	32	32.96	kHz

*1 CLGOSC1.CGI1[2:0] bits = 0x0, Crystal resonator = C-002RX (manufactured by Seiko Epson Corporation, R₁ = 50 kΩ (Max.), C_L = 7 pF)

OSC1 internal oscillation frequency-temperature characteristic

Typ. value

**OSC3 oscillator circuit characteristics**

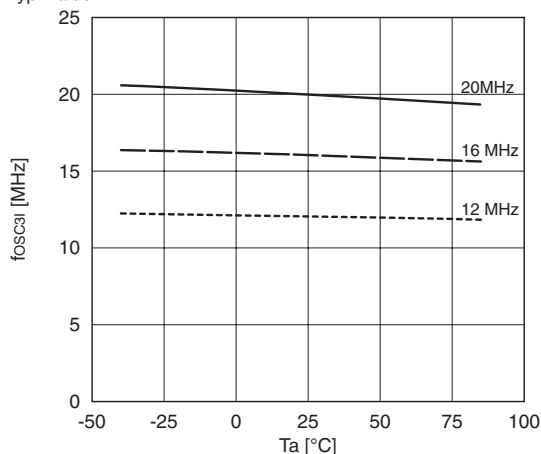
Unless otherwise specified: VDD = 1.8 to 5.5 V, VSS = 0 V, Ta = 25°C

Item	Symbol	Condition	Ta	Min.	Typ.	Max.	Unit
Internal oscillator oscillation start time	tsta3l	CLGOSC3.OSC3MD bit = 0		–	–	3	μs
Internal oscillator oscillation frequency	fosc3l	CLGOSC3.OSC3MD bit = 0, CLGOSC3.OSC3FQ[1:0] bits = 0x2	25°C	18.00	20	22.00	MHz
			-40 to 85°C	17.40	20	22.60	MHz
		CLGOSC3.OSC3MD bit = 0, CLGOSC3.OSC3FQ[1:0] bits = 0x1	25°C	15.20	16	16.8	MHz
			-40 to 85°C	14.88	16	17.12	MHz
		CLGOSC3.OSC3MD bit = 0, CLGOSC3.OSC3FQ[1:0] bits = 0x0	10 to 40°C	11.88	12	12.12	MHz
			-10 to 60°C	11.76	12	12.24	MHz
Crystal/ceramic oscillator oscillation start time*1	tsta3C	CLGOSC3.OSC3MD bit = 1, CLGOSC3.OSC3INV[1:0] bits = 0x0		–	–	10	ms
Crystal/ceramic oscillator internal gate capacitance	Cgi3C	CLGOSC3.OSC3MD bit = 1		–	8	–	pF
Crystal/ceramic oscillator internal drain capacitance	Cdi3C	CLGOSC3.OSC3MD bit = 1		–	8	–	pF
Crystal/ceramic oscillator circuit current - oscillation inverter drivability ratio	Iosc3C	CLGOSC3.OSC3MD bit = 1, CLGOSC3.OSC3INV[1:0] bits = 0x0		–	50	–	%
		CLGOSC3.OSC3MD bit = 1, CLGOSC3.OSC3INV[1:0] bits = 0x1 (reference)		–	100	–	%
		CLGOSC3.OSC3MD bit = 1, CLGOSC3.OSC3INV[1:0] bits = 0x2		–	120	–	%
		CLGOSC3.OSC3MD bit = 1, CLGOSC3.OSC3INV[1:0] bits = 0x3		–	190	–	%
				–	–	–	–

*2 Ceramic resonator = CSBLA_J (manufactured by Murata Manufacturing Co., Ltd., 1 MHz), CG3 = CD3 = 100 pF

OSC3 internal oscillation frequency-temperature characteristic

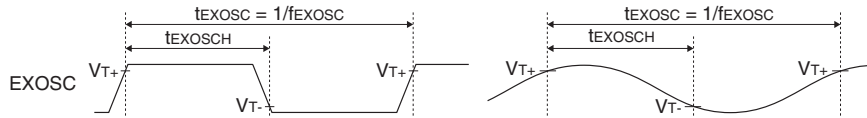
Typ. value



EXOSC external clock input characteristics

Unless otherwise specified: $V_{DD} = 1.8$ to 5.5 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}	$t_{EXOSCD} = t_{EXOSCH}/t_{EXOSC}$	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		180	—	—	mV

**21.6 Flash Memory Characteristics**

Unless otherwise specified: $V_{DD} = 2.4$ to 5.5 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

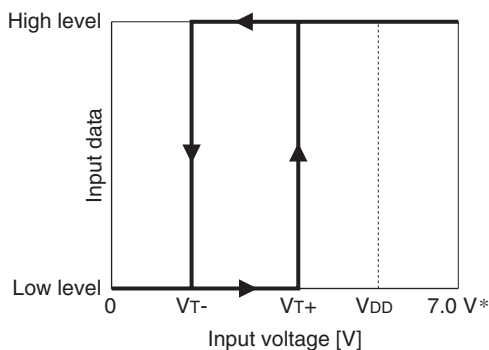
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Programming count *1	C_{FEP}	Programmed data is guaranteed to be retained for 10 years.	1,000	—	—	times

*1 Assumed that Erasing + Programming as count of 1. The count includes programming in the factory for shipment with ROM data programmed.

21.7 Input/Output Port (PPORT) Characteristics

Unless otherwise specified: $V_{DD} = 1.8$ to 5.5 V, $V_{SS} = 0$ V, $T_a = -40$ to 85°C

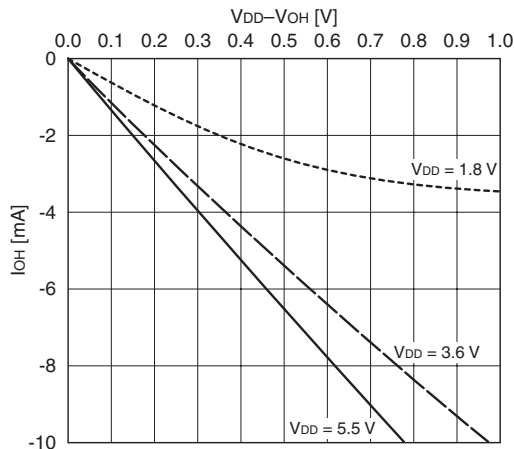
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D1, PD3-D4	$0.5 \times V_{DD}$	—	$0.8 \times V_{DD}$	V
Low level Schmitt input threshold voltage	V_{T-}	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D1, PD3-D4	$0.2 \times V_{DD}$	—	$0.5 \times V_{DD}$	V
Schmitt input hysteresis voltage	ΔV_T	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D1, PD3-D4	180	—	—	mV
High level output current	I_{OH}	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D4, $V_{OH} = 0.9 \times V_{DD}$	—	—	-0.5	mA
Low level output current	I_{OL}	P00-07, P10-17, P20-27, P30-37, P40-2, PD0-D4, $V_{OL} = 0.1 \times V_{DD}$	0.5	—	—	mA
Leakage current	I_{LEAK}	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D4	-150	—	150	nA
Input pull-up resistance	R_{INU}	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D1, PD3-D4	100	200	500	k Ω
Input pull-down resistance	R_{IND}	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D1, PD3-D4	100	200	500	k Ω
Pin capacitance	C_{IN}	P00-07, P10-17, P20-27, P30-37, P40-42, PD0-D1, PD3-D4	—	—	15	pF



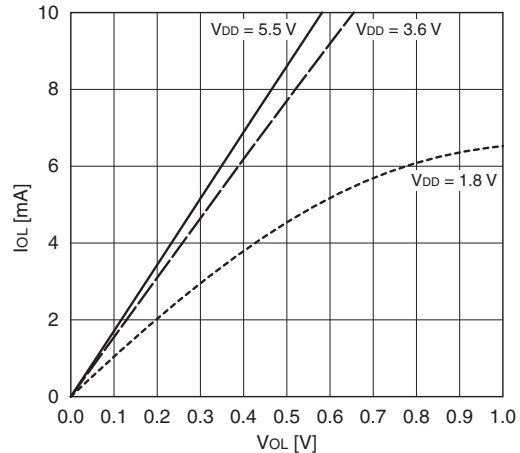
(* For over voltage tolerant fail-safe type port)

High-level output current characteristic

Ta = 85°C, Max. value

**Low-level output current characteristic**

Ta = 85°C, Min. value

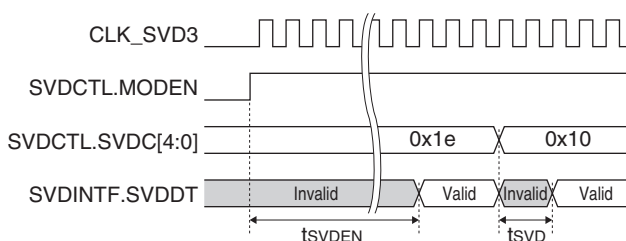
**21.8 Supply Voltage Detector (SVD3) Characteristics**

Unless otherwise specified: VDD = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EXSVD pin input voltage range	VEXSVD		0	—	5.5	V
EXSVD input impedance	REXSVD	SVDCTL.SVDC[4:0] bits = 0x00	253	279	305	kΩ
		SVDCTL.SVDC[4:0] bits = 0x01	274	302	330	kΩ
		SVDCTL.SVDC[4:0] bits = 0x02	317	348	380	kΩ
		SVDCTL.SVDC[4:0] bits = 0x03	338	371	405	kΩ
		SVDCTL.SVDC[4:0] bits = 0x04	380	418	456	kΩ
		SVDCTL.SVDC[4:0] bits = 0x05	421	464	507	kΩ
		SVDCTL.SVDC[4:0] bits = 0x06	443	487	531	kΩ
		SVDCTL.SVDC[4:0] bits = 0x07	464	511	557	kΩ
		SVDCTL.SVDC[4:0] bits = 0x08	486	534	581	kΩ
		SVDCTL.SVDC[4:0] bits = 0x09	507	557	607	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0a	528	580	631	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0b	551	603	655	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0c	571	626	682	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0d	593	649	705	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0e	616	672	727	kΩ
		SVDCTL.SVDC[4:0] bits = 0x0f	635	695	754	kΩ
		SVDCTL.SVDC[4:0] bits = 0x10	658	718	777	kΩ
		SVDCTL.SVDC[4:0] bits = 0x11	679	741	804	kΩ
		SVDCTL.SVDC[4:0] bits = 0x12	698	765	833	kΩ
		SVDCTL.SVDC[4:0] bits = 0x13	739	812	885	kΩ
		SVDCTL.SVDC[4:0] bits = 0x14	761	834	908	kΩ
		SVDCTL.SVDC[4:0] bits = 0x15	804	880	955	kΩ
		SVDCTL.SVDC[4:0] bits = 0x16	842	929	1,016	kΩ
		SVDCTL.SVDC[4:0] bits = 0x17	878	948	1,019	kΩ
		SVDCTL.SVDC[4:0] bits = 0x18	893	972	1,052	kΩ
		SVDCTL.SVDC[4:0] bits = 0x19	922	993	1,064	kΩ
		SVDCTL.SVDC[4:0] bits = 0x1a	963	1,041	1,119	kΩ
		SVDCTL.SVDC[4:0] bits = 0x1b	982	1,063	1,145	kΩ
		SVDCTL.SVDC[4:0] bits = 0x1c	1,001	1,086	1,171	kΩ
		SVDCTL.SVDC[4:0] bits = 0x1d	1,022	1,110	1,198	kΩ
		SVDCTL.SVDC[4:0] bits = 0x1e	1,054	1,129	1,204	kΩ
		SVDCTL.SVDC[4:0] bits = 0x1f	1,072	1,154	1,237	kΩ
EXSVD detection voltage	VSVD_EXT	SVDCTL.SVDC[4:0] bits = 0x0	1.17	1.2	1.23	V
		SVDCTL.SVDC[4:0] bits = 0x1	1.27	1.3	1.33	V
		SVDCTL.SVDC[4:0] bits = 0x2	1.46	1.5	1.54	V
		SVDCTL.SVDC[4:0] bits = 0x3	1.56	1.6	1.64	V
		SVDCTL.SVDC[4:0] bits = 0x04	1.76	1.8	1.85	V
		SVDCTL.SVDC[4:0] bits = 0x05	1.95	2.0	2.05	V
		SVDCTL.SVDC[4:0] bits = 0x06	2.05	2.1	2.15	V
		SVDCTL.SVDC[4:0] bits = 0x07	2.15	2.2	2.26	V
		SVDCTL.SVDC[4:0] bits = 0x08	2.24	2.3	2.36	V
		SVDCTL.SVDC[4:0] bits = 0x09	2.34	2.4	2.46	V

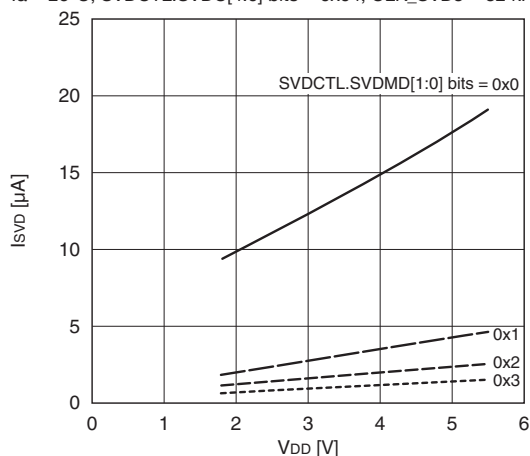
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EXSVD detection voltage	V _{SVD_EXT}	SVDCTL.SVDC[4:0] bits = 0x0a	2.44	2.5	2.56	V
		SVDCTL.SVDC[4:0] bits = 0x0b	2.54	2.6	2.67	V
		SVDCTL.SVDC[4:0] bits = 0x0c	2.63	2.7	2.77	V
		SVDCTL.SVDC[4:0] bits = 0x0d	2.73	2.8	2.87	V
		SVDCTL.SVDC[4:0] bits = 0x0e	2.83	2.9	2.97	V
		SVDCTL.SVDC[4:0] bits = 0x0f	2.93	3.0	3.08	V
		SVDCTL.SVDC[4:0] bits = 0x10	3.02	3.1	3.18	V
		SVDCTL.SVDC[4:0] bits = 0x11	3.12	3.2	3.28	V
		SVDCTL.SVDC[4:0] bits = 0x12	3.22	3.3	3.38	V
		SVDCTL.SVDC[4:0] bits = 0x13	3.41	3.5	3.59	V
		SVDCTL.SVDC[4:0] bits = 0x14	3.51	3.6	3.69	V
		SVDCTL.SVDC[4:0] bits = 0x15	3.71	3.8	3.90	V
		SVDCTL.SVDC[4:0] bits = 0x16	3.90	4.0	4.10	V
		SVDCTL.SVDC[4:0] bits = 0x17	4.00	4.1	4.20	V
		SVDCTL.SVDC[4:0] bits = 0x18	4.10	4.2	4.31	V
		SVDCTL.SVDC[4:0] bits = 0x19	4.19	4.3	4.41	V
		SVDCTL.SVDC[4:0] bits = 0x1a	4.39	4.5	4.61	V
		SVDCTL.SVDC[4:0] bits = 0x1b	4.49	4.6	4.72	V
		SVDCTL.SVDC[4:0] bits = 0x1c	4.58	4.7	4.82	V
		SVDCTL.SVDC[4:0] bits = 0x1d	4.68	4.8	4.92	V
		SVDCTL.SVDC[4:0] bits = 0x1e	4.78	4.9	5.02	V
		SVDCTL.SVDC[4:0] bits = 0x1f	4.88	5.0	5.13	V
SVD detection voltage	V _{SVD}	SVDCTL.SVDC[4:0] bits = 0x04	1.76	1.8	1.85	V
		SVDCTL.SVDC[4:0] bits = 0x05	1.95	2.0	2.05	V
		SVDCTL.SVDC[4:0] bits = 0x06	2.05	2.1	2.15	V
		SVDCTL.SVDC[4:0] bits = 0x07	2.15	2.2	2.26	V
		SVDCTL.SVDC[4:0] bits = 0x08	2.24	2.3	2.36	V
		SVDCTL.SVDC[4:0] bits = 0x09	2.34	2.4	2.46	V
		SVDCTL.SVDC[4:0] bits = 0x0a	2.44	2.5	2.56	V
		SVDCTL.SVDC[4:0] bits = 0x0b	2.54	2.6	2.67	V
		SVDCTL.SVDC[4:0] bits = 0x0c	2.63	2.7	2.77	V
		SVDCTL.SVDC[4:0] bits = 0x0d	2.73	2.8	2.87	V
		SVDCTL.SVDC[4:0] bits = 0x0e	2.83	2.9	2.97	V
		SVDCTL.SVDC[4:0] bits = 0x0f	2.93	3.0	3.08	V
		SVDCTL.SVDC[4:0] bits = 0x10	3.02	3.1	3.18	V
		SVDCTL.SVDC[4:0] bits = 0x11	3.12	3.2	3.28	V
		SVDCTL.SVDC[4:0] bits = 0x12	3.22	3.3	3.38	V
		SVDCTL.SVDC[4:0] bits = 0x13	3.41	3.5	3.59	V
		SVDCTL.SVDC[4:0] bits = 0x14	3.51	3.6	3.69	V
		SVDCTL.SVDC[4:0] bits = 0x15	3.71	3.8	3.90	V
		SVDCTL.SVDC[4:0] bits = 0x16	3.90	4.0	4.10	V
		SVDCTL.SVDC[4:0] bits = 0x17	4.00	4.1	4.20	V
		SVDCTL.SVDC[4:0] bits = 0x18	4.10	4.2	4.31	V
		SVDCTL.SVDC[4:0] bits = 0x19	4.19	4.3	4.41	V
		SVDCTL.SVDC[4:0] bits = 0x1a	4.39	4.5	4.61	V
		SVDCTL.SVDC[4:0] bits = 0x1b	4.49	4.6	4.72	V
		SVDCTL.SVDC[4:0] bits = 0x1c	4.58	4.7	4.82	V
		SVDCTL.SVDC[4:0] bits = 0x1d	4.68	4.8	4.92	V
		SVDCTL.SVDC[4:0] bits = 0x1e	4.78	4.9	5.02	V
		SVDCTL.SVDC[4:0] bits = 0x1f	4.88	5.0	5.13	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 = 0x04, CLK_SVD3 = 32 kHz, Ta = 25°C	—	19	35	μA
		SVDCTL.SVDMd[1:0] bits = 0x1, SVDCTL.SVDC[4:0] bits = 0x04, CLK_SVD3 = 32 kHz, Ta = 25°C	—	4.7	7.7	μA
		SVDCTL.SVDMd[1:0] bits = 0x2, SVDCTL.SVDC[4:0] bits = 0x04, CLK_SVD3 = 32 kHz, Ta = 25°C	—	2.5	4.1	μA
		SVDCTL.SVDMd[1:0] bits = 0x3, SVDCTL.SVDC[4:0] bits = 0x04, CLK_SVD3 = 32 kHz, Ta = 25°C	—	1.5	2.4	μA

*1 If CLK_SVD3 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 = 0x04, CLK_SVD3 = 32 kHz, Typ. value



21.9 UART (UART3) Characteristics

Unless otherwise specified: VDD = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Transfer baud rate	UBRT1	Normal mode	150	–	921,600	bps
	UBRT2	IrDA mode	150	–	115,200	bps

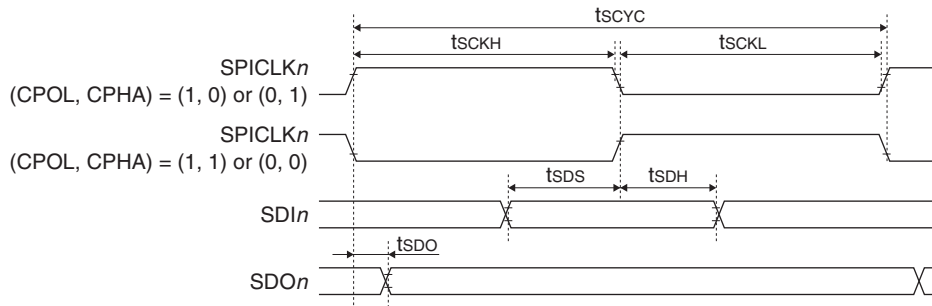
21.10 Synchronous Serial Interface (SPIA) Characteristics

Unless otherwise specified: VDD = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to 85°C

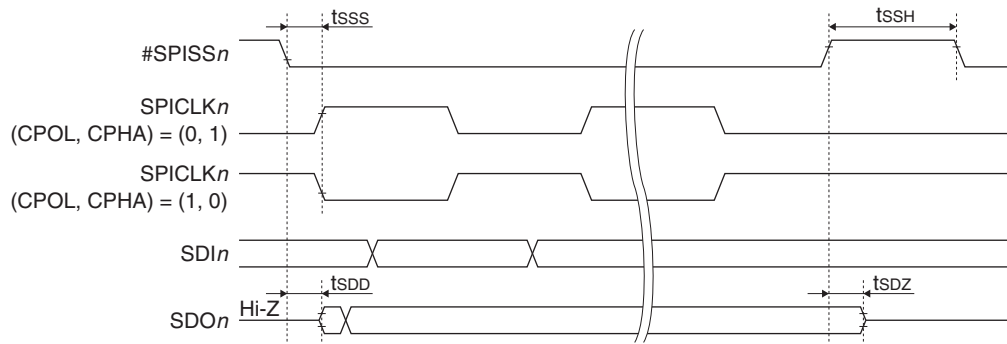
Item	Symbol	Condition	VDD	Min.	Typ.	Max.	Unit
SPICLK _n cycle time	tSCYC		4.5 to 5.5 V	250	–	–	ns
			1.8 to 4.5 V	500	–	–	ns
SPICLK _n High pulse width	tSCKH		4.5 to 5.5 V	100	–	–	ns
			1.8 to 4.5 V	200	–	–	ns
SPICLK _n Low pulse width	tSCKL		4.5 to 5.5 V	100	–	–	ns
			1.8 to 4.5 V	200	–	–	ns
SDIn setup time	tSDS		4.5 to 5.5 V	50	–	–	ns
			1.8 to 4.5 V	80	–	–	ns
SDIn hold time	tSDH		4.5 to 5.5 V	20	–	–	ns
			1.8 to 4.5 V	30	–	–	ns
SDOn output delay time	tSDO	CL = 30 pF *1	4.5 to 5.5 V	–	–	60	ns
			1.8 to 4.5 V	–	–	90	ns
#SPISS _n setup time	tSSS			80	–	–	ns
#SPISS _n High pulse width	tSSH			100	–	–	ns
SDOn output start time	tSDD	CL = 30 pF *1		–	–	90	ns
SDOn output stop time	tSDZ	CL = 30 pF *1		–	–	80	ns

*1 CL = Pin load

Master and slave modes



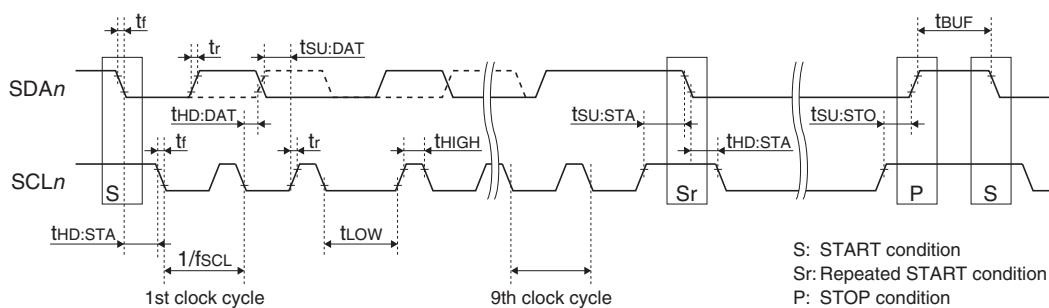
Slave mode

21.11 I²C (I2C) Characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, T_a = -40 to 85°C

Item	Symbol	Condition	Standard mode			Fast mode			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
SCL _n frequency	f _{SCL}		0	—	100	0	—	400	kHz
Hold time (repeated) START condition *	t _{HD:STA}		4.0	—	—	0.6	—	—	μs
SCL _n Low pulse width	t _{LOW}		4.7	—	—	1.3	—	—	μs
SCL _n 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
SDA _n , SCL _n rise time	t _r		—	—	1,000	—	—	300	ns
SDA _n , SCL _n 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

21.12 R/F Converter (RFC) Characteristics

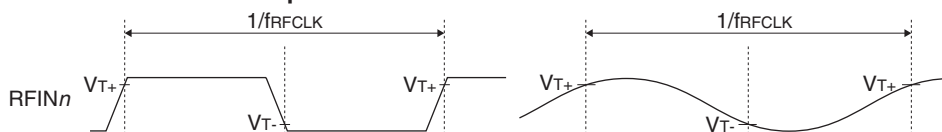
R/F converter characteristics change depending on conditions (board pattern, components used, etc.). Use these characteristic values as a reference and perform evaluation using the actual printed circuit board.

Unless otherwise specified: $V_{DD} = 1.8$ to 5.5 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/sensor oscillation frequency IC deviation	$\Delta f_{RFCLK}/\Delta IC$	$T_a = 25^\circ\text{C}$ *1		-40	—	40	%
Reference resistor/resistive sensor resistance	R_{REF}, R_{SEN}			10	—	—	k Ω
Reference capacitance	C_{REF}			100	—	—	pF
Time base counter clock frequency	f_{TCLK}			—	—	21	MHz
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			180	—	—	mV
R/F converter operating current	I_{RFC}	$C_{REF} = 1,000$ pF, $R_{REF}/R_{SEN} = 100$ k Ω , $T_a = 25^\circ\text{C}$	3.6 to 5.5 V	—	130	200	μA
			1.8 to 3.6 V	—	60	85	μA

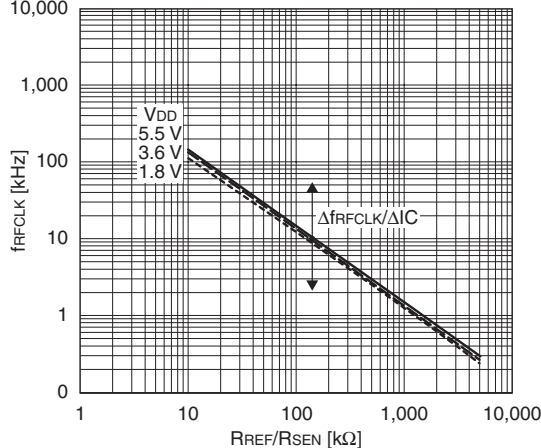
*1 In this characteristic, unevenness between production lots, and variations in measurement board, resistances and capacitances are taken into account.

Waveforms for external clock input mode



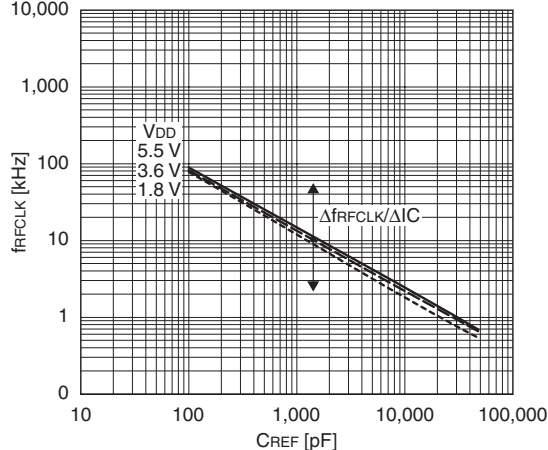
RFC reference/sensor oscillation frequency-resistance characteristic

$C_{REF} = 1,000$ pF, $T_a = 25^\circ\text{C}$, Typ. value
10,000



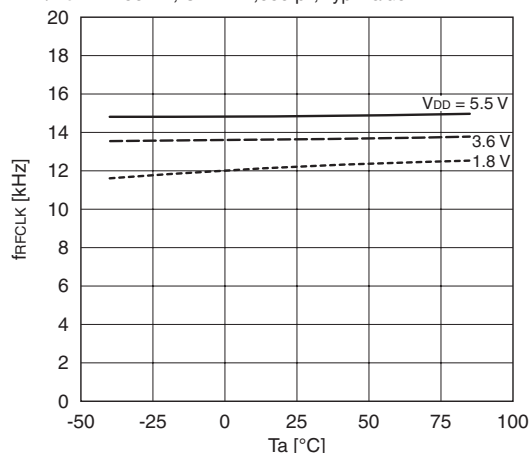
RFC reference/sensor oscillation frequency-capacitance characteristic

$R_{REF}/R_{SEN} = 100$ k Ω , $T_a = 25^\circ\text{C}$, Typ. value
10,000



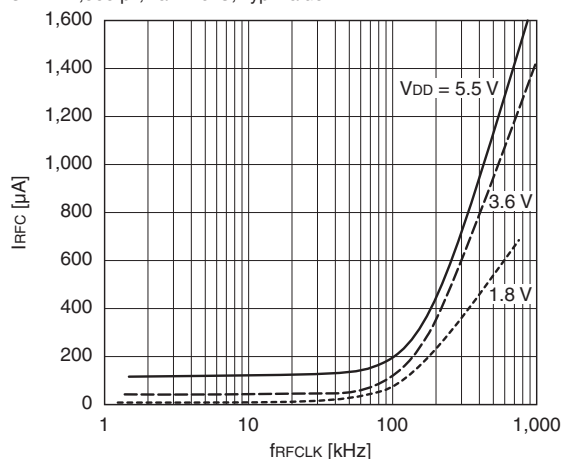
RFC reference/sensor oscillation frequency-temperature characteristic

$R_{REF}/R_{SEN} = 100\text{ k}\Omega$, $C_{REF} = 1,000\text{ pF}$, Typ. value



RFC reference/sensor oscillation current consumption-frequency characteristic

$C_{REF} = 1,000\text{ pF}$, $T_a = 25^\circ\text{C}$, Typ. value



21.13 12-bit A/D Converter (ADC12A) Characteristics

Unless otherwise specified: $V_{DD} = 1.8$ to 5.5 V , $V_{REFAn} = 1.8$ to 5.5 V , $V_{SS} = 0\text{ V}$, $T_a = -40$ to 85°C ,

ADC12_nTRG.SMPCLK[2:0] bits = 0x3 (7cycles)

Item	Symbol	Condition	V _{DD}	Min.	Typ.	Max.	Unit
VREFAn voltage range	VREFA			1.8	–	V _{DD}	V
A/D conversion clock frequency	fCLK_ADC12A			16	–	2,200	kHz
Sampling rate *1	fSMP			–	–	100	ksps
Integral nonlinearity *2	INL	$V_{DD} = V_{REFAn}$ *3		–	–	±3	LSB
Differential nonlinearity	DNL	$V_{DD} = V_{REFAn}$ *3		–	–	±3	LSB
Zero-scale error	ZSE	$V_{DD} = V_{REFAn}$ *3		–	–	±5	LSB
Full-scale error	FSE	$V_{DD} = V_{REFAn}$ *3		–	–	±5	LSB
Analog input resistance	RADIN			–	–	4	kΩ
Analog input capacitance	CADIN			–	–	30	pF
A/D converter circuit current	I _{ADC}	ADC12_nCFG.VRANGE[1:0] bits = 0x3, $V_{DD} = V_{REFA}$, $ADIN = V_{REFA}/2$, $f_{SMP} = 100\text{ ksps}$, $T_a = 25^\circ\text{C}$	3.6 V	–	380	670	μA
		ADC12_nCFG.VRANGE[1:0] bits = 0x2, $V_{DD} = V_{REFA}$, $ADIN = V_{REFA}/2$, $f_{SMP} = 100\text{ ksps}$, $T_a = 25^\circ\text{C}$	4.8 V	–	230	390	μA
		ADC12_nCFG.VRANGE[1:0] bits = 0x1, $V_{DD} = V_{REFA}$, $ADIN = V_{REFA}/2$, $f_{SMP} = 100\text{ ksps}$, $T_a = 25^\circ\text{C}$	5.5 V	–	210	350	μA

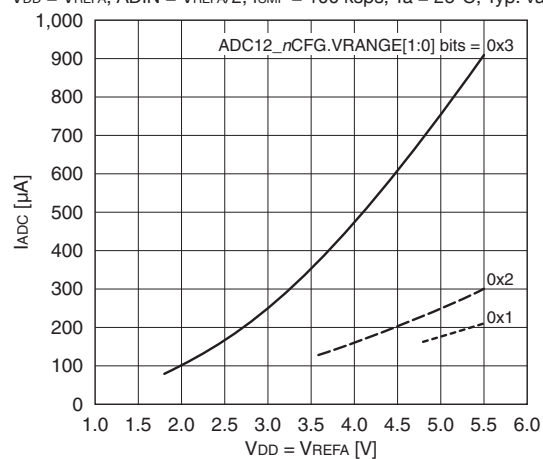
*1 The Max. value is the value when the A/D conversion clock frequency fCLK_ADC12A = 2,000 kHz.

*2 Integral nonlinearity is measured at the end point line.

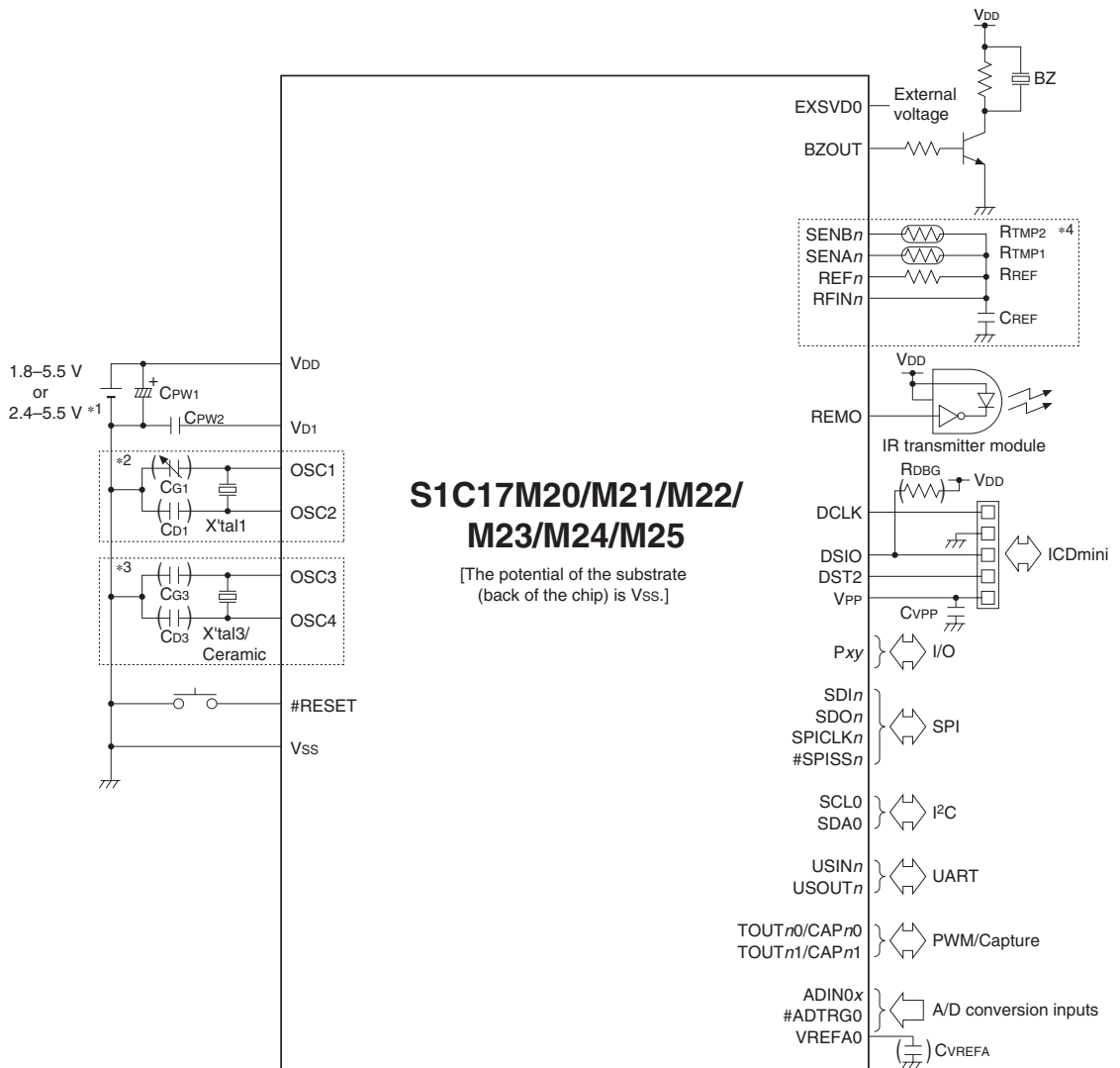
*3 The error will be increased according to the potential difference between V_{DD} and V_{REFAn} .

A/D converter current consumption-power supply voltage characteristic

$V_{DD} = V_{REFA}$, $ADIN = V_{REFA}/2$, $f_{SMP} = 100$ ksp/s, $T_a = 25^\circ\text{C}$, Typ. value



22 Basic External Connection Diagram



*1: For Flash programming

*2: When the OSC1 crystal oscillator is used (except for the S1C17M20/M23 (24-pin package))

*3: When the OSC3 crystal/ceramic oscillator is used (except for the S1C17M20/M23 (24-pin package))

*4: When the R/F converter is used (available in the S1C17M22/M25)

(): Do not mount components if unnecessary.

Sample external components

Symbol	Name	Recommended components
X'tal1	32 kHz crystal resonator	C-002RX (R ₁ = 50 kΩ (Max.), C _L = 7 pF) manufactured by Seiko Epson Corporation
C _{G1}	OSC1 gate capacitor	Trimmer capacitor or ceramic capacitor
C _{D1}	OSC1 drain capacitor	Ceramic capacitor
X'tal3	Crystal resonator	CA-301 (4 MHz) manufactured by Seiko Epson Corporation
Ceramic	Ceramic resonator	CSBLA_J (1 MHz) manufactured by Murata Manufacturing Co., Ltd.
C _{G3}	OSC3 gate capacitor	Ceramic capacitor
C _{D3}	OSC3 drain capacitor	Ceramic capacitor
C _{FW1}	Bypass capacitor between V _{SS} and V _{DD}	Ceramic capacitor or electrolytic capacitor
C _{FW2}	Capacitor between V _{SS} and V _{D1}	Ceramic capacitor
BZ	Piezoelectric buzzer	PS1240P02 manufactured by TDK Corporation
R _{DBG}	DSIO pull-up resistor	Thick film chip resistor
R _{REF}	RFC reference resistor	Thick film chip resistor
R _{TMP1, 2}	Resistive sensors	Temperature sensor 103AP-2 manufactured by SEMITEC Corporation
C _{REF}	RFC reference capacitor	Ceramic capacitor
C _{VREFA}	Capacitor between V _{SS} and V _{REFA}	Ceramic capacitor
C _{VPP}	Capacitor between V _{SS} and V _{PP}	Ceramic capacitor

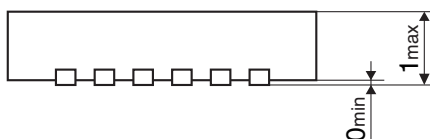
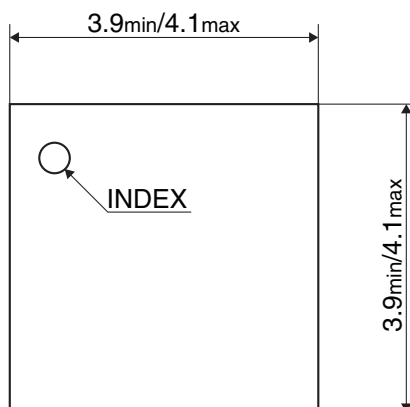
* For recommended component values, refer to “Recommended Operating Conditions” in the “Electrical Characteristics” chapter.

23 Package

SQFN4-24PIN (P-VQFN024-0404-0.50) [S1C17M20/M23]

(Unit: mm)

Top View



Bottom View

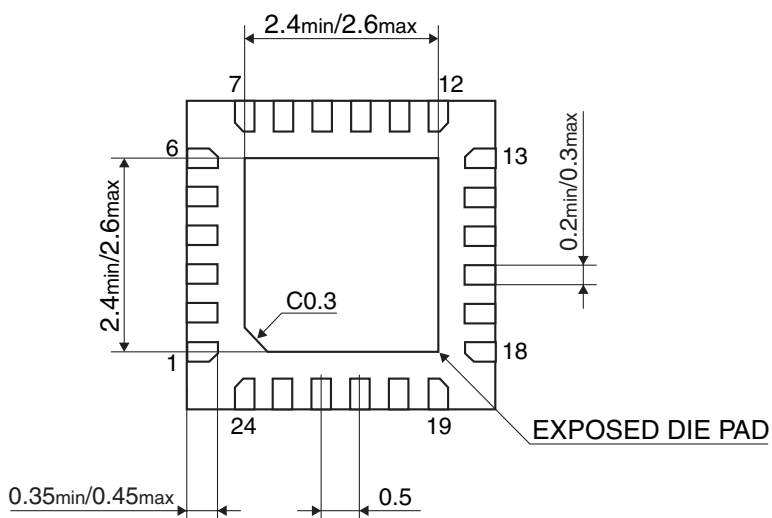


Figure 23.1 SQFN4-24PIN Package Dimensions

* The potential of the EXPOSED DIE PAD is the same as that of the substrate potential (Vss) on the back of the IC.

SQFN5-32PIN (P-VQFN032-0505-0.50) [S1C17M20/M23]

(Unit: mm)

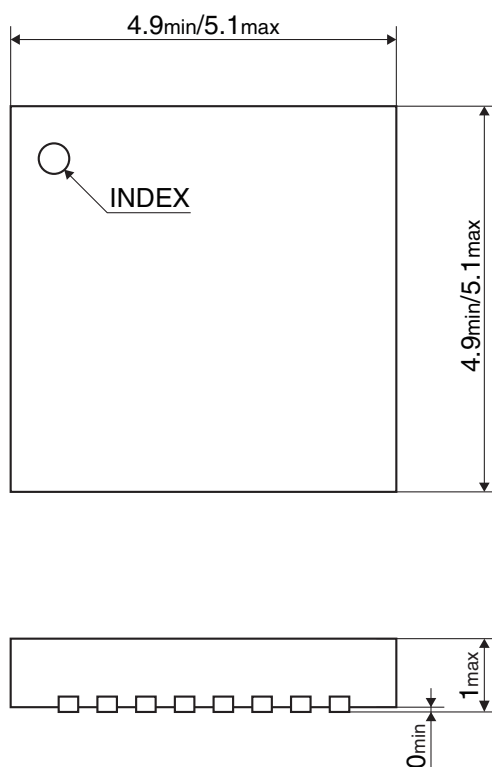
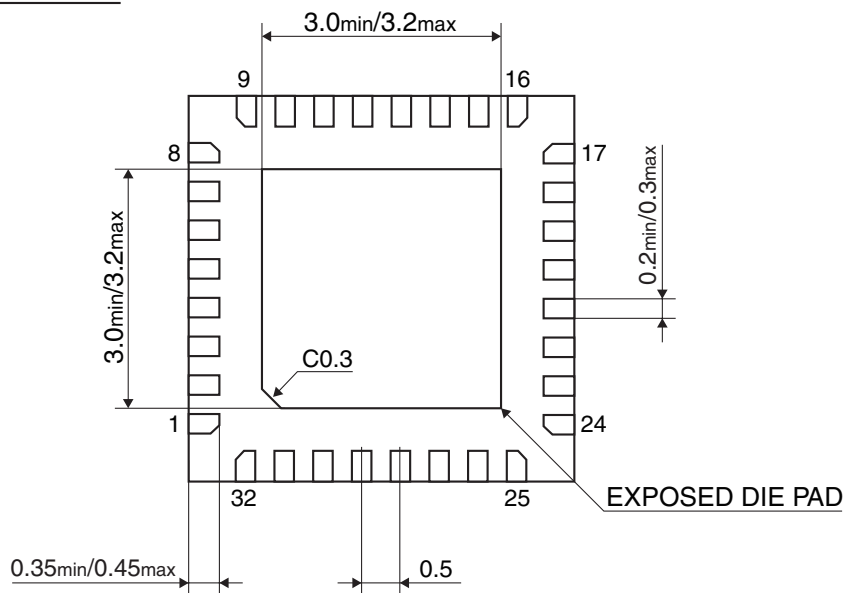
Top ViewBottom View

Figure 23.2 SQFN5-32PIN Package Dimensions

* The potential of the EXPOSED DIE PAD is the same as that of the substrate potential (Vss) on the back of the IC.

TQFP12-32PIN (P-TQFP032-0707-0.80) [S1C17M21/M24]

(Unit: mm)

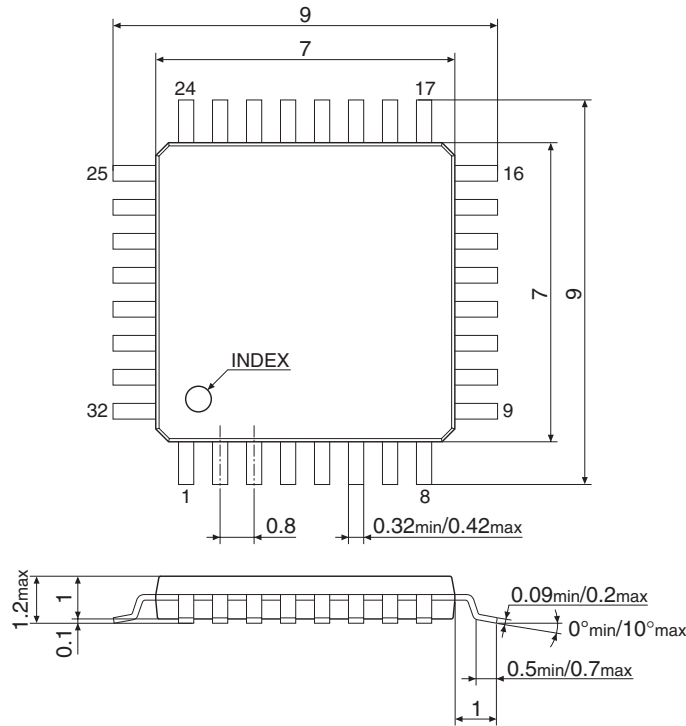


Figure 23.3 TQFP12-32PIN Package Dimensions

TQFP12-48PIN (P-TQFP048-0707-0.50) [S1C17M22/M25]

(Unit: mm)

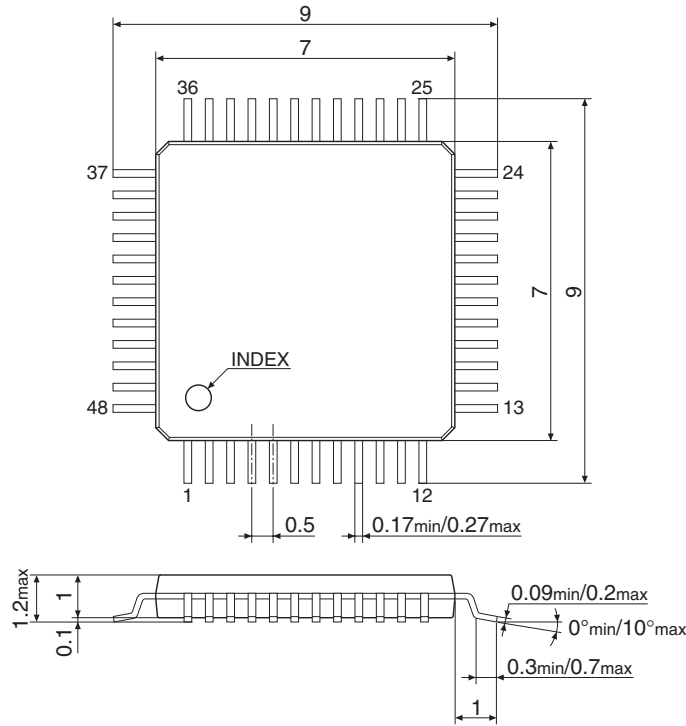


Figure 23.4 TQFP12-48PIN Package Dimensions

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				Power Generator (PWG)			
Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4020	PWGVD1CTL (PWG V _{D1} Regulator Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1–0	REGMODE[1:0]	0x0	H0	R/WP	

0x4040–0x4050				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	–	0	–	R	–
		14	OSDRB	1	H0	R/WP	
		13	OSDEN	0	H0	R/WP	
		12	OSC1BUP	1	H0	R/WP	
		11	OSC1SELCR	0	H0	R/WP	
		10–8	CGI1[2:0]	0x0	H0	R/WP	
		7–6	INV1B[1:0]	0x2	H0	R/WP	
		5–4	INV1N[1:0]	0x1	H0	R/WP	
		3–2	–	0x0	–	R	
		1–0	OSC1WT[1:0]	0x2	H0	R/WP	
0x4048	CLGOSC3 (CLG OSC3 Control Register)	15–12	–	0x0	–	R	–
		11–10	OSC3FQ[1:0]	0x1	H0	R/WP	
		9	OSC3MD	0	H0	R/WP	
		8	–	0	–	R	
		7–6	–	0x0	–	R	
		5–4	OSC3INV[1:0]	0x3	H0	R/WP	
		3	OSC3STM	0	H0	R/WP	
		2–0	OSC3WT[2:0]	0x6	H0	R/WP	
0x404c	CLGINTF (CLG Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7	–	0x0	–	R	
		6	(reserved)	0	H0	R	
		5	OSC1STPIF	0	H0	R/W	Cleared by writing 1.
		4	OSC3TEDIF	0	H0	R/W	
		3	–	0	–	R	–
		2	OSC3STAIF	0	H0	R/W	
		1	OSC1STAIF	0	H0	R/W	
		0	IOSCSTAIF	0	H0	R/W	
0x404e	CLGINTE (CLG Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6	(reserved)	0	H0	R	
		5	OSC1STPIE	0	H0	R/W	
		4	OSC3TEDIE	0	H0	R/W	
		3	–	0	–	R	
		2	OSC3STAIE	0	H0	R/W	
		1	OSC1STAIE	0	H0	R/W	
		0	IOSCSTAIE	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	
		0	FOUTEN	0	H0	R/W	

0x4080–0x4094

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 (ILVSVD3)
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	–

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
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)
		7–3	–	0x00	–	R	–
		2–0	ILV4[2:0]	0x0	H0	R/W	Real-time clock interrupt (ILVRTCA_0)
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	–
		2–0	ILV6[2:0]	0x0	H0	R/W	UART Ch.0 interrupt (ILVUART3_0)
0x4088	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)
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	Sound generator interrupt (ILVSND_0)
		7–3	–	0x00	–	R	–
		2–0	ILV12[2:0]	0x0	H0	R/W	UART Ch.1 interrupt (ILVUART3_1)
0x408e	ITCLV7 (ITC Interrupt Level Setup Register 7)	15–8	–	0x00	–	R	–
		7–3	–	0x00	–	R	–
		2–0	ILV14[2:0]	0x0	H0	R/W	IR remote controller interrupt (ILVREMC3_0)
0x4090	ITCLV8 (ITC Interrupt Level Setup Register 8)	15–11	–	0x00	–	R	–
		10–8	ILV17[2:0]	0x0	H0	R/W	R/F converter Ch.1 interrupt (ILVRFC_1)
		7–3	–	0x00	–	R	–
		2–0	ILV16[2:0]	0x0	H0	R/W	R/F converter Ch.0 interrupt (ILVRFC_0)
0x4092	ITCLV9 (ITC Interrupt Level Setup Register 9)	15–11	–	0x00	–	R	–
		10–8	ILV19[2:0]	0x0	H0	R/W	Synchronous serial interface Ch.1 interrupt (ILVSPIA_1)
		7–3	–	0x00	–	R	–
		2–0	ILV18[2:0]	0x0	H0	R/W	16-bit timer Ch.2 interrupt (ILVT16_2)
0x4094	ITCLV10 (ITC Interrupt Level Setup Register 10)	15–11	–	0x00	–	R	–
		10–8	ILV21[2:0]	0x0	–	R/W	12-bit A/D converter interrupt (ILVADC12A_0)
		7–3	–	0x00	–	R	–
		2–0	ILV20[2:0]	0x0	–	R/W	16-bit timer Ch.3 interrupt (ILVT16_3)

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

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	WDTCNTRST	0	H0	WP	Always read as 0.
		3–0	WDRUN[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	

0x40c0–0x40d2			Real-time Clock (RTCA)				
Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x40c0	RTCCTL (RTC Control Register)	15	RTCTRMBSY	0	H0	R	–
		14–8	RTCTRM[6:0]	0x00	H0	W	Read as 0x00.
		7	–	0	–	R	–
		6	RTCBSY	0	H0	R	–
		5	RTCHLD	0	H0	R/W	Cleared by setting the RTCCTL.RTCRST bit to 1.
		4	RTC24H	0	H0	R/W	–
		3	–	0	–	R	–
		2	RTCADJ	0	H0	R/W	Cleared by setting the RTCCTL.RTCRST bit to 1.
		1	RTCRST	0	H0	R/W	–
		0	RTCRUN	0	H0	R/W	–
0x40c2	RTCALM1 (RTC Second Alarm Register)	15	–	0	–	R	–
		14–12	RTCSHA[2:0]	0x0	H0	R/W	
		11–8	RTCSLA[3:0]	0x0	H0	R/W	
		7–0	–	0x00	–	R	
0x40c4	RTCALM2 (RTC Hour/Minute Alarm Register)	15	–	0	–	R	–
		14	RTCAPA	0	H0	R/W	
		13–12	RTCHHA[1:0]	0x0	H0	R/W	
		11–8	RTCHLA[3:0]	0x0	H0	R/W	
		7	–	0	–	R	
		6–4	RTCMIHA[2:0]	0x0	H0	R/W	
		3–0	RTCMILA[3:0]	0x0	H0	R/W	
0x40c6	RTCSWCTL (RTC Stopwatch Control Register)	15–12	BCD10[3:0]	0x0	H0	R	–
		11–8	BCD100[3:0]	0x0	H0	R	
		7–5	–	0x0	–	R	
		4	SWRST	0	H0	W	Read as 0.
		3–1	–	0x0	–	R	–
		0	SWRUN	0	H0	R/W	

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x40c8	RTCSEC (RTC Second/1Hz Register)	15	–	0	–	R	Cleared by setting the RTCCTL.RTCRST bit to 1.
		14–12	RTCSH[2:0]	0x0	H0	R/W	
		11–8	RTCSL[3:0]	0x0	H0	R/W	
		7	RTC1HZ	0	H0	R	
		6	RTC2HZ	0	H0	R	
		5	RTC4HZ	0	H0	R	
		4	RTC8HZ	0	H0	R	
		3	RTC16HZ	0	H0	R	
		2	RTC32HZ	0	H0	R	
		1	RTC64HZ	0	H0	R	
		0	RTC128HZ	0	H0	R	
0x40ca	RTCHUR (RTC Hour/Minute Register)	15	–	0	–	R	
		14	RTCAP	0	H0	R/W	
		13–12	RTCHH[1:0]	0x1	H0	R/W	
		11–8	RTCHL[3:0]	0x2	H0	R/W	
		7	–	0	–	R	
		6–4	RTCMIH[2:0]	0x0	H0	R/W	
		3–0	RTCMIL[3:0]	0x0	H0	R/W	
0x40cc	RTCMON (RTC Month/Day Register)	15–13	–	0x0	–	R	
		12	RTCMOH	0	H0	R/W	
		11–8	RTCMOL[3:0]	0x1	H0	R/W	
		7–6	–	0x0	–	R	
		5–4	RTCDH[1:0]	0x0	H0	R/W	
		3–0	RTCDL[3:0]	0x1	H0	R/W	
0x40ce	RTCYAR (RTC Year/Week Register)	15–11	–	0x00	–	R	
		10–8	RTCWK[2:0]	0x0	H0	R/W	
		7–4	RTCYH[3:0]	0x0	H0	R/W	
		3–0	RTCYL[3:0]	0x0	H0	R/W	
0x40d0	RTCINTF (RTC Interrupt Flag Register)	15	RTCTRMIF	0	H0	R/W	Cleared by writing 1.
		14	SW1IF	0	H0	R/W	
		13	SW10IF	0	H0	R/W	
		12	SW100IF	0	H0	R/W	
		11–9	–	0x0	–	R	Cleared by writing 1.
		8	ALARMIF	0	H0	R/W	
		7	1DAYIF	0	H0	R/W	
		6	1HURIF	0	H0	R/W	
		5	1MINIF	0	H0	R/W	
		4	1SECIIF	0	H0	R/W	
		3	1_2SECIIF	0	H0	R/W	
		2	1_4SECIIF	0	H0	R/W	
		1	1_8SECIIF	0	H0	R/W	
		0	1_32SECIIF	0	H0	R/W	
0x40d2	RTCINTE (RTC Interrupt Enable Register)	15	RTCTRMIE	0	H0	R/W	
		14	SW1IE	0	H0	R/W	
		13	SW10IE	0	H0	R/W	
		12	SW100IE	0	H0	R/W	
		11–9	–	0x0	–	R	
		8	ALARMIE	0	H0	R/W	
		7	1DAYIE	0	H0	R/W	
		6	1HURIE	0	H0	R/W	
		5	1MINIE	0	H0	R/W	
		4	1SECIE	0	H0	R/W	
		3	1_2SECIE	0	H0	R/W	
		2	1_4SECIE	0	H0	R/W	
		1	1_8SECIE	0	H0	R/W	
		0	1_32SECIE	0	H0	R/W	

0x4100–0x4106**Supply Voltage Detector (SVD3)**

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4100	SVDCLK (SVD3 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 (SVD3 Control Register)	15	VDSEL	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 (SVD3 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.
0x4106	SVDINTE (SVD3 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_OINTF (T16 Ch.0 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x416c	T16_OINTE (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)	15–9	–	0x00	–	R	–
		8	(reserved)	0	H0	R/WP	Always set to 0.
		7–2	–	0x00	–	R	–
		1–0	RDWAIT[1:0]	0x1	H0	R/WP	–

0x4200–0x42e2**I/O Ports (PPORT)**

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
								24pin	32pin	M24	M25
0x4200	P0DAT (P0 Port Data Register)	15	P0OUT7	0	H0	R/W	–	–	–	–	✓
		14	P0OUT6	0	H0	R/W		–	–	–	✓
		13	P0OUT5	0	H0	R/W		–	–	–	✓
		12	P0OUT4	0	H0	R/W		–	–	–	✓
		11	P0OUT3	0	H0	R/W		✓	✓	✓	✓
		10	P0OUT2	0	H0	R/W		✓	✓	✓	✓
		9	P0OUT1	0	H0	R/W		✓	✓	✓	✓
		8	P0OUT0	0	H0	R/W		✓	✓	✓	✓
		7	P0IN7	0	H0	R	–	–	–	–	✓
		6	P0IN6	0	H0	R		–	–	–	✓
		5	P0IN5	0	H0	R		–	–	–	✓
		4	P0IN4	0	H0	R		–	–	–	✓
		3	P0IN3	0	H0	R		✓	✓	✓	✓
		2	P0IN2	0	H0	R		✓	✓	✓	✓
		1	P0IN1	0	H0	R		✓	✓	✓	✓
		0	P0IN0	0	H0	R		✓	✓	✓	✓
0x4202	P0IOEN (P0 Port Enable Register)	15	P0IEN7	0	H0	R/W	–	–	–	–	✓
		14	P0IEN6	0	H0	R/W		–	–	–	✓
		13	P0IEN5	0	H0	R/W		–	–	–	✓
		12	P0IEN4	0	H0	R/W		–	–	–	✓
		11	P0IEN3	0	H0	R/W		✓	✓	✓	✓
		10	P0IEN2	0	H0	R/W		✓	✓	✓	✓
		9	P0IEN1	0	H0	R/W		✓	✓	✓	✓
		8	P0IEN0	0	H0	R/W		✓	✓	✓	✓
		7	P0OEN7	0	H0	R/W	–	–	–	–	✓
		6	P0OEN6	0	H0	R/W		–	–	–	✓
		5	P0OEN5	0	H0	R/W		–	–	–	✓
		4	P0OEN4	0	H0	R/W		–	–	–	✓
		3	P0OEN3	0	H0	R/W		✓	✓	✓	✓
		2	P0OEN2	0	H0	R/W		✓	✓	✓	✓
		1	P0OEN1	0	H0	R/W		✓	✓	✓	✓
		0	P0OEN0	0	H0	R/W		✓	✓	✓	✓
0x4204	P0RCTL (P0 Port Pull-up/down Control Register)	15	P0PDPU7	0	H0	R/W	–	–	–	–	✓
		14	P0PDPU6	0	H0	R/W		–	–	–	✓
		13	P0PDPU5	0	H0	R/W		–	–	–	✓
		12	P0PDPU4	0	H0	R/W		–	–	–	✓
		11	P0PDPU3	0	H0	R/W		✓	✓	✓	✓
		10	P0PDPU2	0	H0	R/W		✓	✓	✓	✓
		9	P0PDPU1	0	H0	R/W		✓	✓	✓	✓
		8	P0PDPU0	0	H0	R/W		✓	✓	✓	✓
		7	P0REN7	0	H0	R/W	–	–	–	–	✓
		6	P0REN6	0	H0	R/W		–	–	–	✓
		5	P0REN5	0	H0	R/W		–	–	–	✓
		4	P0REN4	0	H0	R/W		–	–	–	✓
		3	P0REN3	0	H0	R/W		✓	✓	✓	✓
		2	P0REN2	0	H0	R/W		✓	✓	✓	✓
		1	P0REN1	0	H0	R/W		✓	✓	✓	✓
		0	P0REN0	0	H0	R/W		✓	✓	✓	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23 24pin	M21/ 32pin	M22/ M24	M25
0x4206	POINTF (P0 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7	P0IF7	0	H0	R/W	Cleared by writing 1.	–	–	–	✓
		6	P0IF6	0	H0	R/W		–	–	–	✓
		5	P0IF5	0	H0	R/W		–	–	–	✓
		4	P0IF4	0	H0	R/W		–	–	–	✓
		3	P0IF3	0	H0	R/W		✓	✓	✓	✓
		2	P0IF2	0	H0	R/W		✓	✓	✓	✓
		1	P0IF1	0	H0	R/W		✓	✓	✓	✓
		0	P0IF0	0	H0	R/W		✓	✓	✓	✓
0x4208	POINTCTL (P0 Port Interrupt Control Register)	15	P0EDGE7	0	H0	R/W	–	–	–	–	✓
		14	P0EDGE6	0	H0	R/W		–	–	–	✓
		13	P0EDGE5	0	H0	R/W		–	–	–	✓
		12	P0EDGE4	0	H0	R/W		–	–	–	✓
		11	P0EDGE3	0	H0	R/W		✓	✓	✓	✓
		10	P0EDGE2	0	H0	R/W		✓	✓	✓	✓
		9	P0EDGE1	0	H0	R/W		✓	✓	✓	✓
		8	P0EDGE0	0	H0	R/W		✓	✓	✓	✓
		7	P0IE7	0	H0	R/W	–	–	–	–	✓
		6	P0IE6	0	H0	R/W		–	–	–	✓
		5	P0IE5	0	H0	R/W		–	–	–	✓
		4	P0IE4	0	H0	R/W		–	–	–	✓
		3	P0IE3	0	H0	R/W		✓	✓	✓	✓
		2	P0IE2	0	H0	R/W		✓	✓	✓	✓
		1	P0IE1	0	H0	R/W		✓	✓	✓	✓
		0	P0IE0	0	H0	R/W		✓	✓	✓	✓
0x420a	POCHATEN (P0 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7	POCHATEN7	0	H0	R/W	–	–	–	–	✓
		6	POCHATEN6	0	H0	R/W		–	–	–	✓
		5	POCHATEN5	0	H0	R/W		–	–	–	✓
		4	POCHATEN4	0	H0	R/W		–	–	–	✓
		3	POCHATEN3	0	H0	R/W		✓	✓	✓	✓
		2	POCHATEN2	0	H0	R/W		✓	✓	✓	✓
		1	POCHATEN1	0	H0	R/W		✓	✓	✓	✓
		0	POCHATEN0	0	H0	R/W		✓	✓	✓	✓
0x420c	P0MODESEL (P0 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7	P0SEL7	0	H0	R/W	–	–	–	–	✓
		6	P0SEL6	0	H0	R/W		–	–	–	✓
		5	P0SEL5	0	H0	R/W		–	–	–	✓
		4	P0SEL4	0	H0	R/W		–	–	–	✓
		3	P0SEL3	0	H0	R/W		✓	✓	✓	✓
		2	P0SEL2	0	H0	R/W		✓	✓	✓	✓
		1	P0SEL1	0	H0	R/W		✓	✓	✓	✓
		0	P0SEL0	0	H0	R/W		✓	✓	✓	✓
0x420e	P0FNCSSEL (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		✓	✓	✓	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23 24pin	M21/ 32pin	M22/ M24	M25
0x4210	P1DAT (P1 Port Data Register)	15	P1OUT7	0	H0	R/W	-	-	-	-	✓
		14	P1OUT6	0	H0	R/W		-	-	-	✓
		13	P1OUT5	0	H0	R/W		✓	✓	✓	✓
		12	P1OUT4	0	H0	R/W		✓	✓	✓	✓
		11	P1OUT3	0	H0	R/W		✓	✓	✓	✓
		10	P1OUT2	0	H0	R/W		✓	✓	✓	✓
		9	P1OUT1	0	H0	R/W		-	✓	✓	✓
		8	P1OUT0	0	H0	R/W		-	✓	✓	✓
		7	P1IN7	0	H0	R	-	-	-	-	✓
		6	P1IN6	0	H0	R		-	-	-	✓
		5	P1IN5	0	H0	R		✓	✓	✓	✓
		4	P1IN4	0	H0	R		✓	✓	✓	✓
		3	P1IN3	0	H0	R		✓	✓	✓	✓
		2	P1IN2	0	H0	R		✓	✓	✓	✓
		1	P1IN1	0	H0	R		-	✓	✓	✓
		0	P1IN0	0	H0	R		-	✓	✓	✓
0x4212	P1IOEN (P1 Port Enable Register)	15	P1IEN7	0	H0	R/W	-	-	-	-	✓
		14	P1IEN6	0	H0	R/W		-	-	-	✓
		13	P1IEN5	0	H0	R/W		✓	✓	✓	✓
		12	P1IEN4	0	H0	R/W		✓	✓	✓	✓
		11	P1IEN3	0	H0	R/W		✓	✓	✓	✓
		10	P1IEN2	0	H0	R/W		✓	✓	✓	✓
		9	P1IEN1	0	H0	R/W		-	✓	✓	✓
		8	P1IEN0	0	H0	R/W		-	✓	✓	✓
		7	P1OEN7	0	H0	R/W	-	-	-	-	✓
		6	P1OEN6	0	H0	R/W		-	-	-	✓
		5	P1OEN5	0	H0	R/W		✓	✓	✓	✓
		4	P1OEN4	0	H0	R/W		✓	✓	✓	✓
		3	P1OEN3	0	H0	R/W		✓	✓	✓	✓
		2	P1OEN2	0	H0	R/W		✓	✓	✓	✓
		1	P1OEN1	0	H0	R/W		-	✓	✓	✓
		0	P1OEN0	0	H0	R/W		-	✓	✓	✓
0x4214	P1RCTL (P1 Port Pull-up/down Control Register)	15	P1PDPU7	0	H0	R/W	-	-	-	-	✓
		14	P1PDPU6	0	H0	R/W		-	-	-	✓
		13	P1PDPU5	0	H0	R/W		✓	✓	✓	✓
		12	P1PDPU4	0	H0	R/W		✓	✓	✓	✓
		11	P1PDPU3	0	H0	R/W		✓	✓	✓	✓
		10	P1PDPU2	0	H0	R/W		✓	✓	✓	✓
		9	P1PDPU1	0	H0	R/W		-	✓	✓	✓
		8	P1PDPU0	0	H0	R/W		-	✓	✓	✓
		7	P1REN7	0	H0	R/W	-	-	-	-	✓
		6	P1REN6	0	H0	R/W		-	-	-	✓
		5	P1REN5	0	H0	R/W		✓	✓	✓	✓
		4	P1REN4	0	H0	R/W		✓	✓	✓	✓
		3	P1REN3	0	H0	R/W		✓	✓	✓	✓
		2	P1REN2	0	H0	R/W		✓	✓	✓	✓
		1	P1REN1	0	H0	R/W		-	✓	✓	✓
		0	P1REN0	0	H0	R/W		-	✓	✓	✓
0x4216	P1INTF (P1 Port Interrupt Flag Register)	15-8	-	0x00	-	R	-	-	-	-	-
		7	P1IF7	0	H0	R/W	Cleared by writing 1.	-	-	-	✓
		6	P1IF6	0	H0	R/W		-	-	-	✓
		5	P1IF5	0	H0	R/W		✓	✓	✓	✓
		4	P1IF4	0	H0	R/W		✓	✓	✓	✓
		3	P1IF3	0	H0	R/W		✓	✓	✓	✓
		2	P1IF2	0	H0	R/W		✓	✓	✓	✓
		1	P1IF1	0	H0	R/W		-	✓	✓	✓
		0	P1IF0	0	H0	R/W		-	✓	✓	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
								24pin	32pin	M24	M25
0x4218	P1INTCTL (P1 Port Interrupt Control Register)	15	P1EDGE7	0	H0	R/W	-	-	-	-	✓
		14	P1EDGE6	0	H0	R/W		-	-	-	✓
		13	P1EDGE5	0	H0	R/W		✓	✓	✓	✓
		12	P1EDGE4	0	H0	R/W		✓	✓	✓	✓
		11	P1EDGE3	0	H0	R/W		✓	✓	✓	✓
		10	P1EDGE2	0	H0	R/W		✓	✓	✓	✓
		9	P1EDGE1	0	H0	R/W		-	✓	✓	✓
		8	P1EDGE0	0	H0	R/W		-	✓	✓	✓
		7	P1IE7	0	H0	R/W	-	-	-	-	✓
		6	P1IE6	0	H0	R/W		-	-	-	✓
		5	P1IE5	0	H0	R/W		✓	✓	✓	✓
		4	P1IE4	0	H0	R/W		✓	✓	✓	✓
		3	P1IE3	0	H0	R/W		✓	✓	✓	✓
		2	P1IE2	0	H0	R/W		✓	✓	✓	✓
		1	P1IE1	0	H0	R/W		-	✓	✓	✓
		0	P1IE0	0	H0	R/W		-	✓	✓	✓
0x421a	P1CHATEN (P1 Port Chattering Filter Enable Register)	15-8	-	0x00	-	R	-	-	-	-	-
		7	P1CHATEN7	0	H0	R/W	-	-	-	-	✓
		6	P1CHATEN6	0	H0	R/W		-	-	-	✓
		5	P1CHATEN5	0	H0	R/W		✓	✓	✓	✓
		4	P1CHATEN4	0	H0	R/W		✓	✓	✓	✓
		3	P1CHATEN3	0	H0	R/W		✓	✓	✓	✓
		2	P1CHATEN2	0	H0	R/W		✓	✓	✓	✓
		1	P1CHATEN1	0	H0	R/W		-	✓	✓	✓
		0	P1CHATEN0	0	H0	R/W		-	✓	✓	✓
0x421c	P1MODESEL (P1 Port Mode Select Register)	15-8	-	0x00	-	R	-	-	-	-	-
		7	P1SEL7	0	H0	R/W	-	-	-	-	✓
		6	P1SEL6	0	H0	R/W		-	-	-	✓
		5	P1SEL5	0	H0	R/W		✓	✓	✓	✓
		4	P1SEL4	0	H0	R/W		✓	✓	✓	✓
		3	P1SEL3	0	H0	R/W		✓	✓	✓	✓
		2	P1SEL2	0	H0	R/W		✓	✓	✓	✓
		1	P1SEL1	0	H0	R/W		-	✓	✓	✓
		0	P1SEL0	0	H0	R/W		-	✓	✓	✓
0x421e	P1FNCSSEL (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		-	✓	✓	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23 24pin	M21/ 32pin	M22/ M24	M25
0x4220	P2DAT (P2 Port Data Register)	15	P2OUT7	0	H0	R/W	–	✓	✓	✓	✓
		14	P2OUT6	0	H0	R/W		✓	✓	✓	✓
		13	P2OUT5	0	H0	R/W		✓	✓	✓	✓
		12	P2OUT4	0	H0	R/W		✓	✓	✓	✓
		11	P2OUT3	0	H0	R/W		–	✓	✓	✓
		10	P2OUT2	0	H0	R/W		–	✓	✓	✓
		9	P2OUT1	0	H0	R/W		–	–	–	✓
		8	P2OUT0	0	H0	R/W		–	–	–	✓
		7	P2IN7	0	H0	R	–	✓	✓	✓	✓
		6	P2IN6	0	H0	R		✓	✓	✓	✓
		5	P2IN5	0	H0	R		✓	✓	✓	✓
		4	P2IN4	0	H0	R		✓	✓	✓	✓
		3	P2IN3	0	H0	R		–	✓	✓	✓
		2	P2IN2	0	H0	R		–	✓	✓	✓
		1	P2IN1	0	H0	R		–	–	–	✓
		0	P2IN0	0	H0	R		–	–	–	✓
0x4222	P2IOEN (P2 Port Enable Register)	15	P2IEN7	0	H0	R/W	–	✓	✓	✓	✓
		14	P2IEN6	0	H0	R/W		✓	✓	✓	✓
		13	P2IEN5	0	H0	R/W		✓	✓	✓	✓
		12	P2IEN4	0	H0	R/W		✓	✓	✓	✓
		11	P2IEN3	0	H0	R/W		–	✓	✓	✓
		10	P2IEN2	0	H0	R/W		–	✓	✓	✓
		9	P2IEN1	0	H0	R/W		–	–	–	✓
		8	P2IEN0	0	H0	R/W		–	–	–	✓
		7	P2OEN7	0	H0	R/W	–	✓	✓	✓	✓
		6	P2OEN6	0	H0	R/W		✓	✓	✓	✓
		5	P2OEN5	0	H0	R/W		✓	✓	✓	✓
		4	P2OEN4	0	H0	R/W		✓	✓	✓	✓
		3	P2OEN3	0	H0	R/W		–	✓	✓	✓
		2	P2OEN2	0	H0	R/W		–	✓	✓	✓
		1	P2OEN1	0	H0	R/W		–	–	–	✓
		0	P2OEN0	0	H0	R/W		–	–	–	✓
0x4224	P2RCTL (P2 Port Pull-up/down Control Register)	15	P2PDPU7	0	H0	R/W	–	✓	✓	✓	✓
		14	P2PDPU6	0	H0	R/W		✓	✓	✓	✓
		13	P2PDPU5	0	H0	R/W		✓	✓	✓	✓
		12	P2PDPU4	0	H0	R/W		✓	✓	✓	✓
		11	P2PDPU3	0	H0	R/W		–	✓	✓	✓
		10	P2PDPU2	0	H0	R/W		–	✓	✓	✓
		9	P2PDPU1	0	H0	R/W		–	–	–	✓
		8	P2PDPU0	0	H0	R/W		–	–	–	✓
		7	P2REN7	0	H0	R/W	–	✓	✓	✓	✓
		6	P2REN6	0	H0	R/W		✓	✓	✓	✓
		5	P2REN5	0	H0	R/W		✓	✓	✓	✓
		4	P2REN4	0	H0	R/W		✓	✓	✓	✓
		3	P2REN3	0	H0	R/W		–	✓	✓	✓
		2	P2REN2	0	H0	R/W		–	✓	✓	✓
		1	P2REN1	0	H0	R/W		–	–	–	✓
		0	P2RENO	0	H0	R/W		–	–	–	✓
0x4226	P2INTF (P2 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7	P2IF7	0	H0	R/W	Cleared by writing 1.	✓	✓	✓	✓
		6	P2IF6	0	H0	R/W		✓	✓	✓	✓
		5	P2IF5	0	H0	R/W		✓	✓	✓	✓
		4	P2IF4	0	H0	R/W		✓	✓	✓	✓
		3	P2IF3	0	H0	R/W		–	✓	✓	✓
		2	P2IF2	0	H0	R/W		–	✓	✓	✓
		1	P2IF1	0	H0	R/W		–	–	–	✓
		0	P2IF0	0	H0	R/W		–	–	–	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
								24pin	32pin	M24	M25
0x4228	P2INTCTL (P2 Port Interrupt Control Register)	15	P2EDGE7	0	H0	R/W	-	✓	✓	✓	✓
		14	P2EDGE6	0	H0	R/W		✓	✓	✓	✓
		13	P2EDGE5	0	H0	R/W		✓	✓	✓	✓
		12	P2EDGE4	0	H0	R/W		✓	✓	✓	✓
		11	P2EDGE3	0	H0	R/W		-	✓	✓	✓
		10	P2EDGE2	0	H0	R/W		-	✓	✓	✓
		9	P2EDGE1	0	H0	R/W		-	-	-	✓
		8	P2EDGE0	0	H0	R/W		-	-	-	✓
		7	P2IE7	0	H0	R/W	-	✓	✓	✓	✓
		6	P2IE6	0	H0	R/W		✓	✓	✓	✓
		5	P2IE5	0	H0	R/W		✓	✓	✓	✓
		4	P2IE4	0	H0	R/W		✓	✓	✓	✓
		3	P2IE3	0	H0	R/W		-	✓	✓	✓
		2	P2IE2	0	H0	R/W		-	✓	✓	✓
		1	P2IE1	0	H0	R/W		-	-	-	✓
		0	P2IE0	0	H0	R/W		-	-	-	✓
0x422a	P2CHATEN (P2 Port Chattering Filter Enable Register)	15-8	-	0x00	-	R	-	-	-	-	-
		7	P2CHATEN7	0	H0	R/W	-	✓	✓	✓	✓
		6	P2CHATEN6	0	H0	R/W		✓	✓	✓	✓
		5	P2CHATEN5	0	H0	R/W		✓	✓	✓	✓
		4	P2CHATEN4	0	H0	R/W		✓	✓	✓	✓
		3	P2CHATEN3	0	H0	R/W		-	✓	✓	✓
		2	P2CHATEN2	0	H0	R/W		-	✓	✓	✓
		1	P2CHATEN1	0	H0	R/W		-	-	-	✓
		0	P2CHATEN0	0	H0	R/W		-	-	-	✓
0x422c	P2MODESEL (P2 Port Mode Select Register)	15-8	-	0x00	-	R	-	-	-	-	-
		7	P2SEL7	0	H0	R/W	-	✓	✓	✓	✓
		6	P2SEL6	0	H0	R/W		✓	✓	✓	✓
		5	P2SEL5	0	H0	R/W		✓	✓	✓	✓
		4	P2SEL4	0	H0	R/W		✓	✓	✓	✓
		3	P2SEL3	0	H0	R/W		-	✓	✓	✓
		2	P2SEL2	0	H0	R/W		-	✓	✓	✓
		1	P2SEL1	0	H0	R/W		-	-	-	✓
		0	P2SEL0	0	H0	R/W		-	-	-	✓
0x422e	P2FNCSSEL (P2 Port Function Select Register)	15-14	P27MUX[1:0]	0x0	H0	R/W	-	✓	✓	✓	✓
		13-12	P26MUX[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		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		-	-	-	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
								24pin	32pin	M24	M25
0x4230	P3DAT (P3 Port Data Register)	15	P3OUT7	0	H0	R/W	–	–	–	–	✓
		14	P3OUT6	0	H0	R/W		–	–	–	✓
		13	P3OUT5	0	H0	R/W		–	–	–	✓
		12	P3OUT4	0	H0	R/W		–	–	–	✓
		11	P3OUT3	0	H0	R/W		–	–	–	✓
		10	P3OUT2	0	H0	R/W		✓	✓	✓	✓
		9	P3OUT1	0	H0	R/W		✓	✓	✓	✓
		8	P3OUT0	0	H0	R/W		✓	✓	✓	✓
		7	P3IN7	0	H0	R	–	–	–	–	✓
		6	P3IN6	0	H0	R		–	–	–	✓
		5	P3IN5	0	H0	R		–	–	–	✓
		4	P3IN4	0	H0	R		–	–	–	✓
		3	P3IN3	0	H0	R		–	–	–	✓
		2	P3IN2	0	H0	R		✓	✓	✓	✓
		1	P3IN1	0	H0	R		✓	✓	✓	✓
		0	P3IN0	0	H0	R		✓	✓	✓	✓
0x4232	P3IOEN (P3 Port Enable Register)	15	P3IEN7	0	H0	R/W	–	–	–	–	✓
		14	P3IEN6	0	H0	R/W		–	–	–	✓
		13	P3IEN5	0	H0	R/W		–	–	–	✓
		12	P3IEN4	0	H0	R/W		–	–	–	✓
		11	P3IEN3	0	H0	R/W		–	–	–	✓
		10	P3IEN2	0	H0	R/W		✓	✓	✓	✓
		9	P3IEN1	0	H0	R/W		✓	✓	✓	✓
		8	P3IEN0	0	H0	R/W		✓	✓	✓	✓
		7	P3OEN7	0	H0	R/W	–	–	–	–	✓
		6	P3OEN6	0	H0	R/W		–	–	–	✓
		5	P3OEN5	0	H0	R/W		–	–	–	✓
		4	P3OEN4	0	H0	R/W		–	–	–	✓
		3	P3OEN3	0	H0	R/W		–	–	–	✓
		2	P3OEN2	0	H0	R/W		✓	✓	✓	✓
		1	P3OEN1	0	H0	R/W		✓	✓	✓	✓
		0	P3OEN0	0	H0	R/W		✓	✓	✓	✓
0x4234	P3RCTL (P3 Port Pull-up/down Control Register)	15	P3PDPU7	0	H0	R/W	–	–	–	–	✓
		14	P3PDPU6	0	H0	R/W		–	–	–	✓
		13	P3PDPU5	0	H0	R/W		–	–	–	✓
		12	P3PDPU4	0	H0	R/W		–	–	–	✓
		11	P3PDPU3	0	H0	R/W		–	–	–	✓
		10	P3PDPU2	0	H0	R/W		✓	✓	✓	✓
		9	P3PDPU1	0	H0	R/W		✓	✓	✓	✓
		8	P3PDPU0	0	H0	R/W		✓	✓	✓	✓
		7	P3REN7	0	H0	R/W	–	–	–	–	✓
		6	P3REN6	0	H0	R/W		–	–	–	✓
		5	P3REN5	0	H0	R/W		–	–	–	✓
		4	P3REN4	0	H0	R/W		–	–	–	✓
		3	P3REN3	0	H0	R/W		–	–	–	✓
		2	P3REN2	0	H0	R/W		✓	✓	✓	✓
		1	P3REN1	0	H0	R/W		✓	✓	✓	✓
		0	P3RENO	0	H0	R/W		✓	✓	✓	✓
0x4236	P3INTF (P3 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7	P3IF7	0	H0	R/W	Cleared by writing 1.	–	–	–	✓
		6	P3IF6	0	H0	R/W		–	–	–	✓
		5	P3IF5	0	H0	R/W		–	–	–	✓
		4	P3IF4	0	H0	R/W		–	–	–	✓
		3	P3IF3	0	H0	R/W		–	–	–	✓
		2	P3IF2	0	H0	R/W		✓	✓	✓	✓
		1	P3IF1	0	H0	R/W		✓	✓	✓	✓
		0	P3IF0	0	H0	R/W		✓	✓	✓	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/M22	
								24pin	32pin	M24	M25
0x4238	P3INTCTL (P3 Port Interrupt Control Register)	15	P3EDGE7	0	H0	R/W	-	-	-	-	✓
		14	P3EDGE6	0	H0	R/W		-	-	-	✓
		13	P3EDGE5	0	H0	R/W		-	-	-	✓
		12	P3EDGE4	0	H0	R/W		-	-	-	✓
		11	P3EDGE3	0	H0	R/W		-	-	-	✓
		10	P3EDGE2	0	H0	R/W		✓	✓	✓	✓
		9	P3EDGE1	0	H0	R/W		✓	✓	✓	✓
		8	P3EDGE0	0	H0	R/W		✓	✓	✓	✓
		7	P3IE7	0	H0	R/W	-	-	-	-	✓
		6	P3IE6	0	H0	R/W		-	-	-	✓
		5	P3IE5	0	H0	R/W		-	-	-	✓
		4	P3IE4	0	H0	R/W		-	-	-	✓
		3	P3IE3	0	H0	R/W		-	-	-	✓
		2	P3IE2	0	H0	R/W		✓	✓	✓	✓
		1	P3IE1	0	H0	R/W		✓	✓	✓	✓
		0	P3IE0	0	H0	R/W		✓	✓	✓	✓
0x423a	P3CHATEN (P3 Port Chattering Filter Enable Register)	15-8	-	0x00	-	R	-	-	-	-	-
		7	P3CHATEN7	0	H0	R/W	-	-	-	-	✓
		6	P3CHATEN6	0	H0	R/W		-	-	-	✓
		5	P3CHATEN5	0	H0	R/W		-	-	-	✓
		4	P3CHATEN4	0	H0	R/W		-	-	-	✓
		3	P3CHATEN3	0	H0	R/W		-	-	-	✓
		2	P3CHATEN2	0	H0	R/W		✓	✓	✓	✓
		1	P3CHATEN1	0	H0	R/W		✓	✓	✓	✓
		0	P3CHATEN0	0	H0	R/W		✓	✓	✓	✓
0x423c	P3MODESEL (P3 Port Mode Select Register)	15-8	-	0x00	-	R	-	-	-	-	-
		7	P3SEL7	0	H0	R/W	-	-	-	-	✓
		6	P3SEL6	0	H0	R/W		-	-	-	✓
		5	P3SEL5	0	H0	R/W		-	-	-	✓
		4	P3SEL4	0	H0	R/W		-	-	-	✓
		3	P3SEL3	0	H0	R/W		-	-	-	✓
		2	P3SEL2	0	H0	R/W		✓	✓	✓	✓
		1	P3SEL1	0	H0	R/W		✓	✓	✓	✓
		0	P3SEL0	0	H0	R/W		✓	✓	✓	✓
0x423e	P3FNCSSEL (P3 Port Function Select Register)	15-14	P37MUX[1:0]	0x0	H0	R/W	-	-	-	-	✓
		13-12	P36MUX[1:0]	0x0	H0	R/W		-	-	-	✓
		11-10	P35MUX[1:0]	0x0	H0	R/W		-	-	-	✓
		9-8	P34MUX[1:0]	0x0	H0	R/W		-	-	-	✓
		7-6	P33MUX[1:0]	0x0	H0	R/W		-	-	-	✓
		5-4	P32MUX[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		3-2	P31MUX[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		1-0	P30MUX[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	M20/M23 24pin	M21/ 32pin	M24	M22/ M25
0x4240	P4DAT (P4 Port Data Register)	15–11	–	0x00	–	R	–	–	–	–	–
		10	P4OUT2	0	H0	R/W	–	–	–	–	✓
		9	P4OUT1	0	H0	R/W	–	–	–	–	✓
		8	P4OUT0	0	H0	R/W	–	–	–	–	✓
		7–3	–	0x00	–	R	–	–	–	–	–
		2	P4IN2	0	H0	R	–	–	–	–	✓
		1	P4IN1	0	H0	R	–	–	–	–	✓
		0	P4IN0	0	H0	R	–	–	–	–	✓
0x4242	P4IOEN (P4 Port Enable Register)	15–11	–	0x00	–	R	–	–	–	–	–
		10	P4IEN2	0	H0	R/W	–	–	–	–	✓
		9	P4IEN1	0	H0	R/W	–	–	–	–	✓
		8	P4IEN0	0	H0	R/W	–	–	–	–	✓
		7–3	–	0x00	–	R	–	–	–	–	–
		2	P4OEN2	0	H0	R/W	–	–	–	–	✓
		1	P4OEN1	0	H0	R/W	–	–	–	–	✓
		0	P4OEN0	0	H0	R/W	–	–	–	–	✓
0x4244	P4RCTL (P4 Port Pull-up/down Control Register)	15–11	–	0x00	–	R	–	–	–	–	–
		10	P4PDPJ2	0	H0	R/W	–	–	–	–	✓
		9	P4PDPJ1	0	H0	R/W	–	–	–	–	✓
		8	P4PDPJ0	0	H0	R/W	–	–	–	–	✓
		7–3	–	0x00	–	R	–	–	–	–	–
		2	P4REN2	0	H0	R/W	–	–	–	–	✓
		1	P4REN1	0	H0	R/W	–	–	–	–	✓
		0	P4REN0	0	H0	R/W	–	–	–	–	✓
0x4246	P4INTF (P4 Port Interrupt Flag Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7–3	–	0x00	–	R	–	–	–	–	–
		2	P4IF2	0	H0	R/W	Cleared by writing 1.	–	–	–	✓
		1	P4IF1	0	H0	R/W	–	–	–	–	✓
		0	P4IF0	0	H0	R/W	–	–	–	–	✓
0x4248	P4INTCTL (P4 Port Interrupt Control Register)	15–11	–	0x00	–	R	–	–	–	–	–
		10	P4EDGE2	0	H0	R/W	–	–	–	–	✓
		9	P4EDGE1	0	H0	R/W	–	–	–	–	✓
		8	P4EDGE0	0	H0	R/W	–	–	–	–	✓
		7–3	–	0x00	–	R	–	–	–	–	–
		2	P4IE2	0	H0	R/W	–	–	–	–	✓
		1	P4IE1	0	H0	R/W	–	–	–	–	✓
		0	P4IE0	0	H0	R/W	–	–	–	–	✓
0x424a	P4CHATEN (P4 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7–3	–	0x00	–	R	–	–	–	–	–
		2	P4CHATEN2	0	H0	R/W	–	–	–	–	✓
		1	P4CHATEN1	0	H0	R/W	–	–	–	–	✓
		0	P4CHATEN0	0	H0	R/W	–	–	–	–	✓
0x424c	P4MODSEL (P4 Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7–3	–	0x00	–	R	–	–	–	–	–
		2	P4SEL2	0	H0	R/W	–	–	–	–	✓
		1	P4SEL1	0	H0	R/W	–	–	–	–	✓
		0	P4SEL0	0	H0	R/W	–	–	–	–	✓
0x424e	P4FNCSSEL (P4 Port Function Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7–6	–	0x0	–	R	–	–	–	–	–
		5–4	P42MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
		3–2	P41MUX[1:0]	0x0	H0	R/W	–	–	–	–	✓
		1–0	P40MUX[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	M20/M23		M21/	M22/
								24pin	32pin	M24	M25
0x42d0	PDDAT (Pd Port Data Register)	15–13	–	0x0	–	R	–	–	–	–	–
		12	PDOUT4	0	H0	R/W	–	–	✓	✓	✓
		11	PDOUT3	0	H0	R/W	–	–	✓	✓	✓
		10	PDOUT2	0	H0	R/W	–	✓	✓	✓	✓
		9	PDOUT1	0	H0	R/W	–	✓	✓	✓	✓
		8	PDOUT0	0	H0	R/W	–	✓	✓	✓	✓
		7–5	–	0	–	R	–	–	–	–	–
		4	PDIN4	X	H0	R	–	–	✓	✓	✓
		3	PDIN3	X	H0	R	–	–	✓	✓	✓
		2	–	0	–	R	–	–	–	–	–
		1	PDIN1	X	H0	R	–	✓	✓	✓	✓
		0	PDIN0	X	H0	R	–	✓	✓	✓	✓
0x42d2	PDIOEN (Pd Port Enable Register)	15–13	–	0x0	–	R	–	–	–	–	–
		12	PDIEN4	0	H0	R/W	–	–	✓	✓	✓
		11	PDIEN3	0	H0	R/W	–	–	✓	✓	✓
		10	(reserved)	0	H0	R/W	–	✓	✓	✓	✓
		9	PDIEN1	0	H0	R/W	–	✓	✓	✓	✓
		8	PDIEN0	0	H0	R/W	–	✓	✓	✓	✓
		7–5	–	0	–	R	–	–	–	–	–
		4	PDOEN4	0	H0	R/W	–	–	✓	✓	✓
		3	PDOEN3	0	H0	R/W	–	–	✓	✓	✓
		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)	15–13	–	0x0	–	R	–	–	–	–	–
		12	PDPDPU4	0	H0	R/W	–	–	✓	✓	✓
		11	PDPDPU3	0	H0	R/W	–	–	✓	✓	✓
		10	(reserved)	0	H0	R/W	–	✓	✓	✓	✓
		9	PDPDPU1	0	H0	R/W	–	✓	✓	✓	✓
		8	PDPDPU0	0	H0	R/W	–	✓	✓	✓	✓
		7–5	–	0	–	R	–	–	–	–	–
		4	PDREN4	0	H0	R/W	–	–	✓	✓	✓
		3	PDREN3	0	H0	R/W	–	–	✓	✓	✓
		2	(reserved)	0	H0	R/W	–	✓	✓	✓	✓
		1	PDREN1	0	H0	R/W	–	✓	✓	✓	✓
		0	PDREN0	0	H0	R/W	–	✓	✓	✓	✓
0x42dc	PDMODSEL (Pd Port Mode Select Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7–5	–	0	–	R	–	–	–	–	–
		4	PDSEL4	0	H0	R/W	–	–	✓	✓	✓
		3	PDSEL3	0	H0	R/W	–	–	✓	✓	✓
		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)	15–10	–	0x00	–	R	–	–	–	–	–
		9–8	PD4MUX[1:0]	0x0	H0	R/W	–	–	✓	✓	✓
		7–6	PD3MUX[1:0]	0x0	H0	R/W	–	–	✓	✓	✓
		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	–	✓	✓	✓	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
								24pin	32pin	M24	M25
0x42e2	PINTGRP (P Port Interrupt Flag Group Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7–5	–	0x0	–	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		✓	✓	✓	✓

0x4300–0x431e

Universal Port Multiplexer (UPMUX)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/	M22/
								24pin	32pin	M24	M25
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		✓	✓	✓	✓
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		✓	✓	✓	✓

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23		M21/ M24	M22/ M25
								24pin	32pin		
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		–	–	–	✓
0x4310	P2UPMUX0 (P20–21 Universal Port Multiplexer Setting Register)	15–13	P21PPFNC[2:0]	0x0	H0	R/W	–	–	–	–	✓
		12–11	P21PERICH[1:0]	0x0	H0	R/W		–	–	–	✓
		10–8	P21PERISEL[2:0]	0x0	H0	R/W		–	–	–	✓
		7–5	P20PPFNC[2:0]	0x0	H0	R/W		–	–	–	✓
		4–3	P20PERICH[1:0]	0x0	H0	R/W		–	–	–	✓
		2–0	P20PERISEL[2:0]	0x0	H0	R/W		–	–	–	✓
0x4312	P2UPMUX1 (P22–23 Universal Port Multiplexer Setting Register)	15–13	P23PPFNC[2:0]	0x0	H0	R/W	–	–	✓	✓	✓
		12–11	P23PERICH[1:0]	0x0	H0	R/W		–	✓	✓	✓
		10–8	P23PERISEL[2:0]	0x0	H0	R/W		–	✓	✓	✓
		7–5	P22PPFNC[2:0]	0x0	H0	R/W		–	✓	✓	✓
		4–3	P22PERICH[1:0]	0x0	H0	R/W		–	✓	✓	✓
		2–0	P22PERISEL[2:0]	0x0	H0	R/W		–	✓	✓	✓
0x4314	P2UPMUX2 (P24–25 Universal Port Multiplexer Setting Register)	15–13	P25PPFNC[2:0]	0x0	H0	R/W	–	✓	✓	✓	✓
		12–11	P25PERICH[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		10–8	P25PERISEL[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		7–5	P24PPFNC[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		4–3	P24PERICH[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		2–0	P24PERISEL[2:0]	0x0	H0	R/W		✓	✓	✓	✓
0x4316	P2UPMUX3 (P26–27 Universal Port Multiplexer Setting Register)	15–13	P27PPFNC[2:0]	0x0	H0	R/W	–	✓	✓	✓	✓
		12–11	P27PERICH[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		10–8	P27PERISEL[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		7–5	P26PPFNC[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		4–3	P26PERICH[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		2–0	P26PERISEL[2:0]	0x0	H0	R/W		✓	✓	✓	✓
0x4318	P3UPMUX0 (P30–31 Universal Port Multiplexer Setting Register)	15–13	P31PPFNC[2:0]	0x0	H0	R/W	–	✓	✓	✓	✓
		12–11	P31PERICH[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		10–8	P31PERISEL[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		7–5	P30PPFNC[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		4–3	P30PERICH[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		2–0	P30PERISEL[2:0]	0x0	H0	R/W		✓	✓	✓	✓
0x431a	P3UPMUX1 (P32–33 Universal Port Multiplexer Setting Register)	15–13	P33PPFNC[2:0]	0x0	H0	R/W	–	–	–	–	✓
		12–11	P33PERICH[1:0]	0x0	H0	R/W		–	–	–	✓
		10–8	P33PERISEL[2:0]	0x0	H0	R/W		–	–	–	✓
		7–5	P32PPFNC[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		4–3	P32PERICH[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		2–0	P32PERISEL[2:0]	0x0	H0	R/W		✓	✓	✓	✓
0x431c	P3UPMUX2 (P34–35 Universal Port Multiplexer Setting Register)	15–13	P35PPFNC[2:0]	0x0	H0	R/W	–	–	–	–	✓
		12–11	P35PERICH[1:0]	0x0	H0	R/W		–	–	–	✓
		10–8	P35PERISEL[2:0]	0x0	H0	R/W		–	–	–	✓
		7–5	P34PPFNC[2:0]	0x0	H0	R/W		–	–	–	✓
		4–3	P34PERICH[1:0]	0x0	H0	R/W		–	–	–	✓
		2–0	P34PERISEL[2:0]	0x0	H0	R/W		–	–	–	✓
0x431e	P3UPMUX3 (P36–37 Universal Port Multiplexer Setting Register)	15–13	P37PPFNC[2:0]	0x0	H0	R/W	–	–	–	–	✓
		12–11	P37PERICH[1:0]	0x0	H0	R/W		–	–	–	✓
		10–8	P37PERISEL[2:0]	0x0	H0	R/W		–	–	–	✓
		7–5	P36PPFNC[2:0]	0x0	H0	R/W		–	–	–	✓
		4–3	P36PERICH[1:0]	0x0	H0	R/W		–	–	–	✓
		2–0	P36PERISEL[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	
		0	MODEN	0	H0	R/W	
0x4388	UA0TXD (UART3 Ch.0 Transmit Data Register)	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	
		5	FEIF	0	H0/S0	R/W	
		4	PEIF	0	H0/S0	R/W	
		3	OEIF	0	H0/S0	R/W	
		2	RB2FIF	0	H0/S0	R	
		1	RB1FIF	0	H0/S0	R	
		0	TBEIF	1	H0/S0	R	
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	
		0	TBEIE	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	

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

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	
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	
		0	MODEN	0	H0	R/W	
0x43a6	T16_1TR (T16 Ch.1 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x43a8	T16_1TC (T16 Ch.1 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x43aa	T16_1INTF (T16 Ch.1 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
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	
		0	MODEN	0	H0	R/W	
0x43b4	SPI0TXD (SPIA Ch.0 Transmit Data Register)	15–0	TXD[15:0]	0x0000	H0	R/W	–
0x43b6	SPI0RXD (SPIA Ch.0 Receive Data Register)	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	–
		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	Cleared by writing 1.
		6	GCIF	0	H0/S0	R/W	
		5	NACKIF	0	H0/S0	R/W	
		4	STOIF	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 I2C0RXD register.
		0	TBEIF	0	H0/S0	R	Cleared by writing to the I2C0TXD register.
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	STOIE	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	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
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	
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	
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	
		1	TOUTINV	0	H0	R/W	
0x5012	T16B0CCR0 (T16B Ch.0 Compare/Capture 0 Data Register)	15–0	CC[15:0]	0x0000	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	
		1	TOUTINV	0	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	

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
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	
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	
		0	CNTZEROIF	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	
		0	CNTZEROIE	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	
		1	TOUTINV	0	H0	R/W	
		0	CCMD	0	H0	R/W	
0x5052	T16B1CCR0 (T16B Ch.1 Compare/Capture 0 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
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	
		1	TOUTINV	0	H0	R/W	
		0	CCMD	0	H0	R/W	
0x505a	T16B1CCR1 (T16B Ch.1 Compare/ Capture 1 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–

0x5200–0x5210

UART (UART3) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5200	UA1CLK (UART3 Ch.1 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	
0x5202	UA1MOD (UART3 Ch.1 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	
0x5204	UA1BR (UART3 Ch.1 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	
0x5206	UA1CTL (UART3 Ch.1 Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	SFTRST	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5208	UA1TXD (UART3 Ch.1 Trans- mit Data Register)	15–8	–	0x00	–	R	–
		7–0	TXD[7:0]	0x00	H0	R/W	
0x520a	UA1RXD (UART3 Ch.1 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
0x520c	UA1INTF (UART3 Ch.1 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 UA1RXD 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 UA1RXD register.
		1	RB1FIF	0	H0/S0	R	Cleared by writing to the UA1TXD register.
		0	TBEIF	1	H0/S0	R	
0x520e	UA1INTE (UART3 Ch.1 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	
		0	TBEIE	0	H0	R/W	
0x5210	UA1CAWF (UART3 Ch.1 Carrier Waveform Register)	15–8	–	0x00	–	R	–
		7–0	CRPER[7:0]	0x00	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	
		0	MODEN	0	H0	R/W	
0x5266	T16_2TR (T16 Ch.2 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
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	

0x5270–0x527a**Synchronous Serial Interface (SPIA) Ch.1**

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5270	SPI1MOD (SPIA Ch.1 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	
0x5272	SPI1CTL (SPIA Ch.1 Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	SFTRST	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x5274	SPI1TXD (SPIA Ch.1 Transmit Data Register)	15–0	TXD[15:0]	0x0000	H0	R/W	–
0x5276	SPI1RXD (SPIA Ch.1 Receive Data Register)	15–0	RXD[15:0]	0x0000	H0	R	–
0x5278	SPI1INTF (SPIA Ch.1 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 SPI1RXD register.
0x527a	SPI1INTE (SPIA Ch.1 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	

0x5300–0x530a**Sound Generator (SNDA)**

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5300	SNDCLK (SNDA 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 (SNDA 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 (SNDA Control Register)	15–9	–	0x00	–	R	–
		8	SSTP	0	H0	R/W	
		7–1	–	0x00	–	R	
		0	MODEN	0	H0	R/W	
0x5306	SNDDAT (SNDA 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	

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5308	SNDINTF (SNDA 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 (SNDA Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	EMIE	0	H0	R/W	
		0	EDIE	0	H0	R/W	

0x5320–0x5332

IR Remote Controller (REMC3)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5320	REMCLK (REMC3 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	
0x5322	REMDBCTL (REMC3 Data Bit Counter Control Register)	15–10	–	0x00	–	R	–
		9	PRESET	0	H0/S0	R/W	
		8	PRUN	0	H0/S0	R/W	Cleared by writing 1 to the REMDBCTL.REMCRST bit.
		7–5	–	0x0	–	R	–
		4	REMOINV	0	H0	R/W	
		3	BUFEN	0	H0	R/W	
		2	TRMD	0	H0	R/W	
		1	REMCRST	0	H0	W	
		0	MODEN	0	H0	R/W	
0x5324	REMDBCNT (REMC3 Data Bit Counter Register)	15–0	DBCNT[15:0]	0x0000	H0/S0	R	Cleared by writing 1 to the REMDBCTL.REMCRST bit.
0x5326	REMAPLEN (REMC3 Data Bit Active Pulse Length Register)	15–0	APLEN[15:0]	0x0000	H0	R/W	Writing enabled when REMDBCTL.MODEN bit = 1.
0x5328	REMDBLEN (REMC3 Data Bit Length Register)	15–0	DBLEN[15:0]	0x0000	H0	R/W	Writing enabled when REMDBCTL.MODEN bit = 1.
0x532a	REMINTF (REMC3 Status and Interrupt Flag Register)	15–11	–	0x00	–	R	–
		10	DBCNTRUN	0	H0/S0	R	
		9	DBLENBSY	0	H0	R	Cleared by writing 1 to the REMDBCTL.REMCRST bit.
		8	APLENBSY	0	H0	R	Effective when the REMDBCTL.BUFEN bit = 1.
		7–2	–	0x00	–	R	–
		1	DBIF	0	H0/S0	R/W	
		0	APIF	0	H0/S0	R/W	Cleared by writing 1 to this bit or the REMDBCTL.REMCRST bit.
0x532c	REMINTE (REMC3 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	DBIE	0	H0	R/W	
		0	APIE	0	H0	R/W	
0x5330	REMCARR (REMC3 Carrier Waveform Register)	15–8	CRDTY[7:0]	0x00	H0	R/W	–
		7–0	CRPER[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
0x5332	REMCCTL (REMC3 Carrier Modulation Control Register)	15–9	–	0x00	–	R	–
		8	OUTINVEN	0	H0	R/W	
		7–1	–	0x00	–	R	
		0	CARREN	0	H0	R/W	

0x5440–0x5450

R/F Converter (RFC) Ch.0 (S1C17M22/M25)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5440	RFC0CLK (RFC 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	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x5442	RFC0CTL (RFC 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	
0x5444	RFC0TRG (RFC 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 (RFC Ch.0 Measure- ment Counter Low Register)	15–0	MC[15:0]	0x0000	H0	R/W	–
0x5448	RFC0MCH (RFC Ch.0 Measure- ment Counter High Register)	15–8	–	0x00	–	R	–
		7–0	MC[23:16]	0x00	H0	R/W	
0x544a	RFC0TCL (RFC Ch.0 Time Base Counter Low Register)	15–0	TC[15:0]	0x0000	H0	R/W	–
0x544c	RFC0TCH (RFC Ch.0 Time Base Counter High Register)	15–8	–	0x00	–	R	–
		7–0	TC[23:16]	0x00	H0	R/W	
0x544e	RFC0INTF (RFC 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	
		0	EREFIF	0	H0	R/W	
0x5450	RFC0INTE (RFC 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	

APPENDIX A LIST OF PERIPHERAL CIRCUIT CONTROL REGISTERS

0x5460–0x5470		R/F Converter (RFC) Ch.1 (S1C17M22/M25)					
Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5460	RFC1CLK (RFC Ch.1 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	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x5462	RFC1CTL (RFC Ch.1 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	Setting to 0x1 is invalid.
		3–1	–	0x0	–	R	–
		0	MODEN	0	H0	R/W	
0x5464	RFC1TRG (RFC Ch.1 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	
0x5466	RFC1MCL (RFC Ch.1 Measurement Counter Low Register)	15–0	MC[15:0]	0x0000	H0	R/W	–
0x5468	RFC1MCH (RFC Ch.1 Measurement Counter High Register)	15–8	–	0x00	–	R	–
		7–0	MC[23:16]	0x00	H0	R/W	
0x546a	RFC1TCL (RFC Ch.1 Time Base Counter Low Register)	15–0	TC[15:0]	0x0000	H0	R/W	–
0x546c	RFC1TCH (RFC Ch.1 Time Base Counter High Register)	15–8	–	0x00	–	R	–
		7–0	TC[23:16]	0x00	H0	R/W	
0x546e	RFC1INTF (RFC Ch.1 Interrupt Flag Register)	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	
0x5470	RFC1INTE (RFC Ch.1 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	

0x5480–0x548c**16-bit Timer (T16) Ch.3**

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x5480	T16_3CLK (T16 Ch.3 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	
0x5482	T16_3MOD (T16 Ch.3 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x5484	T16_3CTL (T16 Ch.3 Control Register)	15–9	–	0x00	–	R	–
		8	PRUN	0	H0	R/W	
		7–2	–	0x00	–	R	
		1	PRESET	0	H0	R/W	
0x5486	T16_3TR (T16 Ch.3 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x5488	T16_3TC (T16 Ch.3 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x548a	T16_3INTF (T16 Ch.3 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x548c	T16_3INTE (T16 Ch.3 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x54a0–0x54ba**12-bit A/D Converter (ADC12A)**

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks	M20/M23 24pin	M21/ 32pin	M22/ M24	M25
0x54a2	ADC12_0CTL (ADC12A Ch.0 Control Register)	15	–	0	–	R	–	–	–	–	–
		14–12	ADSTAT[2:0]	0x0	H0	R		✓	✓	✓	✓
		11	–	0	–	R		–	–	–	–
		10	BSYSTAT	0	H0	R		✓	✓	✓	✓
		9–8	–	0x0	–	R		–	–	–	–
		7–2	–	0x00	–	R		–	–	–	–
		1	ADST	0	H0	R/W		✓	✓	✓	✓
		0	MODEN	0	H0	R/W		✓	✓	✓	✓
0x54a4	ADC12_0TRG (ADC12A Ch.0 Trigger/Analog Input Select Register)	15–14	–	0x0	–	R	–	–	–	–	–
		13–11	ENDAIN[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		10–8	STAAIN[2:0]	0x0	H0	R/W		✓	✓	✓	✓
		7	STMD	0	H0	R/W		✓	✓	✓	✓
		6	CNVMD	0	H0	R/W		✓	✓	✓	✓
		5–4	CNVTRG[1:0]	0x0	H0	R/W		✓	✓	✓	✓
		3	–	0	–	R		–	–	–	–
0x54a6	ADC12_0CFG (ADC12A Ch.0 Configuration Register)	15–8	–	0x00	–	R	–	–	–	–	–
		7–2	–	0x00	–	R		–	–	–	–
		1–0	VRANGE[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	M20/M23		M21/ M24	M22/ M25
								24pin	32pin		
0x54a8	ADC12_OINTF (ADC12A Ch.0 Interrupt Flag Register)	15	AD7OVIF	0	H0	R/W	Cleared by writing 1.	–	–	–	✓
		14	AD6OVIF	0	H0	R/W		–	–	–	✓
		13	AD5OVIF	0	H0	R/W		–	✓	✓	✓
		12	AD4OVIF	0	H0	R/W		–	✓	✓	✓
		11	AD3OVIF	0	H0	R/W		✓	✓	✓	✓
		10	AD2OVIF	0	H0	R/W		✓	✓	✓	✓
		9	AD1OVIF	0	H0	R/W		✓	✓	✓	✓
		8	AD0OVIF	0	H0	R/W		✓	✓	✓	✓
		7	AD7CIF	0	H0	R/W	Cleared by writing 1.	–	–	–	✓
		6	AD6CIF	0	H0	R/W		–	–	–	✓
		5	AD5CIF	0	H0	R/W		–	✓	✓	✓
		4	AD4CIF	0	H0	R/W		–	✓	✓	✓
		3	AD3CIF	0	H0	R/W		✓	✓	✓	✓
		2	AD2CIF	0	H0	R/W		✓	✓	✓	✓
		1	AD1CIF	0	H0	R/W		✓	✓	✓	✓
		0	AD0CIF	0	H0	R/W		✓	✓	✓	✓
0x54aa	ADC12_OINTE (ADC12A Ch.0 Interrupt Enable Register)	15	AD7OVIE	0	H0	R/W	–	–	–	–	✓
		14	AD6OVIE	0	H0	R/W		–	–	–	✓
		13	AD5OVIE	0	H0	R/W		–	✓	✓	✓
		12	AD4OVIE	0	H0	R/W		–	✓	✓	✓
		11	AD3OVIE	0	H0	R/W		✓	✓	✓	✓
		10	AD2OVIE	0	H0	R/W		✓	✓	✓	✓
		9	AD1OVIE	0	H0	R/W		✓	✓	✓	✓
		8	AD0OVIE	0	H0	R/W		✓	✓	✓	✓
		7	AD7CIE	0	H0	R/W	–	–	–	–	✓
		6	AD6CIE	0	H0	R/W		–	–	–	✓
		5	AD5CIE	0	H0	R/W		–	✓	✓	✓
		4	AD4CIE	0	H0	R/W		–	✓	✓	✓
		3	AD3CIE	0	H0	R/W		✓	✓	✓	✓
		2	AD2CIE	0	H0	R/W		✓	✓	✓	✓
		1	AD1CIE	0	H0	R/W		✓	✓	✓	✓
		0	AD0CIE	0	H0	R/W		✓	✓	✓	✓
0x54ac	ADC12_0AD0D (ADC12A Ch.0 Result Register 0)	15–0	AD0D[15:0]	0x0000	H0	R	–	✓	✓	✓	✓
0x54ae	ADC12_0AD1D (ADC12A Ch.0 Result Register 1)	15–0	AD1D[15:0]	0x0000	H0	R	–	✓	✓	✓	✓
0x54b0	ADC12_0AD2D (ADC12A Ch.0 Result Register 2)	15–0	AD2D[15:0]	0x0000	H0	R	–	✓	✓	✓	✓
0x54b2	ADC12_0AD3D (ADC12A Ch.0 Result Register 3)	15–0	AD3D[15:0]	0x0000	H0	R	–	✓	✓	✓	✓
0x54b4	ADC12_0AD4D (ADC12A Ch.0 Result Register 4)	15–0	AD4D[15:0]	0x0000	H0	R	–	–	✓	✓	✓
0x54b6	ADC12_0AD5D (ADC12A Ch.0 Result Register 5)	15–0	AD5D[15:0]	0x0000	H0	R	–	–	✓	✓	✓
0x54b8	ADC12_0AD6D (ADC12A Ch.0 Result Register 6)	15–0	AD6D[15:0]	0x0000	H0	R	–	–	–	–	✓
0x54ba	ADC12_0AD7D (ADC12A Ch.0 Result Register 7)	15–0	AD7D[15:0]	0x0000	H0	R	–	–	–	–	✓

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	H0	R	
				07c0			

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	RTCA	CPU	Current consumption listed in electrical characteristics		
Standby	↑ Low	Economy	OFF	OFF	OFF	SLEEP	I _{SLP}		
Clock counting			ON			SLEEP or HALT	I _{HALT2}		
Low-speed processing						OSC1 RUN	I _{RUN2}		
Peripheral circuit operations	High ↓	Normal		ON	ON	SLEEP or HALT	I _{HALT1}		
High-speed processing						IOSC/OSC3/EXOSC RUN	I _{RUN1}		

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_{SLP} 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. Note that the real-time clock has no MODEN bit, therefore, current consumption does not vary if it is counting or idle.

OSC1 (crystal) oscillator circuit configurations

The OSC1 (crystal) oscillator circuit provides some configuration items to support various crystal resonators with ranges from cylinder type through surface-mount type. These configurations trade off current consumption for performance as shown below.

- The lower oscillation inverter gain setting (CLGOSC1.INV1B[1:0]/INV1N[1:0] bits) decreases current consumption.
- The lower OSC1 internal gate capacitance setting (CLGOSC1.CGI1[2:0] bits) decreases current consumption.
- Using lower OSC1 external gate and drain capacitances decreases current consumption.
- Using a crystal resonator with lower C_L value decreases current consumption.

However, these configurations may reduce the oscillation margin and increase the frequency error, therefore, be sure to perform matching evaluation using the actual printed circuit board.

OSC3 (crystal/ceramic) oscillator circuit configurations

The OSC3 (crystal/ceramic) oscillator circuit provides some configuration items to support various crystal and ceramic resonators. These configurations trade off current consumption for performance as shown below.

- The lower oscillation inverter gain setting (CLGOSC3.OSC3INV[1:0] bits) decreases current consumption.
- Using lower OSC3 external gate and drain capacitances decreases current consumption.
- Using a resonator with lower CL value decreases current consumption.

However, these configurations may reduce the oscillation margin and increase the frequency error, therefore, be sure to perform matching evaluation using the actual printed circuit board.

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.

Appendix C Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

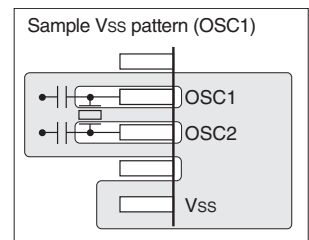
OSC1/OSC3 oscillator circuit

- Oscillation characteristics depend on factors such as components used (resonator, C_G , C_D) and circuit board patterns. In particular, with crystal resonators, select the appropriate capacitors (C_G , C_D) only after fully evaluating components actually mounted on the circuit board.
- Oscillator clock disturbances caused by noise may cause malfunctions. To prevent such disturbances, consider the following points.

- (1) Components such as a resonator, resistors, and capacitors connected to the OSC1 (OSC3) and OSC2 (OSC4) pins should have the shortest connections possible.
- (2) Wherever possible, avoid locating digital signal lines within 3 mm of the OSC1 (OSC3) and OSC2 (OSC4) pins or related circuit components and wiring. Rapidly-switching signals, in particular, should be kept at a distance from these components. Since the spacing between layers of multi-layer printed circuit boards is a mere 0.1 mm to 0.2 mm, the above precautions also apply when positioning digital signal lines on other layers. Never place digital signal lines alongside such components or wiring, even if more than 3 mm distance or located on other layers. Avoid crossing wires.

- (3) Use Vss to shield the OSC1 (OSC3) and OSC2 (OSC4) pins and related wiring (including wiring for adjacent circuit board layers). Layers wired should be adequately shielded as shown to the right. Fully ground adjacent layers, where possible. At minimum, shield the area at least 5 mm around the above pins and wiring.

Even after implementing these precautions, avoid configuring digital signal lines in parallel, as described in (2) above. Avoid crossing even on discrete layers, except for lines carrying signals with low switching frequencies.



- (4) After implementing these precautions, check the FOUT pin output clock waveform by running the actual application program within the product.

For the OSC1 waveform, enlarge the areas before and after the clock rising and falling edges and take special care to confirm that the regions approximately 100 ns to either side are free of clock or spiking noise. For the OSC3 waveform, confirm that the frequency is as designed, is free of noise, and has minimal jitter.

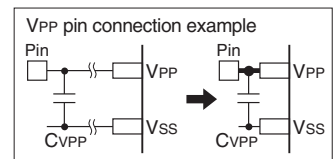
Failure to observe precautions (1) to (3) adequately may lead to noise in OSC1CLK and jitter in OSC3CLK. Noise in the OSC1CLK will destabilize timers that use OSC1CLK as well as CPU Core operations. Jitter in the OSC3 output will reduce operating frequencies.

#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.

VPP pin

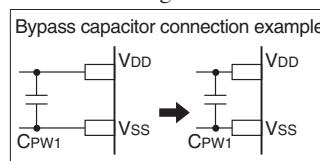
Connect a capacitor C_{VPP} between the Vss and VPP pins to suppress fluctuations within $V_{PP} \pm 1$ V. The C_{VPP} should be placed as close to the VPP pin as possible and use a sufficiently thick wiring pattern that allows current of several tens of mA to flow.



Power supply circuit

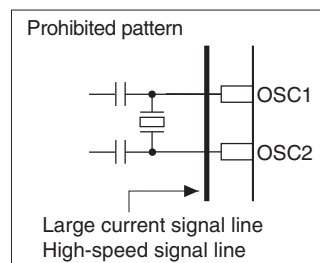
Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

- (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

- 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.
- 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.



Handling of light (for bare chip mounting)

The characteristics of semiconductor components can vary when exposed to light. ICs may malfunction or non-volatile memory data may be corrupted if ICs are exposed to light.

Consider the following precautions for circuit boards and products in which this IC is mounted to prevent IC malfunctions attributable to light exposure.

- (1) Design and mount the product so that the IC is shielded from light during use.
- (2) Shield the IC from light during inspection processes.
- (3) Shield the IC on the upper, underside, and side faces of the IC chip.
- (4) Mount the IC chip within one week of opening the package. If the IC chip must be stored before mounting, take measures to ensure light shielding.
- (5) Adequate evaluations are required to assess nonvolatile memory data retention characteristics before product delivery if the product is subjected to heat stress exceeding regular reflow conditions during mounting processes.

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) OSC1, OSC2, OSC3, OSC4, and EXOSC pins

If the OSC1 crystal oscillator circuit is not used, the OSC1 and OSC2 pins should be left open. If the OSC3 crystal/ceramic oscillator circuit or EXOSC input circuit is not used, the pin should be configured as a general-purpose I/O port. The control registers should be fixed at the initial status (disabled).

Treatment of exposed die pad

The exposed die pad of the packages such as QFN has the same potential as that of the substrate on the back of the IC. When mounting these packages on a circuit board, please note the following:

- (1) When soldering exposed die pad to mounting board

Connect the exposed die pad with a wiring pattern that has the same potential as the substrate potential on the back of the IC, or do not connect it electrically (leave it open electrically). Even if connected to the same potential on the back of the IC, the power supply pins must be connected to the power source (the exposed die pad cannot be used as a power supply pad).

- (2) When not soldering exposed die pad to mounting board

Do not place any signal wiring pattern on the exposed die pad area of the mounting board.

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 VDD and VSS 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 Oscillator Pins

The oscillator input pins must pass a signal of small amplitude, so they are hypersensitive to noise. 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.

Noise Measures for Input Pins Connected to Signal with High Driving Capability Such As Power Supply

There is a possibility of a large current flow into the pins that are directly connected to a power supply or an output of a device with high driving capability if noise is input to those pins. To prevent this, connect a 30 Ω or more pin protection resistor to the pins in series. The resistance value should be determined by evaluating it on the mounting board.

When connecting a power supply directly to the VREFA pin, insert a 100 Ω resistor in series. This resistance does not affect the A/D converter characteristics.

Appendix E Initialization Routine

The following lists typical vector tables and initialization routines:

boot.s

```
.org      0x8000
.section .rodata
; =====
;      Vector table
; =====
;
;      ; interrupt  vector  interrupt
;      ; number    offset  source
;
.long BOOT      ; 0x00      0x00      reset
.long unalign_handler ; 0x01      0x04      unalign
.long nmi_handler ; 0x02      0x08      NMI
.long int03_handler ; 0x03      0x0c      -
.long svd3_handler ; 0x04      0x10      SVD3
.long pport_handler ; 0x05      0x14      PPORT
.long int06_handler ; 0x06      0x18      -
.long clg_handler ; 0x07      0x1c      CLG
.long rtca_handler ; 0x08      0x20      RTCA
.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 uart3_1_handler ; 0x10      0x40      UART3 ch1
.long snda_handler ; 0x11      0x44      SNDA
.long remc3_handler ; 0x12      0x48      REMC3
.long int13_handler ; 0x13      0x4c      -
.long rfc_0_handler ; 0x14      0x50      RFC ch0
.long rfc_1_handler ; 0x15      0x54      RFC ch1
.long t16_2_handler ; 0x16      0x58      T16 ch2
.long spia_1_handler ; 0x17      0x5c      SPIA ch1
.long t16_3_handler ; 0x18      0x60      T16 ch3
.long adc12a_handler ; 0x19      0x64      ADC12A
.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
.align 1
BOOT:
; ===== Initialize =====
;
; ----- Stack pointer -----
xld.a    %sp, 0x7c0
; ----- Memory controller -----
xld.a    %r1, 0x41b0      ; FLASHC register address
; Flash read wait cycle
xld.a    %r0, 0x00      ; 0x00 = No wait
ld.b     [%r1], %r0      ; [0x41b0] <= 0x00
; ===== Main routine =====
...
```

APPENDIX E INITIALIZATION 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
413557000	All	New establishment
413557001	1-2 to 3	<p>1.1 Features</p> <p>Modified Table 1.1.</p> <p>Power supply voltage: V_{DD} operating voltage for Flash programming (When V_{PP} is generated internally) 2.7 to 5.5 V → 2.4 to 5.5 V</p> <p>Shipping form: A JEITA name was added to the package name.</p>
	3-3	<p>3.3.3 List of debugger input/output pins</p> <p>Added a note.</p> <p>Notes: ...</p> <ul style="list-style-type: none"> 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.
	4-3	<p>4.3.3 Flash Programming</p> <p>Corrected the note.</p> <p>Notes: • The Flash programming requires a 2.4 V or higher V_{DD} voltage.</p>
	9-2	<p>9.3.2 Theoretical Regulation Function</p> <p>Corrected Step 1.</p> <p>1. <u>Measure f_{OSC1} and calculate the frequency tolerance correction value.</u> $m [ppm] = -\{(f_{OSC1} - 32,768 [Hz]) / 32,768 [Hz]\} \times 10^6$ <u>(Eq. 9.1) m: OSC1 frequency tolerance correction value [ppm]</u></p>
	9-4	<p>9.4.2 Real-Time Clock Counter Operations</p> <p>Corrective operation when a value out of the effective range is set</p> <p>Added a note.</p> <p>Note: <u>Do not set the RTCMON.RTCMOL[3:0] bits to 0x0 if the RTCMON.RTCMOH bit = 0.</u></p>
	9-11	<p>9.6 Control Registers</p> <p>RTC Month/Day Register</p> <p>Bit 12 RTCMOH</p> <p>Bits 11–8 RTCMOL[3:0]</p> <p>Added a note.</p> <p>Notes: ...</p> <ul style="list-style-type: none"> Be sure to avoid setting the RTCMON.RTCMOH/RTCMOL[3:0] bits to 0x00.
	16-10	<p>16.6 Control Registers</p> <p>SNDA Clock Control Register</p> <p>Modified Table 16.6.1.</p> <p>The IOSC and OSC3 division ratios were corrected.</p>
	21-1	<p>21.1 Absolute Maximum Ratings</p> <p>Modified the characteristics table.</p> <p>V_I: #RESET was added to the condition.</p>
	21-1	<p>21.2 Recommended Operating Conditions</p> <p>Modified the characteristics table.</p> <p>V_{DD}: Min. = 2.7 → 2.4 V, For Flash programming (When V_{PP} is generated internally)</p> <p>CV_{REFA}: *3 and the note were deleted.</p>
	21-4	<p>21.4 System Reset Controller (SRC) Characteristics</p> <p>Reset hold circuit characteristics</p> <p>Modified the characteristics table.</p> <p>t_{RSTR}: Max. = 0.9 ms</p>
	22-1	<p>22 Basic External Connection Diagram</p> <p>V_{DD} for Flash programming → 2.4 V to 5.5 V</p>
	23-1 to 3	<p>23 Package</p> <p>TQFP12-48PIN: S1C17M20/M21 → S1C17M22/M25</p> <p>A JEITA name was added to the package name.</p>
	AP-D-2	<p>Appendix D Measures Against Noise</p> <p>Added a description.</p> <p>Noise Measures for Input Pins Connected to Signal with High Driving Capability Such As Power Supply</p>

America

Epson America, Inc.

Headquarter:
3840 Kilroy Airport Way
Long Beach, California 90806-2452 USA
Phone: +1-562-290-4677

San Jose Office:
214 Devcon Drive
San Jose, CA 95112 USA
Phone: +1-800-228-3964 or +1-408-922-0200

Europe

Epson Europe Electronics GmbH

Riesstrasse 15, 80992 Munich, Germany
Phone: +49-89-14005-0 Fax: +49-89-14005-110

Asia

Epson (China) Co., Ltd.

4th Floor, Tower 1 of China Central Place, 81 Jianguo Road,
Chaoyang District, Beijing 100025, China
Phone: +86-10-8522-1199 Fax: +86-10-8522-1120

Shanghai Branch

Room 1701 & 1704, 17 Floor, Greenland Center II,
562 Dong An Road, Xu Hui District, Shanghai, China
Phone: +86-21-5330-4888 Fax: +86-21-5423-4677

Shenzhen Branch

Room 804-805, 8 Floor, Tower 2, Ali Center, No.3331
Keyuan South RD (Shenzhen bay), Nanshan District,
Shenzhen 518054, China
Phone: +86-10-3299-0588 Fax: +86-10-3299-0560

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.

1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 Fax: +65-6271-3182

Epson Korea Co., Ltd

10F Posco Tower Yeoksam, Teheranro 134 Gangnam-gu,
Seoul, 06235, Korea
Phone: +82-2-3420-6695

Seiko Epson Corp.

Sales & Marketing Division

Device Sales & Marketing Department

29th Floor, JR Shinjuku Miraina Tower, 4-1-6 Shinjuku,
Shinjuku-ku, Tokyo 160-8801, Japan

CMOS 16-BIT SINGLE CHIP MICROCOMPUTER

S1C17 Family

Startup Manual C language

NOTICE

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

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

Windows 2000 and Windows XP are trademarks or registered trademarks of Microsoft Corporation.
PC/AT and IBM are trademarks or registered trademarks of International Business Machines Corporation.
All other brand and product names are trademarks or registered trademarks of respective owners.

Configuration of product number

Devices

S1 C 17xxx F 00E1 00

Packing specifications

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

Specification

Package

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

Model number

Model name

[C: microcomputer, digital products]

Product classification

[S1: semiconductor]

Development tools

S5U1 C 17000 H2 1 00

Packing specifications

[00: standard packing]

Version

[1: Version 1]

Tool type

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

Corresponding model number

[17xxx: for S1C17xxx]

Tool classification

[C: microcomputer use]

Product classification

[S5U1: development tool for semiconductor products]

- Introduction -

The S1C17 Family, 16-bit RISC Processors have various peripheral circuits including abundant interfaces that are compatible with different kinds of sensors, and the LCD driver and controller that covers wide display area. The products suitable for mobile devices are provided in this Family with high-speed operation and low power consumption. The Family has a lineup of many products with built-in Flash ROM. Rich development environment and on-chip IC function provided to the products enable customers to minimize any development period.

This document has been released for an application developer who will use the S1C17 Family products, and it explains the basic embedded programming procedure of the S1C17 Family products.

The reader of this document should have the following basic software knowledge.

- The C language (ANSI C) and C source program creation
- General knowledge of the assembler language
- The knowledge of GNU
- The basic operations of the Windows 2000 or Windows XP OS

Note that sample programs given in this manual have been created by using the S1C17 Family C/C++ Compiler Package (S5U1C17001C) Version 1.2.1.

Manual Configuration:

This manual consists of the following two chapters.

Chapter 1 provides the basic knowledge to create an embedded software.

Chapter 2 explains the basic programming procedure of the S1C17 Family products by using sample programs.

Chapter 3 explains the volatile qualifiers by the use of sample programs.

Chapter 4 explains how to access I/O registers by the use of sample programs.

Related Manuals:

The following lists the related manuals that you should reference to

- S1C17 Core Manual
- S5U1C17001C Manual (S1C17 Family C Compiler Package)
- Technical Manual for each S1C17 Family Model

Table of Contents

1. BASIC KNOWLEDGE OF EMBEDDED PROGRAMS	1
1.1 Basic Mechanism for Program Operation	1
1.2 Startup (Initialize) Routine.....	2
2. PROGRAMMING OF S1C17 FAMILY PROCESSORS.....	3
2.1 Program Development Procedure using the GNU17	3
2.2 Creating a Vector Table	4
2.2.1 Explanation of Vector Table	5
2.2.2 Relationship between Vector Table and Memory Models	6
2.3 Interrupts	8
2.3.1 Prototype Declaration	8
2.3.2 Reset	8
2.3.3 Address Misalignment	9
2.3.4 NMI	9
2.4 Creating a Startup Routine.....	10
2.4.1 Explanation of Startup Routine	11
2.4.2 Setting the SP.....	12
2.4.3 Initializing the .bss/.data Section	13
2.4.4 Enabling an Interrupt (IE)	15
3. “volatile” QUALIFIER.....	16
4. HOW TO ACCESS TO I/O REGISTERS	17
4.1 Read/Write of I/O Registers	17
4.2 Bit Fields.....	18
REVISION HISTORY	19

1. BASIC KNOWLEDGE OF EMBEDDED PROGRAMS

This chapter is intended to be read by the user who develops an embedded software in the first time, and this chapter explains the basic concept that is very important for the user, including the basic mechanism to operate programs, system initialization by the startup routine and others.

1.1 Basic Mechanism for Program Operation

The operation when the S1C17 processor (called “the MCU” hereafter) starts, the basic operation mechanism is explained first.

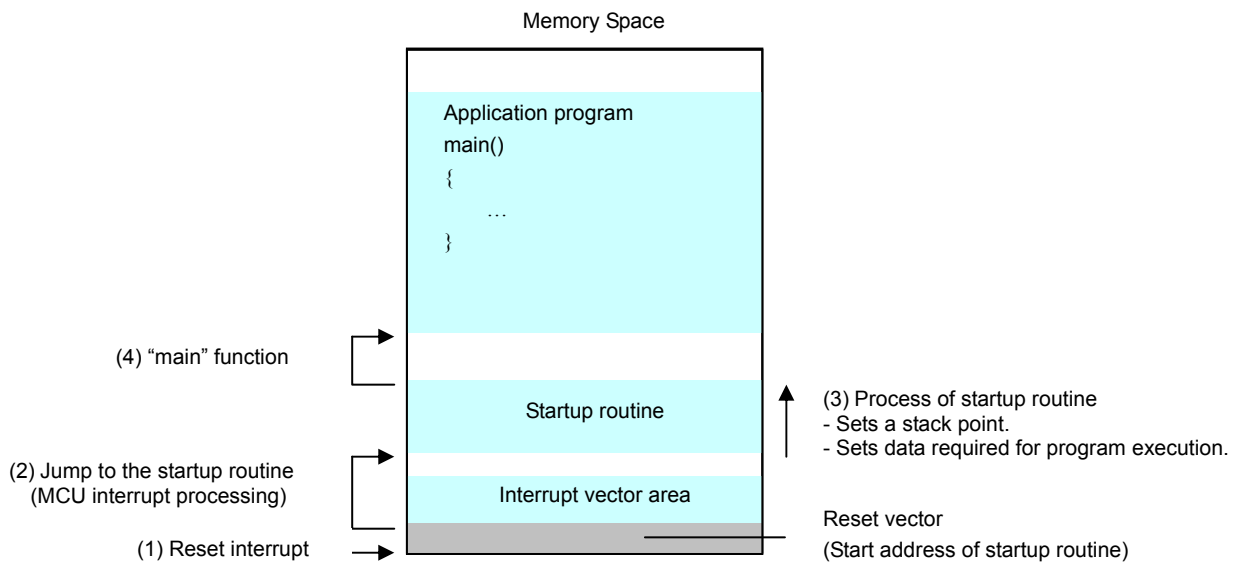


Fig.1.1 Basic Mechanism during Startup of S1C17 Processor

- (1) When the MCU is turned ON, a Reset interrupt occurs and the MCU reads the start address of the vector table.
- (2) The MCU jumps to the content (address) which was read in Step (1) and calls the Startup (Initialize) routine.
- (3) The Startup routine first executes initialization that is required for stack setting and program execution.
- (4) When the initialization process is complete, the Startup routine calls the “main” function.

Note: Addresses of various interrupt process routines are written on the Vector Table. When an interrupt occurs, the appropriate routine address is read from the table and control jumps to the corresponding process routine.

An embedded application program must begin with the Startup routine, but not with the “main” function. The user needs to understand the Startup routine to operate the program when developing an embedded software.

1.2 Startup (Initialize) Routine

The embedded software uses the Startup routine to execute the required initialization before executing the main routine. Generally, the following processes are executed:

- Setup of stack pointer
- Data setup required for program execution
 - Clear the memory area that has no defaults. (Clearing of .bss section)
 - Transfer the default data from ROM area to RAM area. (Copy of .data section)
- Hardware initialization and interrupt setting

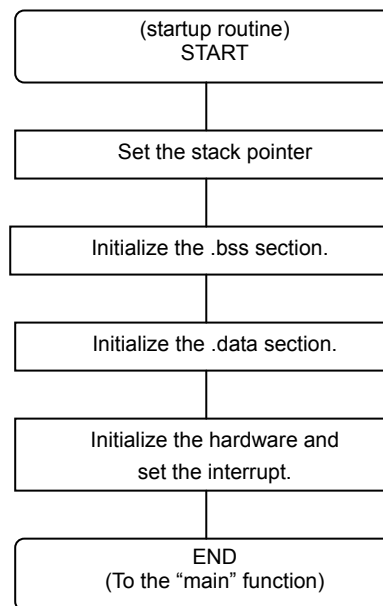


Fig.1.2 Startup Routine

The “stack” is a RAM area where the current processing data, return address and others are saved temporarily when a subroutine or a function is called. As the stack is also used by the interrupt, the stack area needs to be assigned by the startup routine.

When you execute the program, you need to initialize the global variables which have no defaults. Because their setting may be indefinite when reset, you need to initialize them (by clearing the .bss section). Also, if global variables have defaults, you need to copy their defaults from the ROM to RAM (by copying the .data section).

In addition, you need to initialize not only the variables relating to the software execution but also the values required for MCU and other hardware operations. During interrupt setting, you also need to enable an external interrupt that can be masked.

For the embedded applications, the startup routine is first executed and then the “main” function is called.

You should consider these basics and develop an embedded device program.

2. PROGRAMMING OF S1C17 FAMILY PROCESSORS

This chapter explains how to create a program that is common to the S1C17 Family processors.

As explained in Chapter 1, the startup routine needs to be executed for an embedded application as the preprocessing of “main” function execution. The following gives a sample program and explains the standard process flow until you call the startup routine and the “main” function.

2.1 Program Development Procedure using the GNU17

GNU17 is the integrated program development environment that contains a series of software tools and utilities for compiling of C source programs and for assembling and debugging of assembler source programs.

To install the GNU17, access to the “EPSON Microcontroller User Site” and to the “S1C17 Family” and download the “S1C17 Software Integrated Development Environment GNU17.”

The following shows the standard program development flow using the GNU17.

- (1) Developing a project
Develop a new project using the GNU17.
- (2) Creating a source program
Create a source file using the GNU17 editor or a general-purpose editor, and add this file to the project.
- (3) Building the program
Using GNU17, set up the startup options and linker scripts from C compiler to the linker. When you execute the build from GNU17, an elf-format object file that can be debugged and a ROM data file (psa file) which is the object file converted into S-record format.
- (4) Debugging
Check the program operation and debug it using the elf-format object file (created by the linker) and the S-record format ROM data file. You can set and start debugging from the GNU17.

For the detailed information, refer to the “Software Development Procedure” of the “S5U1C17001C MANUAL.” You can find out the “S5U1C17001C MANUAL” under the “EPSON¥GNU17¥doc” directory when you have installed the GNU17.

2.2 Creating a Vector Table

The vector table and the startup routine are minimum required to execute an S1C17 program. This section explains the vector table. The startup routine is explained in Section 2.4 of this manual.

List 2.1 An Example Vector Table

```

/* Prototype */
void boot(void);
void dummy(void) __attribute__((interrupt_handler));
void nmi(void) __attribute__((interrupt_handler));

/* special definitions for vector table */
typedef void func(void);
#ifdef __POINTER16
#define VECTOR(vec) ((vec), 0)
#else
#define VECTOR(vec) (vec)
#endif

```

→ ①

```

/* vector table */
func *const vector[] = {
    VECTOR(boot),           // No   Base+
    VECTOR(dummy),          // 0    00
    VECTOR(dummy),          // 1    04
    VECTOR(nmi),             // 2    08
    VECTOR(dummy),          // 3    0c
    VECTOR(dummy),          // 4    10
    .,                      .
    .,                      .
    .,                      .
    VECTOR(dummy),          // 30   78
    VECTOR(dummy),          // 31   7c
}

/* Dummy Interrupt function */
void dummy(void) {
    /* infinite loop */
    while(1) {
    }
}

/* Non Maskable Interrupt function */
void nmi(void) {
    /* Processing is undecided. */
}

```

2.2.1 Explanation of Vector Table

The “vector table” stores an array of vectors (destination addresses) to each interrupt process routine that is executed if an interrupt occurs during program execution.

Table 2.1 Configuration of Vector Table

Vector No. or software interrupt No.	interrupt	Vector address
0 (0x00)	Reset	TTBR+0x00
1 (0x01)	Address misaligned interrupt	TTBR+0x04
2 (0x02)	NMI	TTBR+0x08
3 (0x03)	Maskable external interrupt 3	TTBR+0x0c
:	:	:
31 (0x1f)	Maskable external interrupt 31	TTBR+0x7c

The “TTBR” (Trap Table Base Register) shown on Table 2.1 identifies the start address of the vector table.

Note: As the TTBR value depends on the processor model used, refer to the corresponding Technical Manual for the actual TTBR.

2.2.2 Relationship between Vector Table and Memory Models

When developing a project using the GNU17, you select a memory model and set the address space. As the pointer size varies depending on your settings, you must take care to create a vector table.

There are three (3) types of memory models: REGULAR, MIDDLE and SMALL.

The REGULAR model has a 24-bit long address, and you can use the 16M-byte space. The pointer size is 32 bits in the C program.

The MIDDLE model has a 20-bit long address, and you can use the 1M-byte space. The pointer size is 32 bits in the C program.

The SMALL model has a 16-bit long address, and you can use the 64K-byte space. The pointer size is 16 bits in the C program.

Although the address space increases when the SMALL, MIDDLE and REGULAR memory models are used in this sequence, their coding efficiency is dropped.

Table 2.2 Memory Models and Pointer Size

Memory model	Address size	Address space	Pointer size
REGULAR	24 bits	16M bytes	32 bits
MIDDLE	20 bits	1M bytes	32 bits
SMALL	16 bits	64K bytes	16 bits

From the hardware viewpoint, the vector table is accessed on the 32-bit boundaries. The REGULAR and MIDDLE models have the 32-bit long pointers and there is no problem as shown on Table 2.2. However, you need to consider that the SMALL model has the 16-bit long pointer. List 2.1 gives a coding example of vector table that can be executed correctly even if any memory model is selected.

The following explains the example vector table of List 2.1.

“__POINTER16” indicated by number <1> is a macro that is defined automatically by the compiler if the memory size is SMALL. This causes the “#define VECTOR(vec)0)” to be used for the SMALL model, but the “#define VECTOR(vec) (vec)” to be used for the REGULAR and MIDDLE models.

Next, the following explains how the vector table is replaced by the macro definition of #define statement. The following compares the replacement if “#define VECTOR(vec) ((vec),0)” and “#define VECTOR(vec) (vec)” are used on List 2.1.

List 2.2 After Replacement of “#define VECTOR(vec)”

```

/* vector table */
func *const vector[] = {           // No   Base+
    (boot), 0,                     // 0    00
    (dummy), 0,                     // 1    04
    (nmi), 0,                       // 2    08
    (dummy), 0,                     // 3    0c
    (dummy), 0,                     // 4    10
    ., ., .,
    (dummy), 0,                     // 30   78
    (dummy), 0                      // 31   7c
}

```

The example on List 2.2 has been corrected so that each item of the vector table has 32 bits long. The 16-bit area after the 16-bit pointer is set to 0s and the 32-bit length is set. The pointer is defined in the first half because the S1C17 Series CPU is accessed in the Little Endian format.

List 2.3 After Replacement of “#define VECTOR(vec) vec”

```

/* vector table */
func *const vector[] = {           // No   Base+
    (boot),                          // 0    00
    (dummy),                          // 1    04
    (nmi),                            // 2    08
    (dummy),                          // 3    0c
    (dummy),                          // 4    10
    ., ., .,
    (dummy),                          // 30   78
    (dummy)                          // 31   7c
}

```

On List 2.3, the vector table is configured since the pointer is 32 bits long.

2.3 Interrupts

The S1C17 Core can accept up to 32 types of interrupts. (The first 3 interrupts are used for reset, address misalignment, and NMI.)

The interrupt process routine is called when the corresponding interrupt cause is accepted. You must code the appropriate routine process. As the interrupt cause and setting vary depending on the model used, refer to the corresponding Technical Manual for details.

2.3.1 Prototype Declaration

Declare the prototype of interrupt process routine in the following format.

<type><function-name>__attribute__((interrupt_handler));

List 2.4 Prototype Declaration Example

```
/* Prototype */  
void dummy(void) __attribute__((interrupt_handler));
```

If the prototype declaration is given as shown on List 2.4, a process to restore the PSR (Process Status Register) and PC (Program Counter) contents at the end of interrupt process, that have been saved by the hardware, is included automatically.

To call the interrupt process routine, you need to specify each interrupt process routine for each vector (destination address) of the vector table. On List 2.1, if an interrupt is not used, control jumps to the “dummy” function.

2.3.2 Reset

A reset interrupt occurs at system power-on. During the reset process, the reset vector is called from the beginning of vector table and it is set on the PC. This allows control to jump to the startup routine of reset vector and to execute the program.

2.3.3 Address Misalignment

A load instruction to access to memory or I/O area has the fixed size of data to be transferred by the instruction. Its address must be on a boundary of each data size.

Table 2.3 Load Instructions and Address Boundaries

Instruction	Transfer data size	Address
ld.b / ld.ub	Bytes (8 bits)	Byte boundaries (for all addresses)
ld	16 bits	16-bit boundaries (The least significant bit of address is 0.)
ld.a	32 bits	32-bit boundaries (The low-order 2 bits of address are 00.)

If the specified address of the load instruction does not meet these conditions, the processor considers an address misaligned interrupt and transitions to the interrupt process.

On List 2.1, if an address misaligned interrupt occurs, control jumps to the “dummy” function and starts the indefinite loop process. Correct it appropriately.

2.3.4 NMI

There are two types of interrupts: maskable interrupts and non-maskable interrupts.

A non-maskable interrupt is shorted as NMI. The CPU accepts this NMI unconditionally in preference to other interrupts.

On List 2.1, if an NMI occurs, control jumps to the “nmi” function and the system is restored automatically. Correct it appropriately.

2.4 Creating a Startup Routine

List 2.5 An Example of Startup Routine

```

/* #include */
#include <string.h>

/* Prototype */
void boot(void);
void clearBss(void);
void copyLmaToVma(void);

/* extern */
extern int main(void);
extern unsigned char __START_bss;
extern unsigned char __END_bss;
extern unsigned char __START_data;
extern unsigned char __START_data_lma;
extern unsigned char __END_data;

/* Boot function */
void boot(void) {
    asm("xld.a %sp, 0x0fc0"); // Set SP in RAM
    clearBss();              // clear the bss area
    copyLmaToVma();          // copy LMA data to VMA data
    asm("ei");               // interrupt enable
    main();                  // Call main
}

/* ClearBss function */
void clearBss(void) {
    memset(&__START_bss, 0, (size_t) (&__END_bss - &__START_bss));
}

/* copyLmaToVma function */
void copyLmaToVma(void) {
    memcpy(&__START_data, &__START_data_lma, (size_t) (&__END_data - &__START_data));
}

```


2.4.1 Explanation of Startup Routine

The startup routine specifies that a reset interrupt occurs during system power-on (at initial reset) and that the function is called from the vector table that corresponds to the interrupt.

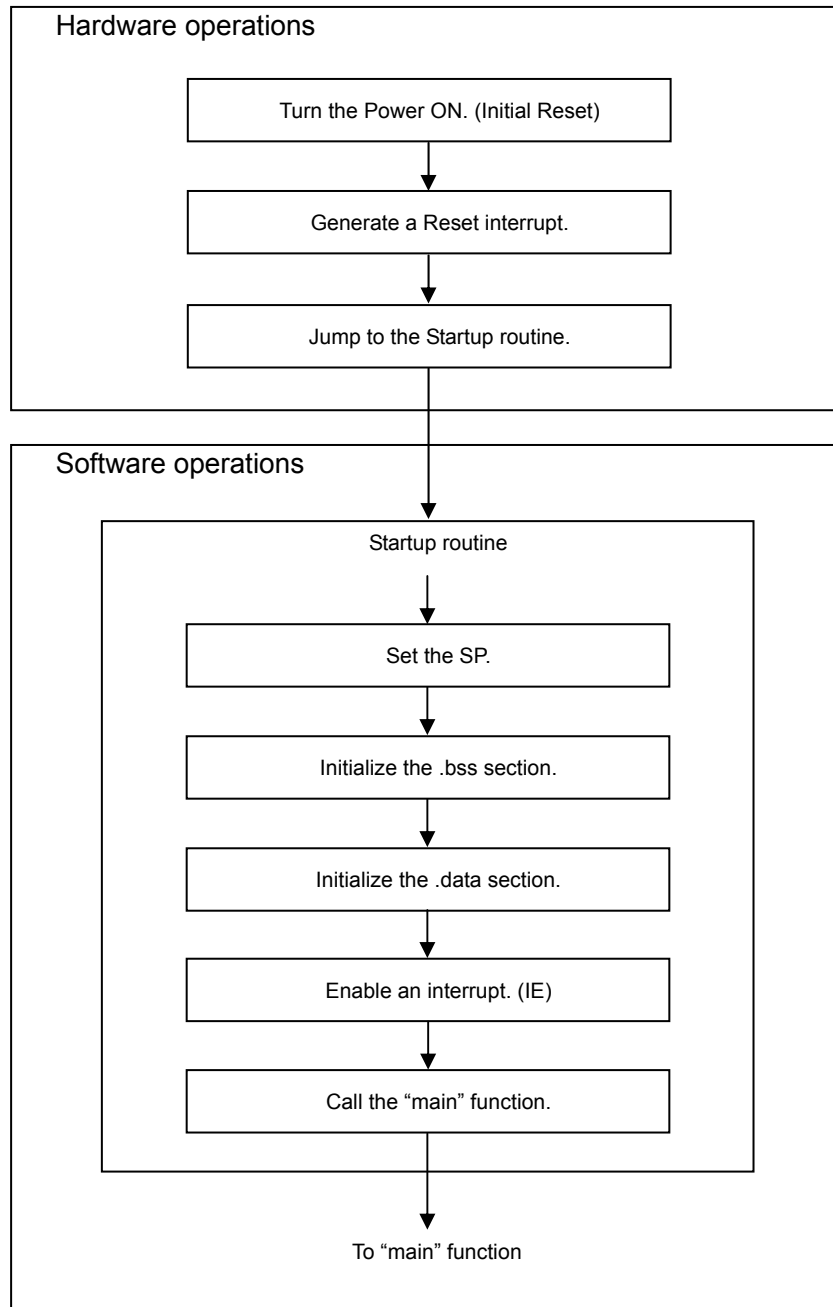


Fig.2.1 Operation Flow of Startup Routine

2.4.2 Setting the SP

First, set the start address of the stack in the SP (Stack Pointer) by issuing the “ld” instruction of the assembler.

List 2.6 SP Setup Example

```
asm("xld.a %sp, 0x0fc0"); // Set SP in RAM
```

Although address “0x0fc0” is set on List 2.6, you can set an address of any RAM area. Take care not to overlap the stack area by the RAM data storage area.

The following explains the SP setting by considering that address 0x0fc0 is set on List 2.6.

The S1C17 Series CPU places the stacks in the smaller address direction. As the S1C17701 has the RAM area of 0x0000 to 0x1000, the maximum value you can set for the SP is 0x1000. However, the 0x0fc0 to 0x0fff area is reserved for on-chip debugging. Therefore, address 0x0fc0 is set to avoid the area overlapping.

Note: For the address of RAM area and on-chip debugger area, refer to the corresponding Technical Manual.

The following gives a reference chart to save registers in the stack and to transition to the SP setting.

Example: ld.a -[%SP],%r0

Explanation of instructions: The stack pointer value is decremented by 4 bytes, and the 24-bit data of “r0” register is transferred to its address. The 32-bit data, having the high-order 8 bits of all 0s, is written in the memory.

(1) SP=SP-4

(2) R0→[SP]

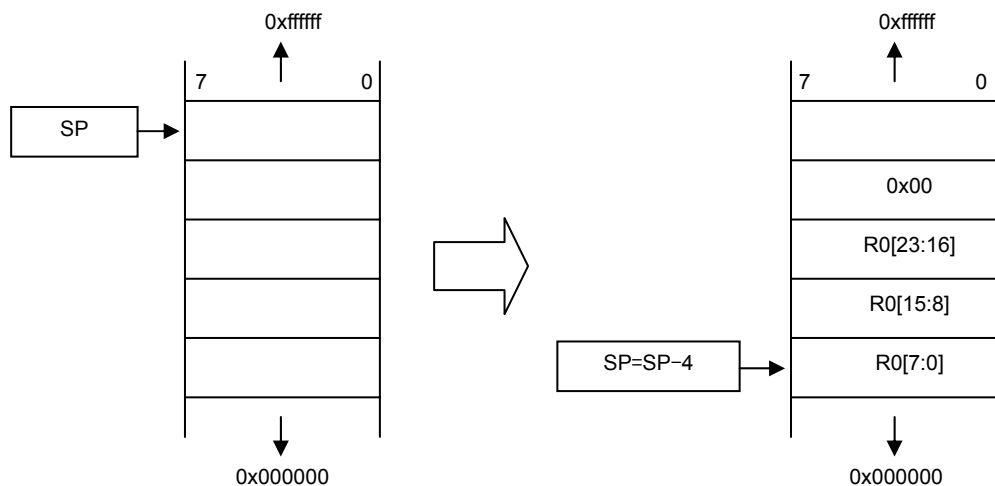


Fig.2.2 SP and Stack

2.4.3 Initializing the .bss/.data Section

Before explaining the initialization of “.bss/.data sections,” the following explains the memory configuration of the project you have created with GNU17. Fig.2.4 shows the memory configuration of the S1C17701.

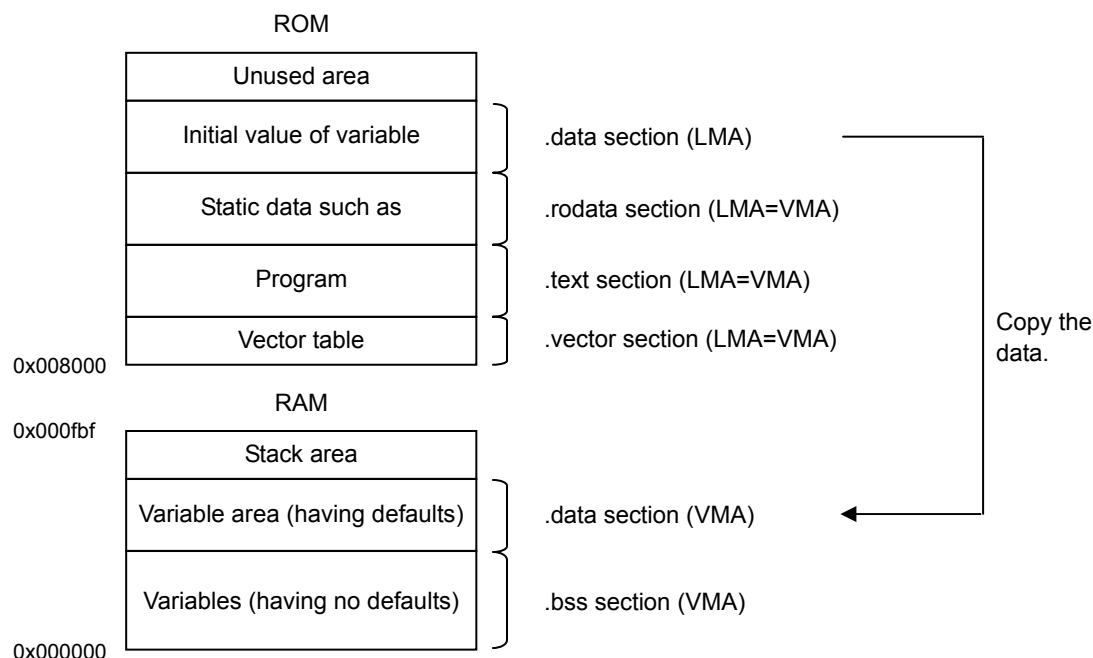


Fig.2.2 Memory Configuration Example (S1C17701)

Place the program and data in the ROM being assigned at address 0x8000 and later as shown in Fig.2.4. The program is assumed to be executed as it is in the storage address (LMA) of the ROM. Also, the static data is assumed to be read directly from the ROM and used.

Place the variable area (having no defaults) in the RAM at address 0x0 and later, and use it as the variable area (having defaults) later. Store the defaults of variables in the ROM, and the application program copies them to the RAM.

For more information about these sections, refer to the “S5U1C17001C MANUAL.”

2. PROGRAMMING OF S1C17 FAMILY PROCESSORS

The following explains how to initialize the .bss section.

The “.bss section” stores variables having no defaults. The area from “__START_bss” to “__END_bss” is set to all 0s, and its data is cleared.

List 2.7 Initializing the .bss Section

```
/* #include */
#include <string.h>

/* extern */
extern unsigned char __START_bss;
extern unsigned char __END_bss;

/* ClearBss function */
void clearBss (void) {
    memset(&__START_bss, 0 , (size_t) (&__END_bss - &__START_bss));
}
```

The sample program of List 2.7 uses the “memset” function and clears the data by setting the area between “START_bss” and “END_bss” to zero (0).

The “START_bss” and “END_bss” of external reference are defined in the “Linker script file (file.lds).”

__START_bss The start address of .bss section
__END_bss The end address of .bss section

Note: When using the “memset” function, include the “string.h” file.

List 2.8 Initializing the .data Section

```
/* #include */
#include <string.h>

/* extern */
extern unsigned char __START_data;
extern unsigned char __START_data_lma;
extern unsigned char __END_data;

/* copyLmaToVma function */
void copyLmaToVma (void) {
    memcpy(&__START_data, &__START_data_lma, (size_t) (&__END_data - &__START_data));
}
```

The sample program of List 2.8 uses the “memcpy” function and copies data of “START_data_lma” to the area between “START_data” and “END_data” in the ROM (VMA).

The “START_data” “START_data_lma” and “__END_data” of external reference are defined in the “Linker script file (file.lds).”

__START_data The start address of .data section
__START_data_lma The start address of .data section LMA part
__END_data The end address of .data section

Note: When using the “memcpy” function, include the “string.h” file.

2.4.4 Enabling an Interrupt (IE)

The IE (Interrupt Enable) bit of the PSR (Processor Status Register) is set to 1 by the “ei” instruction of the assembler, and an external maskable interrupt is enabled.

List 2.9 Enabling an Interrupt (IE)

```
asm("ei");           // interrupt enable
```

PSR is an 8-bit register that holds the CPU status data, and its content varies depending on the executed instruction result. Except for the IE bit state, you cannot directly change the content of this PSR using the program.

	7	6	5	4	3	2	1	0
PSR	IL [2:0]			IE	C	V	Z	N
Default	0	0	0	0	0	0	0	0

IL:	Interrupt level	(0 to 7: Interrupt)
IE:	Interrupt enabled	(1: Enabled; 0: Disabled)
C:	Carry flag	(1: With carry/borrow; 0: None)
V:	Overflow flag	(1: Overflowed; 0: None)
Z:	Zero flag	(1: Zero, 0: None-zero)
N:	Negative flag	(1: Negative; 0: Positive)

Fig.2.3 PSR

This is a reference information only. To disable an external maskable interrupt, use the “di” instruction of the assembler as shown on List 2.10.

List 2.10 Disabling an Interrupt

```
asm("di");           // interrupt disable
```

3. “volatile” QUALIFIER

An access to a variable by the compiler or program is affected by the “volatile” qualifier. Generally, the C compiler can efficiently create programming codes at a higher speed. Therefore, the process is optimized by reducing the access frequency to memory and by reusing the values loaded in registers as many as possible. However, this process optimization may cause a skip of actual memory access that is written in the program. In such case, the control register values changed by the hardware and the variables changed by the interrupt processing may be processed without reflection of those changed values. The “volatile” qualifier specifies the compiler to get and update the variable contents each time they are referenced to, by assuming that those variables may have been changed. In other words, if a variable is defined with the “volatile” declaration, a memory access always occurs during referencing. The variable is updated to the latest value and processed.

List 3.1 below compares the process if the “volatile” qualifier is declared and if it is not declared.

List 3.1 A coding example if “volatile” qualifier is declared and if not declared

```

/* prototype */
int main(void);
void sample_Int(void) __attribute__((interrupt_handler));

/* Global variable define*/
int Normal_flg;           // volatile no declaration
volatile int Volatile_flg; // volatile declaration

int main(void){
    Normal_flg = 0;
    /* (1)volatile no declaration */
    while(Normal_flg == 1 ){
        }

    Volatile_flg = 0;
    /* (2)volatile declaration */
    while(Volatile_flg == 2 ){
        }

    return 0;
}

/* sample interrupt handler */
void sample_Int(void){
    Normal_flg = 1;
    Volatile_flg = 2;
    return;
}

```

(1) Explanation if the variable without “volatile” declaration is converted into the assembler code

```

ld  %r2,[0x0]  ← Transfers a value of address 0x0 (Normal_flg) to r2.
cmp  %r2,0x1   ← Compares the r2 and 0x1 values.
jne  0x7e      ← Branches to the arrow point if the above comparison result is not equal to each other.

```

Because the actual memory data is not referenced to even if the variable value is changed by the interrupt process routine (sample_Int function), the register value is not updated and control

(2) Explanation if the variable with “volatile” declaration is converted into the assembler coder

```

ld  %r2,[0x2]  ← Assigns a value of address 0x2 (Volatile_flg) into r2.
cmp  %r2,0x2   ← Compares the r2 and 0x2 values.
jne  0x7d      ← Branches to the arrow point if the above comparison result is not equal to each other.

```

If a variable value is changed by the interrupt process routine (sample_Int function), the memory value is always reflected on it during referencing and, therefore, control cannot exit the while statement.

Note: List 3.1 gives an example where the “-O1” optimize option is used for compiling.

4. HOW TO ACCESS TO I/O REGISTERS

4.1 Read/Write of I/O Registers

To read data from I/O registers, code the program as shown on List 3.2

List 3.2 Data reading from I/O registers

```
(unsigned char )I = *(volatile unsigned char *) ( 0x4020 )      ←For 8-bit device
(unsigned short )j = *(volatile unsigned short *) ( 0x4200 )    ←For 16-bit device
```

The program of List 3.2 transfers 8-bit data of address 0x4020 to variable “i”, and it transfers 16-bit data of address 0x4200 to variable “j”.

To write data in I/O registers, code the program as shown on List 3.3

List 3.3 Data writing in I/O registers

```
*(volatile unsigned char *) ( 0x4020 ) = (unsigned char )0x01    ←For 8-bit device
*(volatile unsigned short *) ( 0x4200 ) = (unsigned short )0x0001 ←For 16-bit device
```

The program of List 3.3 writes value “0x01” in address 0x4020, and it writes value “0x0001” in address 0x4200.

Note: Remember the device size of peripheral circuits when you code a program to access to I/O registers. For information about the peripheral circuit device size, refer to the technical document of the corresponding model.

4. HOW TO ACCESS TO I/O REGISTERS

4.2 Bit Fields

A bit field can also be used for I/O registers.

List 3.4 gives a sample coding that uses the bit field for I/O registers (having device size 8). Bit 0 of address 0x4020 is set to logical 1 on List 3.4.

List 3.4 A sample coding to use the bit field

```
/* Bit Field For Byte Data. */
struct BbitF {
    unsigned int b0 : 1;
    unsigned int b1 : 1;
    unsigned int b2 : 1;
    unsigned int b3 : 1;
    unsigned int b4 : 1;
    unsigned int b5 : 1;
    unsigned int b6 : 1;
    unsigned int b7 : 1;
};

/* Sample Register */
#define BF_SAMPLE (*(volatile struct BbitF *) ( 0x4020))

/* Sample Register writing */
BF_SAMPLE.b0 = 1;
```

Explanation after data conversion into assembler code

xld.a	%r3, 0x4020	Transfers value 0x4020 to r3 .
ld.b	%r2, [%r3]	← Transfers value of r3 (address 0x4020) to r2 .
or	%r2, 0x1	← Transfers value r2 exclusively ORed with 0x1 (bit 0) to r2 .
ld.b	[%r3], %r2	← Transfers value of r3 (address 0x4020) to r2 .

Remember that if you use the bit field to access to an I/O register, the register is accessed in unit of bytes. For details, refer to the “Data the detailed information, refer to the “Data representation” section of the “Compile Output” in the “S5U1C17001C Manual.”

Do not try to write data in I/O registers using the bit field if a register is reset if a register bit is set to logical 1. If you use such I/O registers, access to them in the method as described in Section 4.1.

List 3.5 gives a sample coding which uses the bit field and writes data in an I/O register if the register is reset when a register bit is set to logical 1.

List 3.5 A coding example

```
/* If address 0x4020 (the I/O register which is reset to 0 if register bit is set to logical 1) is 0xaa */
xld.a    %r3, 0x4020
ld.b     %r2, [%r3]
or       %r2, 0x1
ld.b     [%r3], %r2
```

←	r2 value (01010101)
←	r2 value (01010101)
←	r3 value (01010101) ⇒ r3 value (00000000)

The sample coding of List 3.5 resets all logical-1 bits to zero (0) although you wish to reset only bit 0 of address 0x4020.

REVISION HISTORY

[illegible]

AMERICA

EPSON ELECTRONICS AMERICA, INC.**HEADQUARTERS**

2580 Orchard Parkway
San Jose, CA 95131, USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

SALES OFFICES**Northeast**

301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 FAX: +1-781-246-5443

EUROPE

EPSON EUROPE ELECTRONICS GmbH**HEADQUARTERS**

Riesstrasse 15 Muenchen Bayern,
80992 GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.

7F, Jinbao Bldg., No. 89 Jinbao St.,
Dongcheng District,
Beijing 100005, China
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH

7F, Block B, Hi-Tech Bldg., 900, Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

EPSON HONG KONG LTD.

20/F., Harbour Centre, 25 Harbour Road
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON (CHINA) CO., LTD.**SHENZHEN BRANCH**

12/F, Dawning Mansion, Keji South 12th Road,
Hi-Tech Park, Shenzhen
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.

14F, No. 7, Song Ren Road,
Taipei 110
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.

1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORPORATION**KOREA OFFICE**

50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

GUMI OFFICE

2F, Grand B/D, 457-4 Songjeong-dong,
Gumi-City, KOREA
Phone: +82-54-454-6027 FAX: +82-54-454-6093

SEIKO EPSON CORPORATION**SEMICONDUCTOR OPERATIONS DIVISION****IC Sales Dept.****IC International Sales Group**

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

■ EPSON Electronic Devices Website

http://www.epson.jp/device/semicon_e/