EPSON

CMOS 32-BIT SINGLE CHIP MICROCONTROLLER

S1C31W65
Technical Manual

arm

SEIKO EPSON CORPORATION

Rev. 1.2

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

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

1. This document may not be copied, reproduced, or used for any other purpose, in whole or in part, without the consent of the Seiko Epson Corporation (“Epson”).
2. Before purchasing or using Epson products, please contact our sales representative for the latest information and always be sure to check the latest information published on Epson’s official web sites and other sources.
3. Information provided in this document such as application circuits, programs, usage, etc., are for reference purposes only. Using the application circuits, programs, usage, etc. in the design of your equipment or systems is your own responsibility. Epson makes no guarantees against any infringements or damages to any third parties’ intellectual property rights or any other rights resulting from the information. This document does not grant you any licenses, intellectual property rights or any other rights with respect to Epson products owned by Epson or any third parties.
4. Epson is committed to constantly improving quality and reliability, but semiconductor products in general are subject to malfunction and failure. By using Epson products, you shall be responsible for your hardware. Software and systems must be designed well enough to prevent death or injury as well as any property damage even if any of the malfunctions or failures might be caused by Epson products. When designing your products using Epson products, please be sure to check and comply with the latest information regarding Epson products (this document, specifications, data sheets, manuals, Epson’s web site, etc.). When using the information included above materials such as product data, charts, technical contents, programs, algorithms and application circuit examples, you shall evaluate your products both on a stand-alone basis as well as within your overall systems. You shall be solely responsible for deciding whether or not to adopt and use Epson products.
5. Epson has prepared this document and programs provided in this document carefully to be accurate and dependable, but Epson does not guarantee that the information and the programs are always accurate and complete. Epson assumes no responsibility for any damages which you incur due to misinformation in this document and the programs.
6. No dismantling, analysis, reverse engineering, modification, alteration, adaptation, reproduction, etc., of Epson products is allowed.
7. Epson products have been designed, developed and manufactured to be used in general electronic applications (office equipment, communications equipment, measuring instruments, home electronics, etc.) (“General Purpose”) and applications which is individually listed in this document or designated by Epson (“Designated Purpose”). Epson products are NOT intended for any use beyond the General Purpose and Designated Purpose uses that requires particular/higher quality or reliability in order to refrain from causing any malfunction or failure leading to death, injury, serious property damage or severe impact on society, including, but not limited to those listed below (“Particular Purpose”). Therefore, you are advised to use Epson products only for General Purpose and Designated Purpose uses. Should you desire to buy and use Epson products for a Particular Purpose, Epson makes no warranty and disclaims with respect to Epson products, whether express or implied, including without limitation any implied warranty of merchantability or fitness for any Particular Purpose. Please be sure to contact our sales representative and obtain approval in advance.

   [Examples of Particular Purpose]
   Space equipment (artificial satellites, rockets, etc.) /
   Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.) /
   Medical equipment / Relay equipment to be placed on ocean floor /
   Power station control equipment / Disaster or crime prevention equipment / Traffic control equipment / Financial equipment

   Other applications requiring similar levels of reliability as those listed above. Please be sure to contact our sales representative for details of the other applications.
8. Epson products listed in this document and our associated technologies shall not be used in any equipment or systems that laws and regulations in Japan or any other countries prohibit to manufacture, use or sell. Furthermore, Epson products and our associated technologies shall not be used for developing weapons of mass destruction, or any other military purposes or applications. If exporting Epson products or our associated technologies, you shall comply with the Foreign Exchange and Foreign Trade Control Act in Japan, Export Administration Regulations in the U.S.A. (EAR) and other export-related laws and regulations in Japan and any other countries and follow the required procedures as provided by the relevant laws and regulations.
9. Epson assumes no responsibility for any damages (whether direct or indirect) caused by or in relation with your non-compliance with the terms and conditions in this document.
10. Epson assumes no responsibility for any damages (whether direct or indirect) incurred by any third party that you assign, transfer, loan, etc., Epson products to.
11. For more details or other concerns about this document, please contact our sales representative.
12. Company names and product names listed in this document are trademarks or registered trademarks of their respective companies.

Evaluation board/kit and Development tool important notice

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

Rev. e1.4, 2023. 4

Preface

This is a technical manual for designers and programmers who develop a product using the S1C31W65. This document describes the functions of the IC, embedded peripheral circuit operations, and their control methods.

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

Low power mode

This manual describes the low power modes as HALT mode and SLEEP mode. These terms refer to sleep mode and deep sleep mode in the Cortex®-M0+ processor, respectively.

– 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 Pin Configuration Diagram 1-4
1.3.2 Pin Descriptions 1-5

2 Power Supply, Reset, and Clocks 2-1
2.1 Power Generator (PWGA) 2-1
2.1.1 Overview 2-1
2.1.2 Pins 2-1
2.1.3 V_{D1} Regulator Operation Mode 2-2
2.1.4 V_{D1} Regulator Voltage Mode 2-2
2.2 System Reset Controller (SRC) 2-3
2.2.1 Overview 2-3
2.2.2 Input Pin 2-4
2.2.3 Reset Sources 2-4
2.2.4 Reset Request Flag 2-5
2.2.5 #RESET Pin Input Control 2-5
2.2.6 Initialization Conditions (Reset Groups) 2-6
2.3 Clock Generator (CLG) 2-7
2.3.1 Overview 2-7
2.3.2 Input/Output Pins 2-8
2.3.3 Clock Sources 2-8
2.3.4 Operations 2-11
2.4 Operating Mode 2-16
2.4.1 Initial Boot Sequence 2-16
2.4.2 Transition between Operating Modes 2-16
2.5 Interrupts 2-18
2.6 Control Registers 2-18
PWGA Control Register 2-18
SRC Reset Request Flag Register 2-19
SRC #RESET Port Control Register 2-19
CLG System Clock Control Register 2-20
CLG Oscillation Control Register 2-21
CLG IOSC Control Register 2-22
CLG OSC1 Control Register 2-23
CLG OSC3 Control Register 2-24
CLG Interrupt Flag Register 2-25
CLG Interrupt Enable Register 2-26
CLG FOUT Control Register 2-26
CLG Oscillation Frequency Trimming Register 1 2-27
CLG Oscillation Frequency Trimming Register 2 2-28

3 CPU and Debugger 3-1
3.1 Overview 3-1
3.2 CPU 3-1
3.3 Debugger 3-1
3.3.1 List of Debugger Input/Output Pins 3-1
3.3.2 External Connection 3-1

3.4 Reference Documents 3-2

4 Memory and Bus 4-1

4.1 Overview 4-1
4.2 Bus Access Cycle 4-2
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 4-2
4.4 RAM 4-3
4.5 Display Data RAM 4-3
4.6 Peripheral Circuit Control Registers 4-3
4.6.1 System-Protect Function 4-9
4.7 Control Registers 4-9
System Protect Register 4-9
FLASHC Flash Read Cycle Register 4-10

5 Interrupt 5-1

5.1 Overview 5-1
5.2 Vector Table 5-1
5.2.1 Vector Table Offset Address (VTOR) 5-3
5.2.2 Priority of Interrupts 5-3
5.3 Peripheral Circuit Interrupt Control 5-3
5.4 NMI 5-3

6 DMA Controller (DMAC) 6-1

6.1 Overview 6-1
6.2 Operations 6-2
6.2.1 Initialization 6-2
6.3 Priority 6-2
6.4 Data Structure 6-2
6.4.1 Transfer Source End Pointer 6-3
6.4.2 Transfer Destination End Pointer 6-3
6.4.3 Control Data 6-4
6.5 DMA Transfer Mode 6-5
6.5.1 Basic Transfer 6-5
6.5.2 Auto-Request Transfer 6-5
6.5.3 Ping-Pong Transfer 6-6
6.5.4 Memory Scatter-Gather Transfer 6-7
6.5.5 Peripheral Scatter-Gather Transfer 6-8
6.6 DMA Transfer Cycle 6-9
6.7 Interrupts 6-9
6.8 Control Registers 6-10
DMAC Status Register 6-10
DMAC Configuration Register 6-10
DMAC Control Data Base Pointer Register 6-11
DMAC Alternate Control Data Base Pointer Register 6-11
DMAC Software Request Register 6-11
DMAC Request Mask Set Register 6-11
DMAC Request Mask Clear Register 6-12
DMAC Enable Set Register 6-12
DMAC Enable Clear Register 6-12
DMAC Primary-Alternate Set Register 6-12
DMAC Primary-Alternate Clear Register 6-13

DMAC Priority Set Register ... 6-13
DMAC Priority Clear Register ... 6-13
DMAC Error Interrupt Flag Register ... 6-13
DMAC Transfer Completion Interrupt Flag Register ... 6-14
DMAC Transfer Completion Interrupt Enable Set Register ... 6-14
DMAC Transfer Completion Interrupt Enable Clear Register ... 6-14
DMAC Error Interrupt Enable Set Register ... 6-14
DMAC Error Interrupt Enable Clear Register ... 6-15

7 I/O Ports (PPORT) ... 7-1
7.1 Overview ... 7-1
7.2 I/O Cell Structure and Functions ... 7-2
7.2.1 Schmitt Input ... 7-2
7.2.2 Over Voltage Tolerant Fail-Safe Type I/O Cell ... 7-2
7.2.3 Pull-Up/Pull-Down ... 7-2
7.2.4 CMOS Output and High Impedance State ... 7-3
7.3 Clock Settings ... 7-3
7.3.1 PPORT Operating Clock ... 7-3
7.3.2 Clock Supply in SLEEP Mode ... 7-3
7.3.3 Clock Supply During Debugging ... 7-3
7.4 Operations ... 7-3
7.4.1 Initialization ... 7-3
7.4.2 Port Input/Output Control ... 7-5
7.5 Interrupts ... 7-6
7.6 Control Registers ... 7-6
P*x* Port Data Register ... 7-6
P*x* Port Enable Register ... 7-7
P*x* Port Pull-up/down Control Register ... 7-7
P*x* Port Interrupt Flag Register ... 7-8
P*x* Port Interrupt Control Register ... 7-8
P*x* Port Chattering Filter Enable Register ... 7-8
P*x* Port Mode Select Register ... 7-8
P*x* Port Function Select Register ... 7-9
P Port Clock Control Register ... 7-9
P Port Interrupt Flag Group Register ... 7-10
7.7 Control Register and Port Function Configuration of this IC ... 7-11
7.7.1 P0 Port Group ... 7-11
7.7.2 P1 Port Group ... 7-12
7.7.3 P2 Port Group ... 7-13
7.7.4 P3 Port Group ... 7-14
7.7.5 P4 Port Group ... 7-15
7.7.6 P5 Port Group ... 7-16
7.7.7 P6 Port Group ... 7-17
7.7.8 Pd Port Group ... 7-18
7.7.9 Common Registers between Port Groups ... 7-19

8 Universal Port Multiplexer (UPMUX) ... 8-1
8.1 Overview ... 8-1
8.2 Peripheral Circuit I/O Function Assignment ... 8-1
8.3 Control Registers ... 8-2
P*xy*–*xz* Universal Port Multiplexer Setting Register ... 8-2

9 Watchdog Timer (WDT2) ... 9-1
9.1 Overview ... 9-1
9.2 Clock Settings ... 9-1

9.2.1 WDT2 Operating Clock 9-1
9.2.2 Clock Supply in DEBUG Mode 9-1
9.3 Operations 9-2
9.3.1 WDT2 Control 9-2
9.3.2 Operations in HALT and SLEEP Modes 9-3
9.4 Control Registers 9-3
WDT2 Clock Control Register 9-3
WDT2 Control Register 9-4
WDT2 Counter Compare Match Register 9-5
10 Real-Time Clock (RTCA) 10-1
10.1 Overview 10-1
10.2 Output Pin and External Connection 10-1
10.2.1 Output Pin 10-1
10.3 Clock Settings 10-2
10.3.1 RTCA Operating Clock 10-2
10.3.2 Theoretical Regulation Function 10-2
10.4 Operations 10-3
10.4.1 RTCA Control 10-3
10.4.2 Real-Time Clock Counter Operations 10-4
10.4.3 Stopwatch Control 10-4
10.4.4 Stopwatch Count-up Pattern 10-4
10.5 Interrupts 10-5
10.6 Control Registers 10-6
RTCA Control Register (Low Byte) 10-6
RTCA Control Register (High Byte) 10-7
RTCA Second Alarm Register 10-7
RTCA Hour/Minute Alarm Register 10-8
RTCA Stopwatch Control Register 10-8
RTCA Second/1Hz Register 10-9
RTCA Hour/Minute Register 10-10
RTCA Month/Day Register 10-11
RTCA Year/Week Register 10-11
RTCA Interrupt Flag Register 10-12
RTCA Interrupt Enable Register 10-13
11 Supply Voltage Detector (SVD4) 11-1
11.1 Overview 11-1
11.2 Input Pins and External Connection 11-2
11.2.1 Input Pins 11-2
11.2.2 External Connection 11-2
11.3 Clock Settings 11-2
11.3.1 SVD4 Operating Clock 11-2
11.3.2 Clock Supply in SLEEP Mode 11-2
11.3.3 Clock Supply in DEBUG Mode 11-3
11.4 Operations 11-3
11.4.1 SVD4 Control 11-3
11.4.2 SVD4 Operations 11-4
11.5 SVD4 Interrupt and Reset 11-4
11.5.1 SVD4 Interrupt 11-4
11.5.2 SVD Reset 11-5
11.6 Control Registers 11-5
SVD4 Ch.*n* Clock Control Register 11-5
SVD4 Ch.*n* Control Register 11-6

SVD4 Ch.*n* Status and Interrupt Flag Register 11-7
SVD4 Ch.*n* Interrupt Enable Register 11-8

12 16-bit Timers (T16) 12-1
12.1 Overview 12-1
12.2 Input Pin 12-1
12.3 Clock Settings 12-2
12.3.1 T16 Operating Clock 12-2
12.3.2 Clock Supply in SLEEP Mode 12-2
12.3.3 Clock Supply During Debugging 12-2
12.3.4 Event Counter Clock 12-2
12.4 Operations 12-2
12.4.1 Initialization 12-2
12.4.2 Counter Underflow 12-3
12.4.3 Operations in Repeat Mode 12-3
12.4.4 Operations in One-shot Mode 12-3
12.4.5 Counter Value Read 12-4
12.5 Interrupt 12-4
12.6 Control Registers 12-4
T16 Ch.*n* Clock Control Register 12-4
T16 Ch.*n* Mode Register 12-5
T16 Ch.*n* Control Register 12-5
T16 Ch.*n* Reload Data Register 12-6
T16 Ch.*n* Counter Data Register 12-6
T16 Ch.*n* Interrupt Flag Register 12-6
T16 Ch.*n* Interrupt Enable Register 12-7

13 UART (UART3) 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 Input Pin Pull-Up Function 13-2
13.2.4 Output Pin Open-Drain Output Function 13-2
13.2.5 Input/Output Signal Inverting Function 13-2
13.3 Clock Settings 13-2
13.3.1 UART3 Operating Clock 13-2
13.3.2 Clock Supply in SLEEP Mode 13-3
13.3.3 Clock Supply During Debugging 13-3
13.3.4 Baud Rate Generator 13-3
13.4 Data Format 13-3
13.5 Operations 13-4
13.5.1 Initialization 13-4
13.5.2 Data Transmission 13-5
13.5.3 Data Reception 13-6
13.5.4 IrDA Interface 13-7
13.5.5 Carrier Modulation 13-8
13.6 Receive Errors 13-9
13.6.1 Framing Error 13-9
13.6.2 Parity Error 13-9
13.6.3 Overrun Error 13-9
13.7 Interrupts 13-10
13.8 DMA Transfer Requests 13-10
13.9 Control Registers 13-11

UART3 Ch.*n* Clock Control Register 13-11
UART3 Ch.*n* Mode Register 13-11
UART3 Ch.*n* Baud–Rate Register 13-13
UART3 Ch.*n* Control Register 13-13
UART3 Ch.*n* Transmit Data Register 13-14
UART3 Ch.*n* Receive Data Register 13-14
UART3 Ch.*n* Status and Interrupt Flag Register 13-14
UART3 Ch.*n* Interrupt Enable Register 13-15
UART3 Ch.*n* Transmit Buffer Empty DMA Request Enable Register 13-16
UART3 Ch.*n* Receive Buffer One Byte Full DMA Request Enable Register 13-16
UART3 Ch.*n* Carrier Waveform Register 13-16

14 Synchronous Serial Interface (SPIA) 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
14.2.3 Pin Functions in Master Mode and Slave Mode 14-3
14.2.4 Input Pin Pull-Up/Pull-Down Function 14-3
14.3 Clock Settings 14-3
14.3.1 SPIA Operating Clock 14-3
14.3.2 Clock Supply During Debugging 14-4
14.3.3 SPI Clock (SPICLK*n*) Phase and Polarity 14-4
14.4 Data Format 14-5
14.5 Operations 14-5
14.5.1 Initialization 14-5
14.5.2 Data Transmission in Master Mode 14-6
14.5.3 Data Reception in Master Mode 14-8
14.5.4 Terminating Data Transfer in Master Mode 14-10
14.5.5 Data Transfer in Slave Mode 14-10
14.5.6 Terminating Data Transfer in Slave Mode 14-11
14.6 Interrupts 14-12
14.7 DMA Transfer Requests 14-13
14.8 Control Registers 14-13
SPIA Ch.*n* Mode Register 14-13
SPIA Ch.*n* Control Register 14-14
SPIA Ch.*n* Transmit Data Register 14-15
SPIA Ch.*n* Receive Data Register 14-15
SPIA Ch.*n* Interrupt Flag Register 14-15
SPIA Ch.*n* Interrupt Enable Register 14-16
SPIA Ch.*n* Transmit Buffer Empty DMA Request Enable Register 14-16
SPIA Ch.*n* Receive Buffer Full DMA Request Enable Register 14-16

15 I²C (I2C) 15-1

15.1 Overview 15-1
15.2 Input/Output Pins and External Connections 15-2
15.2.1 List of Input/Output Pins 15-2
15.2.2 External Connections 15-2
15.3 Clock Settings 15-3
15.3.1 I2C Operating Clock 15-3
15.3.2 Clock Supply During Debugging 15-3
15.3.3 Baud Rate Generator 15-3
15.4 Operations 15-4
15.4.1 Initialization 15-4
15.4.2 Data Transmission in Master Mode 15-5

15.4.3 Data Reception in Master Mode ... 15-7
15.4.4 10-bit Addressing in Master Mode ... 15-10
15.4.5 Data Transmission in Slave Mode ... 15-11
15.4.6 Data Reception in Slave Mode ... 15-13
15.4.7 Slave Operations in 10-bit Address Mode ... 15-15
15.4.8 Automatic Bus Clearing Operation ... 15-15
15.4.9 Error Detection ... 15-16
15.5 Interrupts ... 15-17
15.6 DMA Transfer Requests ... 15-18
15.7 Control Registers ... 15-18
I2C Ch.*n* Clock Control Register ... 15-18
I2C Ch.*n* Mode Register ... 15-19
I2C Ch.*n* Baud-Rate Register ... 15-19
I2C Ch.*n* Own Address Register ... 15-20
I2C Ch.*n* Control Register ... 15-20
I2C Ch.*n* Transmit Data Register ... 15-21
I2C Ch.*n* Receive Data Register ... 15-21
I2C Ch.*n* Status and Interrupt Flag Register ... 15-22
I2C Ch.*n* Interrupt Enable Register ... 15-23
I2C Ch.*n* Transmit Buffer Empty DMA Request Enable Register ... 15-24
I2C Ch.*n* Receive Buffer Full DMA Request Enable Register ... 15-24

16 16-bit PWM Timers (T16B) ... 16-1
16.1 Overview ... 16-1
16.2 Input/Output Pins ... 16-2
16.3 Clock Settings ... 16-3
16.3.1 T16B Operating Clock ... 16-3
16.3.2 Clock Supply in SLEEP Mode ... 16-3
16.3.3 Clock Supply During Debugging ... 16-3
16.3.4 Event Counter Clock ... 16-3
16.4 Operations ... 16-4
16.4.1 Initialization ... 16-4
16.4.2 Counter Block Operations ... 16-5
16.4.3 Comparator/Capture Block Operations ... 16-8
16.4.4 TOUT Output Control ... 16-17
16.5 Interrupt ... 16-23
16.6 DMA Transfer Requests ... 16-23
16.7 Control Registers ... 16-23
T16B Ch.*n* Clock Control Register ... 16-23
T16B Ch.*n* Counter Control Register ... 16-24
T16B Ch.*n* Max Counter Data Register ... 16-25
T16B Ch.*n* Timer Counter Data Register ... 16-25
T16B Ch.*n* Counter Status Register ... 16-26
T16B Ch.*n* Interrupt Flag Register ... 16-27
T16B Ch.*n* Interrupt Enable Register ... 16-28
T16B Ch.*n* Comparator/Capture *m* Control Register ... 16-29
T16B Ch.*n* Compare/Capture *m* Data Register ... 16-31
T16B Ch.*n* Counter Max/Zero DMA Request Enable Register ... 16-32
T16B Ch.*n* Compare/Capture *m* DMA Request Enable Register ... 16-32

17 Sound Generator (SNDA) ... 17-1
17.1 Overview ... 17-1
17.2 Output Pins and External Connections ... 17-2
17.2.1 List of Output Pins ... 17-2
17.2.2 Output Pin Drive Mode ... 17-2

17.2.3 External Connections 17-2
17.3 Clock Settings 17-3
17.3.1 SNDA Operating Clock 17-3
17.3.2 Clock Supply in SLEEP Mode 17-3
17.3.3 Clock Supply in DEBUG Mode 17-3
17.4 Operations 17-3
17.4.1 Initialization 17-3
17.4.2 Buzzer Output in Normal Buzzer Mode 17-3
17.4.3 Buzzer Output in One-shot Buzzer Mode 17-6
17.4.4 Output in Melody Mode 17-7
17.5 Interrupts 17-9
17.6 DMA Transfer Requests 17-10
17.7 Control Registers 17-10
SNDA Clock Control Register 17-10
SNDA Select Register 17-11
SNDA Control Register 17-12
SNDA Data Register 17-12
SNDA Interrupt Flag Register 17-13
SNDA Interrupt Enable Register 17-13
SNDA Sound Buffer Empty DMA Request Enable Register 17-14

18 IR Remote Controller (REMC3) 18-1
18.1 Overview 18-1
18.2 Output Pins and External Connections 18-1
18.2.1 List of Output Pins 18-1
18.2.2 External Connections 18-2
18.3 Clock Settings 18-2
18.3.1 REMC3 Operating Clock 18-2
18.3.2 Clock Supply in SLEEP Mode 18-2
18.3.3 Clock Supply During Debugging 18-2
18.4 Operations 18-2
18.4.1 Initialization 18-2
18.4.2 Data Transmission Procedures 18-3
18.4.3 REMO Output Waveform 18-3
18.4.4 Continuous Data Transmission and Compare Buffers 18-5
18.5 Interrupts 18-6
18.6 Application Example: Driving EL Lamp 18-7
18.7 Control Registers 18-7
REMC3 Clock Control Register 18-7
REMC3 Data Bit Counter Control Register 18-8
REMC3 Data Bit Counter Register 18-9
REMC3 Data Bit Active Pulse Length Register 18-10
REMC3 Data Bit Length Register 18-10
REMC3 Status and Interrupt Flag Register 18-10
REMC3 Interrupt Enable Register 18-11
REMC3 Carrier Waveform Register 18-11
REMC3 Carrier Modulation Control 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 DMA Transfer Requests 19-6
19.7 Control Registers 19-7
ADC12A Ch.*n* Control Register 19-7
ADC12A Ch.*n* Trigger/Analog Input Select Register 19-8
ADC12A Ch.*n* Configuration Register 19-9
ADC12A Ch.*n* Interrupt Flag Register 19-10
ADC12A Ch.*n* Interrupt Enable Register 19-10
ADC12A Ch.*n* DMA Request Enable Register *m* 19-11
ADC12A Ch.*n* Result Register 19-11

20 Temperature Sensor/Reference Voltage Generator (TSRVR) 20-1
20.1 Overview 20-1
20.2 Output Pin and External Connections 20-1
20.2.1 Output Pin 20-1
20.2.2 External Connections 20-2
20.3 Operations 20-2
20.3.1 Reference Voltage Setting 20-2
20.3.2 Temperature Sensor Setting 20-2
20.4 Control Registers 20-3
TSRVR Ch.*n* Temperature Sensor Control Register 20-3
TSRVR Ch.*n* Reference Voltage Generator Control Register 20-3

21 LCD Driver (LCD8D) 21-1
21.1 Overview 21-1
21.2 Output Pins and External Connections 21-2
21.2.1 List of Output Pins 21-2
21.2.2 External Connections 21-2
21.3 Clock Settings 21-3
21.3.1 LCD8D Operating Clock 21-3
21.3.2 Clock Supply in SLEEP Mode 21-3
21.3.3 Clock Supply in DEBUG Mode 21-3
21.3.4 Frame Frequency 21-3
21.4 LCD Power Supply 21-4
21.4.1 Internal Generation Mode 21-4
21.4.2 External Voltage Application Mode 1 21-4
21.4.3 External Voltage Application Mode 2 21-5
21.4.4 External Voltage Application Mode 3 21-5
21.4.5 LCD Power Supply Circuit Settings 21-6
21.5 Operations 21-6
21.5.1 Initialization 21-6
21.5.2 Display On/Off 21-7
21.5.3 Inverted Display 21-7
21.5.4 Drive Duty Switching 21-7
21.5.5 Drive Waveforms 21-8
21.5.6 Partial Common Output Drive 21-17
21.5.7 n-Segment-Line Inverse AC Drive 21-17

21.6 Display Data RAM ... 21-17
21.6.1 Display Area Selection ... 21-17
21.6.2 Segment Pin Assignment ... 21-18
21.6.3 Common Pin Assignment ... 21-18
21.7 Interrupt ... 21-20
21.8 Control Registers ... 21-20
LCD8D Clock Control Register ... 21-20
LCD8D Control Register ... 21-21
LCD8D Timing Control Register 1 ... 21-21
LCD8D Timing Control Register 2 ... 21-22
LCD8D Power Control Register ... 21-22
LCD8D Display Control Register ... 21-24
LCD8D COM Pin Control Register 0 ... 21-24
LCD8D Interrupt Flag Register ... 21-25
LCD8D Interrupt Enable Register ... 21-25

22 R/F Converter (RFC) ... 22-1
22.1 Overview ... 22-1
22.2 Input/Output Pins and External Connections ... 22-2
22.2.1 List of Input/Output Pins ... 22-2
22.2.2 External Connections ... 22-2
22.3 Clock Settings ... 22-3
22.3.1 RFC Operating Clock ... 22-3
22.3.2 Clock Supply in SLEEP Mode ... 22-3
22.3.3 Clock Supply in DEBUG Mode ... 22-3
22.4 Operations ... 22-3
22.4.1 Initialization ... 22-3
22.4.2 Operating Modes ... 22-4
22.4.3 RFC Counters ... 22-4
22.4.4 Converting Operations and Control Procedure ... 22-5
22.4.5 CR Oscillation Frequency Monitoring Function ... 22-7
22.5 Interrupts ... 22-7
22.6 Control Registers ... 22-8
RFC Ch.*n* Clock Control Register ... 22-8
RFC Ch.*n* Control Register ... 22-8
RFC Ch.*n* Oscillation Trigger Register ... 22-9
RFC Ch.*n* Measurement Counter Low and High Registers ... 22-10
RFC Ch.*n* Time Base Counter Low and High Registers ... 22-10
RFC Ch.*n* Interrupt Flag Register ... 22-11
RFC Ch.*n* Interrupt Enable Register ... 22-11

23 Electrical Characteristics ... 23-1
23.1 Absolute Maximum Ratings ... 23-1
23.2 Recommended Operating Conditions ... 23-1
23.3 Current Consumption ... 23-2
23.4 System Reset Controller (SRC) Characteristics ... 23-4
23.5 Clock Generator (CLG) Characteristics ... 23-5
23.6 Flash Memory Characteristics ... 23-7
23.7 Input/Output Port (PPORT) Characteristics ... 23-8
23.8 Supply Voltage Detector (SVD4) Characteristics ... 23-9
23.9 UART (UART3) Characteristics ... 23-11
23.10 Synchronous Serial Interface (SPIA) Characteristics ... 23-12
23.11 I^{2}C (I2C) Characteristics ... 23-14
23.12 LCD Driver (LCD8D) Characteristics ... 23-14

23.13 R/F Converter (RFC) Characteristics 23-19
23.14 12-bit A/D Converter (ADC12A) Characteristics 23-21
23.15 Temperature Sensor/Reference Voltage Generator (TSRVR) Characteristics 23-22

24 Basic External Connection Diagram 24-1

25 Package 25-1

Appendix A List of Peripheral Circuit Control Registers AP-A-1

0x4000 0000	System Register (SYS)	AP-A-1
0x4000 0020	Power Generator (PWGA)	AP-A-1
0x4000 0040–0x4000 0050	Clock Generator (CLG)	AP-A-1
0x4000 0060–0x4000 0062	System Reset Controller (SRC)	AP-A-2
0x4000 00a0–0x4000 00a4	Watchdog Timer (WDT2)	AP-A-3
0x4000 00c0–0x4000 00d2	Real-time Clock (RTCA)	AP-A-3
0x4000 0100–0x4000 0106	Supply Voltage Detector (SVD4) Ch.0	AP-A-5
0x4000 0140–0x4000 014c	16-bit Timer (T16) Ch.0	AP-A-5
0x4000 01b0	Flash Controller (FLASHC)	AP-A-5
0x4000 0200–0x4000 02e2	I/O Ports (PPORT)	AP-A-6
0x4000 0300–0x4000 031e	Universal Port Multiplexer (UPMUX)	AP-A-10
0x4000 0380–0x4000 0394	UART (UART3) Ch.0	AP-A-12
0x4000 03a0–0x4000 03ac	16-bit Timer (T16) Ch.1	AP-A-13
0x4000 03b0–0x4000 03be	Synchronous Serial Interface (SPIA) Ch.0	AP-A-13
0x4000 03c0–0x4000 03d6	I^2C (I2C) Ch.0	AP-A-14
0x4000 0400–0x4000 042c	16-bit PWM Timer (T16B) Ch.0	AP-A-15
0x4000 0440–0x4000 046c	16-bit PWM Timer (T16B) Ch.1	AP-A-18
0x4000 0480–0x4000 048c	16-bit Timer (T16) Ch.3	AP-A-20
0x4000 04a0–0x4000 04ac	16-bit Timer (T16) Ch.4	AP-A-20
0x4000 04c0–0x4000 04cc	16-bit Timer (T16) Ch.5	AP-A-21
0x4000 0600–0x4000 0614	UART (UART3) Ch.1	AP-A-21
0x4000 0660–0x4000 066c	16-bit Timer (T16) Ch.6	AP-A-23
0x4000 0670–0x4000 067e	Synchronous Serial Interface (SPIA) Ch.1	AP-A-23
0x4000 0680–0x4000 068c	16-bit Timer (T16) Ch.2	AP-A-24
0x4000 06c0–0x4000 06d6	I^2C (I2C) Ch.1	AP-A-25
0x4000 0700–0x4000 070c	Sound Generator (SNDA)	AP-A-26
0x4000 0720–0x4000 0732	IR Remote Controller (REMC3)	AP-A-26
0x4000 0740–0x4000 076c	16-bit PWM Timer (T16B) Ch.2	AP-A-27
0x4000 0780–0x4000 078c	16-bit Timer (T16) Ch.7	AP-A-30
0x4000 07a0–0x4000 07bc	12-bit A/D Converter (ADC12A) Ch.0	AP-A-30
0x4000 07c0–0x4000 07c2	Temperature Sensor/ Reference Voltage Generator (TSRVR) Ch.0	AP-A-31
0x4000 0800–0x4000 0812	LCD Driver (LCD8D)	AP-A-32
0x4000 0840–0x4000 0850	R/F Converter (RFC) Ch.0	AP-A-33
0x4000 1000–0x4000 2014	DMA Controller (DMAC)	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

Revision History

1 Overview

The S1C31W65 is a 32-bit MCU with an Arm® Cortex®-M0+ processor included that features low-power operation. It incorporates a lot of serial interface circuits and is suitable for various kinds of battery-driven controller applications.

1.1 Features

Table 1.1.1 Features

Model		S1C31W65
CPU		
CPU		Arm® 32-bit RISC processor Cortex®-M0+
Other		Serial-wire debug ports (SW-DP) and a micro trace buffer (MTB) included
Embedded Flash memory		
Capacity		128K bytes (for both instructions and data)
Erase/program count		1,000 times (min.) ∗ When being programmed by the dedicated flash loader
Other		On-board programming function
		Flash programming voltage can be generated internally.
Embedded RAMs		
General-purpose RAM		16K bytes (shared with MTB)
Display RAM		112 bytes
DMA Controller (DMAC)		
Number of channels		4 channels
Data transfer path		Memory to memory, memory to peripheral, and peripheral to memory
Transfer mode		Basic, ping-pong, scatter-gather
DMA trigger source		UART3, SPIA, I2C, T16B, SNDA, ADC12A, and software
Clock generator (CLG)		
System clock source		4 sources (IOSC/OSC1/OSC3/EXOSC)
System clock frequency (operating frequency)		V_{D1} voltage mode = mode0: 33 MHz (max.) V_{D1} voltage mode = mode1: 2.16 MHz (max.)
IOSC oscillator circuit (boot clock source)		V_{D1} voltage mode = mode0: 32/24/16/12/8/2/1 MHz (typ.) selectable embedded oscillator V_{D1} voltage mode = mode1: 2/1 MHz (typ.) selectable embedded oscillator
		2 µs (max.) starting time (time from cancelation of SLEEP state to vector table read by the CPU when the system clock = 32 MHz)
OSC1 oscillator circuit		32.768 kHz (typ.) crystal oscillator
		32 kHz (typ.) embedded oscillator
		Oscillation stop detection circuit included
OSC3 oscillator circuit		33 MHz (max.) crystal/ceramic oscillator
		32/24/16/12/8 MHz (typ.) selectable embedded oscillator
EXOSC clock input		33 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 ports	63 bits (max.)
	Output port	1 bit
	Other	Pins are shared with the peripheral I/O.
Input interrupt	Number of interrupt ports	56 bits (max.)
	Interrupt type	Rising edge interrupts and falling edge interrupts can be enabled individually.
Number of ports that support universal port multiplexer (UPMUX)		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)		8 channels
		Generates the SPIA master clock and the ADC12A trigger signal.
16-bit PWM timer (T16B)		3 channels
		Event counter/capture function
		PWM waveform generation function
		Number of PWM output or capture input ports: 4 ports/channel

Supply voltage detector (SVD4)		
Number of channels	1 channel	
Detection voltage	V_{DD} or an external voltage (2 external detection ports are available.)	
Detection level	V_{DD}: 32 levels (1.7 to 5.0 V)/external voltage: 32 levels (1.7 to 5.0 V)	
Other	Intermittent operation mode	
	Generates an interrupt or reset according to the detection level evaluation.	
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^2C (I2C) *1	2 channels	
	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 (by the hardware) for an application example.	
	Output inversion function	
12-bit A/D converter (ADC12A)		
Conversion method	Successive approximation type	
Resolution	12 bits	
Number of conversion channels	1 channel	
Number of analog signal inputs	8 ports/channel (The temperature sensor output is connected to a port.)	
Temperature sensor/reference voltage generator (TSRVR)		
Temperature sensor circuit	Sensor output can be measured using ADC12A.	
Reference voltage generator	Reference voltage for ADC12A is selectable from 2.0 V, 2.5 V, V_{DD}, and external input.	
LCD driver (LCD8D)		
LCD output	52 SEG × 5–8 COM(max.), 56 SEG × 1–4 COM(max.)	
LCD contrast	32 levels	
LCD drive waveform	2 types (Waveform A, Waveform B) selectable	
Other	1/3 or 1/2 bias power supply with voltage booster included, external voltage can be applied.	
R/F converter (RFC)		
Conversion method	CR oscillation type with 24-bit counters	
Number of conversion channels	1 channel (Up to two sensors can be connected.)	
Supported sensors	DC-bias resistive sensors	
Reset		
#RESET pin	Reset when the reset pin is set to low.	Can be enabled/disabled using a register.
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.	
Supply voltage detector reset	Reset when the supply voltage detector detects the set voltage level.	
Interrupt		
Non-maskable interrupt	6 systems (Reset, NMI, HardFault, SVCall, PendSV, SysTic)	
Programmable interrupt	External interrupt: 1 system	
	Internal interrupt: 26 systems	
Power supply voltage		
V_{DD} operating voltage	1.8 to 5.5 V	
V_{DD} operating voltage for Flash programming	2.2 to 5.5 V	
V_{DD} operating voltage when LCD driver is used	1.8 to 5.5 V	
Operating temperature		
Operating temperature range	-40 to 105°C	
Flash programming temperature range	-40 to 85°C	
Current consumption (Typ. value)		
SLEEP mode *2	0.3 µA IOSC = OFF, OSC1 = OFF, OSC3 = OFF	
	0.8 µA IOSC = OFF, OSC1 = 32.768 kHz (crystal oscillator), OSC3 = OFF, RTCA = ON	
HALT mode *3	1.5 µA OSC1 = 32.768 kHz (crystal oscillator)	
	4.0 µA OSC1 = 32.768 kHz (crystal oscillator), LCD = ON (no panel load)	

Current consumption (Typ. value)		
RUN mode	195 µA/MHz	V_{D1} voltage mode = mode0, CPU = IOSC (16 MHz)
	130 µA/MHz	V_{D1} voltage mode = mode1, CPU = IOSC (2 MHz)
Shipping form		
Package *4	TQFP15-100PIN (P-TQFP100-1414-0.50, 14 × 14 mm, t = 1.2 mm, 0.5 mm pitch)	

*1 The input filter in I2C (SDA and SCL inputs) does not comply with the standard for removing noise spikes less than 50 ns.
*2 SLEEP mode refers to deep sleep mode in the Cortex®-M0+ processor. The RAM retains data even in SLEEP mode.
*3 HALT mode refers to sleep mode in the Cortex®-M0+ processor.
*4 Shown in parentheses is a JEITA package name.

1.2 Block Diagram

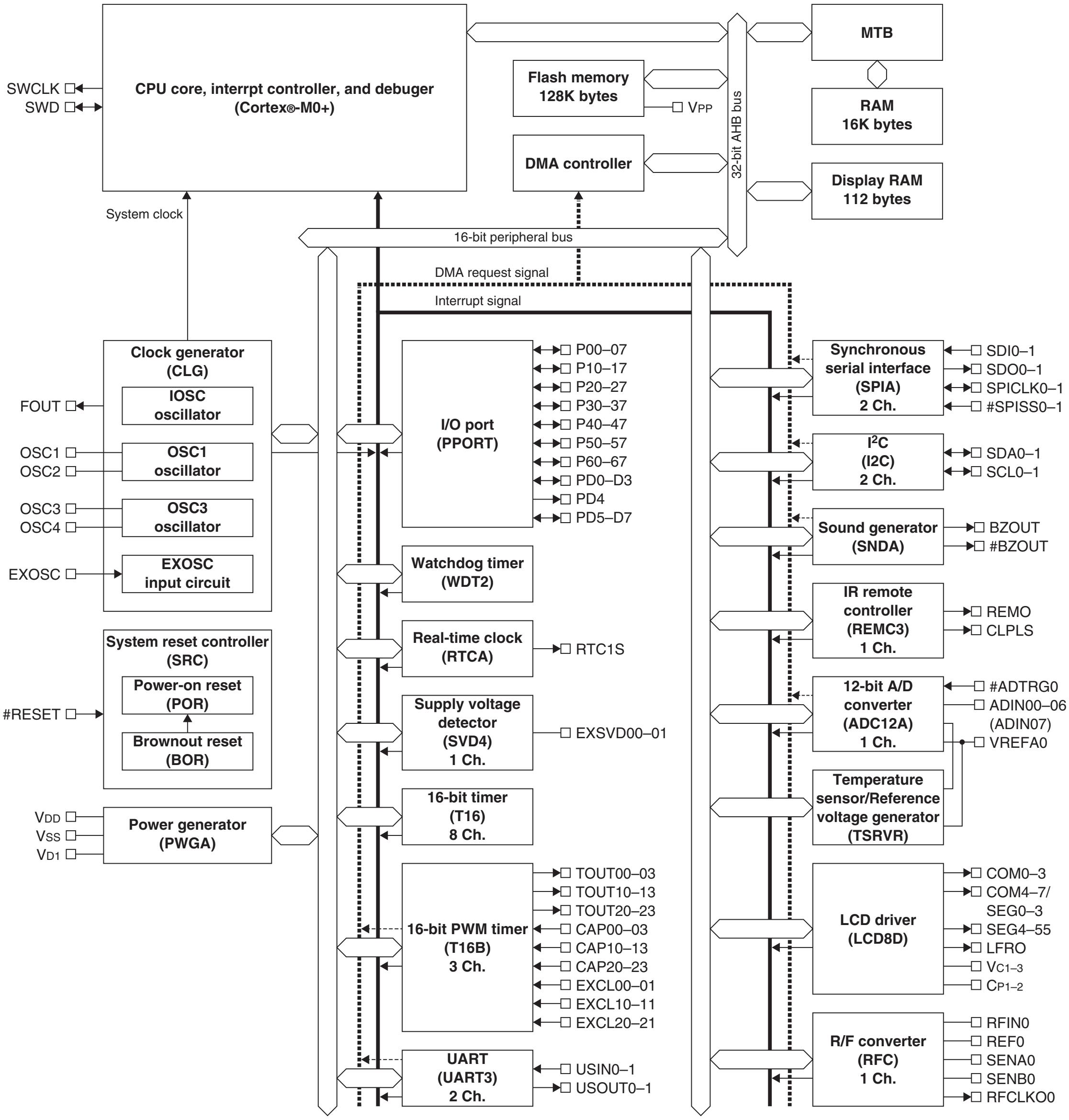


Figure 1.2.1 S1C31W65 Block Diagram

1.3 Pins

1.3.1 Pin Configuration Diagram

TQFP15-100PIN

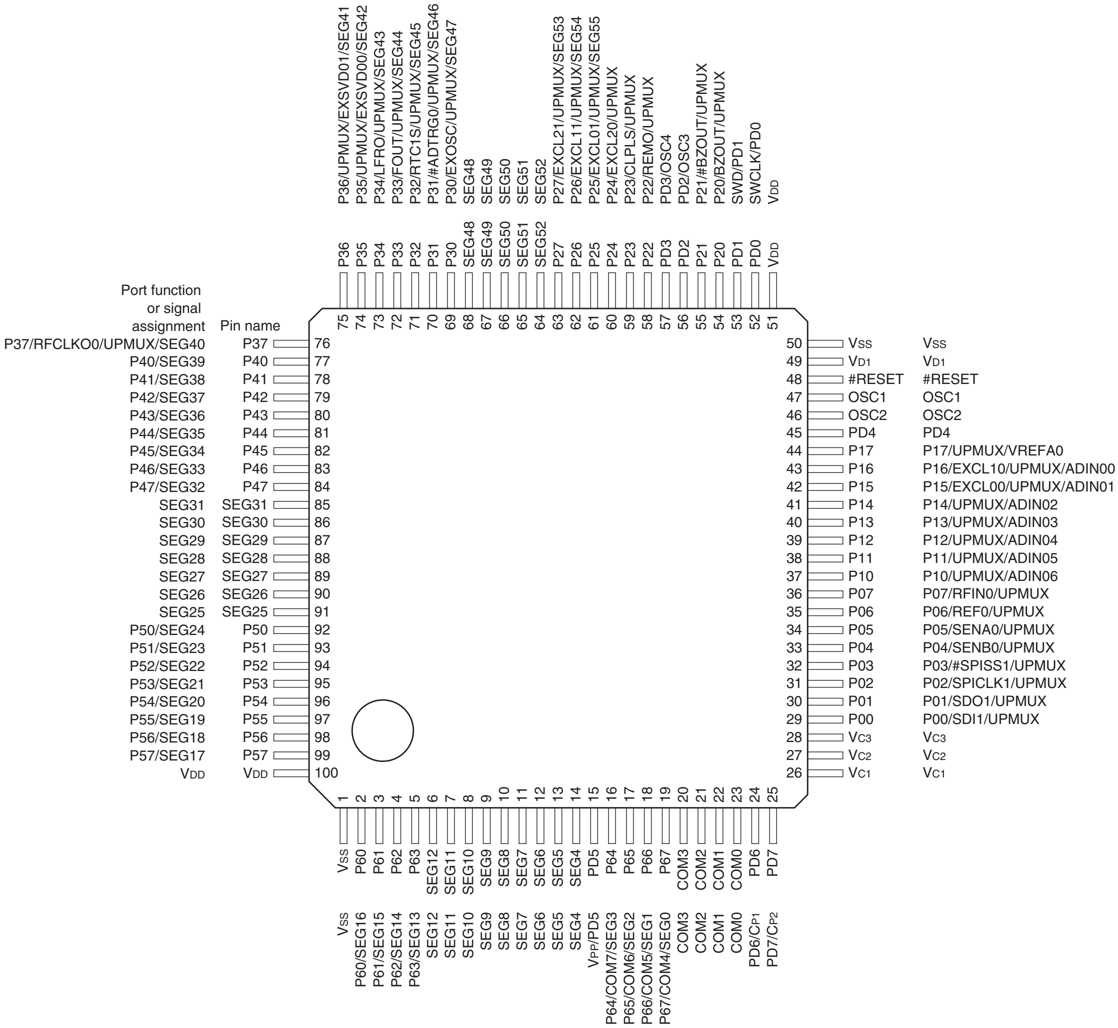


Figure 1.3.1.1 S1C31W65 Pin Configuration Diagram (TQFP15-100PIN)

1.3.2 Pin Descriptions

Symbol meanings

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

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

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

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

Table 1.3.2.1 Pin Description

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function
V_{DD}	V_{DD}	P	–	–	Power supply (+)
V_{SS}	V_{SS}	P	–	–	GND
V_{D1}	V_{D1}	A	–	–	V_{D1} regulator output
V_{C1-3}	V_{C1-3}	P	–	–	LCD panel driver power supply
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
	SDI1	I			Synchronous serial interface Ch.1 data input
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P01	P01	I/O	Hi-Z	✓	I/O port
	SDO1	O			Synchronous serial interface Ch.1 data output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P02	P02	I/O	Hi-Z	✓	I/O port
	SPICLK1	I/O			Synchronous serial interface Ch.1 clock input/output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P03	P03	I/O	Hi-Z	✓	I/O port
	#SPISS1	I			Synchronous serial interface Ch.1 slave-select input
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P04	P04	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)
P05	P05	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)
P06	P06	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)
P07	P07	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)
P10	P10	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
P11	P11	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
P12	P12	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

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function
P13	P13	I/O	Hi-Z	–	I/O port
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	ADIN03	A			12-bit A/D converter Ch.0 analog signal input 3
P14	P14	I/O	Hi-Z	–	I/O port
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	ADIN02	A			12-bit A/D converter Ch.0 analog signal input 2
P15	P15	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)
	ADIN01	A			12-bit A/D converter Ch.0 analog signal input 1
P16	P16	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)
	ADIN00	A			12-bit A/D converter Ch.0 analog signal input 0
P17	P17	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
P20	P20	I/O	Hi-Z	✓	I/O port
	BZOUT	O			Sound generator output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P21	P21	I/O	Hi-Z	✓	I/O port
	#BZOUT	O			Sound generator inverted output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P22	P22	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)
P23	P23	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)
P24	P24	I/O	Hi-Z	✓	I/O port
	EXCL20	I			16-bit PWM timer Ch.2 event counter input 0
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
P25	P25	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)
	SEG55	A			LCD segment output
P26	P26	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)
	SEG54	A			LCD segment output
P27	P27	I/O	Hi-Z	✓	I/O port
	EXCL21	I			16-bit PWM timer Ch.2 event counter input 1
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG53	A			LCD segment output
P30	P30	I/O	Hi-Z	✓	I/O port
	EXOSC	I			Clock generator external clock input
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG47	A			LCD segment output
P31	P31	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)
	SEG46	A			LCD segment output
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)
	SEG45	A			LCD segment output
P33	P33	I/O	Hi-Z	✓	I/O port
	FOUT	O			Clock external output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG44	A			12-bit A/D converter Ch.0 analog signal input 3

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function
P34	P34	I/O	Hi-Z	✓	I/O port
	LFRO	O			LCD frame signal monitor output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG43	A			LCD segment output
P35	P35	I/O	Hi-Z	✓	I/O port
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	EXSVD00	A			Supply voltage detector Ch.0 external voltage detection input 0
	SEG42	A			LCD segment output
P36	P36	I/O	Hi-Z	✓	I/O port
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	EXSVD01	A			Supply voltage detector Ch.0 external voltage detection input 1
	SEG41	A			LCD segment output
P37	P37	I/O	Hi-Z	✓	I/O port
	RFCLKO0	O			R/F converter Ch.0 clock monitor output
	UPMUX	I/O			User-selected I/O (universal port multiplexer)
	SEG40	A			LCD segment output
P40	P40	I/O	Hi-Z	✓	I/O port
	SEG39	A			LCD segment output
P41	P41	I/O	Hi-Z	✓	I/O port
	SEG38	A			LCD segment output
P42	P42	I/O	Hi-Z	✓	I/O port
	SEG37	A			LCD segment output
P43	P43	I/O	Hi-Z	✓	I/O port
	SEG36	A			LCD segment output
P44	P44	I/O	Hi-Z	✓	I/O port
	SEG35	A			LCD segment output
P45	P45	I/O	Hi-Z	✓	I/O port
	SEG34	A			LCD segment output
P46	P46	I/O	Hi-Z	✓	I/O port
	SEG33	A			LCD segment output
P47	P47	I/O	Hi-Z	✓	I/O port
	SEG32	A			LCD segment output
P50	P50	I/O	Hi-Z	✓	I/O port
	SEG24	A			LCD segment output
P51	P51	I/O	Hi-Z	✓	I/O port
	SEG23	A			LCD segment output
P52	P52	I/O	Hi-Z	✓	I/O port
	SEG22	A			LCD segment output
P53	P53	I/O	Hi-Z	✓	I/O port
	SEG21	A			LCD segment output
P54	P54	I/O	Hi-Z	✓	I/O port
	SEG20	A			LCD segment output
P55	P55	I/O	Hi-Z	✓	I/O port
	SEG19	A			LCD segment output
P56	P56	I/O	Hi-Z	✓	I/O port
	SEG18	A			LCD segment output
P57	P57	I/O	Hi-Z	✓	I/O port
	SEG17	A			LCD segment output
P60	P60	I/O	Hi-Z	✓	I/O port
	SEG16	A			LCD segment output
P61	P61	I/O	Hi-Z	✓	I/O port
	SEG15	A			LCD segment output
P62	P62	I/O	Hi-Z	✓	I/O port
	SEG14	A			LCD segment output
P63	P63	I/O	Hi-Z	✓	I/O port
	SEG13	A			LCD segment output
P64	P64	I/O	Hi-Z	✓	I/O port
	COM7	A			LCD common output
	SEG3	A			LCD segment output
P65	P65	I/O	Hi-Z	✓	I/O port
	COM6	A			LCD common output
	SEG2	A			LCD segment output

Pin name	Assigned signal	I/O	Initial state	Tolerant fail-safe structure	Function
P66	P66	I/O	Hi-Z	✓	I/O port
	COM5	A			LCD common output
	SEG1	A			LCD segment output
P67	P67	I/O	Hi-Z	✓	I/O port
	COM4	A			LCD common output
	SEG0	A			LCD segment output
PD0	SWCLK	I	I (Pull-up)	✓	Serial-wire debugger clock input
	PD0	I/O			I/O port
PD1	SWD	I/O	I (Pull-up)	✓	Serial-wire debugger data input/output
	PD1	I/O			I/O port
PD2	PD2	I/O	Hi-Z	–	I/O port
	OSC3	A			OSC3 oscillator circuit input
PD3	PD3	I/O	Hi-Z	–	I/O port
	OSC4	A			OSC3 oscillator circuit output
PD4	PD4	O	O (L)	–	Output port
PD5	V_{PP}	P	Hi-Z	✓	Power supply for Flash programming
	PD5	I/O			I/O port
PD6	PD6	I/O	Hi-Z	–	I/O port (power supply voltage: V_{C3})
	C_{P1}	A			LCD voltage boost capacitor connect pin
PD7	PD7	I/O	Hi-Z	–	I/O port (power supply voltage: V_{C3})
	C_{P2}	A			LCD voltage boost capacitor connect pin
COM0–3	COM0–3	A	Hi-Z	–	LCD common outputs
SEG4–12, 25–31, 48–52	SEG4–12, 25–31, 48–52	A	Hi-Z	–	LCD segment outputs

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

Universal port multiplexer (UPMUX)

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

Table 1.3.3.2 Peripheral Circuit Input/output Function Selectable by UPMUX

Peripheral circuit	Signal to be assigned	I/O	Channel number *n*	Function
Synchronous serial interface (SPIA)	SDI*n*	I	*n* = 0	SPIA Ch.*n* data input
	SDO*n*	O		SPIA Ch.*n* data output
	SPICLK*n*	I/O		SPIA Ch.*n* clock input/output
	#SPISS*n*	I		SPIA Ch.*n* slave-select input
I^2C (I2C)	SCL*n*	I/O	*n* = 0, 1	I2C Ch.*n* clock input/output
	SDA*n*	I/O		I2C Ch.*n* data input/output
UART (UART3)	USIN*n*	I	*n* = 0, 1	UART3 Ch.*n* data input
	USOUT*n*	O		UART3 Ch.*n* data output
16-bit PWM timer (T16B)	TOUT*n*0/CAP*n*0	I/O	*n* = 0, 1, 2	T16B Ch.*n* PWM output/capture input 0
	TOUT*n*1/CAP*n*1	I/O		T16B Ch.*n* PWM output/capture input 1
	TOUT*n*2/CAP*n*2	I/O		T16B Ch.*n* PWM output/capture input 2
	TOUT*n*3/CAP*n*3	I/O		T16B Ch.*n* PWM output/capture input 3

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 (PWGA)

2.1.1 Overview

PWGA is the power generator that controls the internal power supply system to drive this IC with stability and low power. The main features of PWGA 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.
  - The V_{D1} regulator supports two voltage modes, mode0 and mode1, and setting the V_{D1} regulator into mode1 during low-speed operation helps achieve low-power operations.

Figure 2.1.1.1 shows the PWGA configuration.

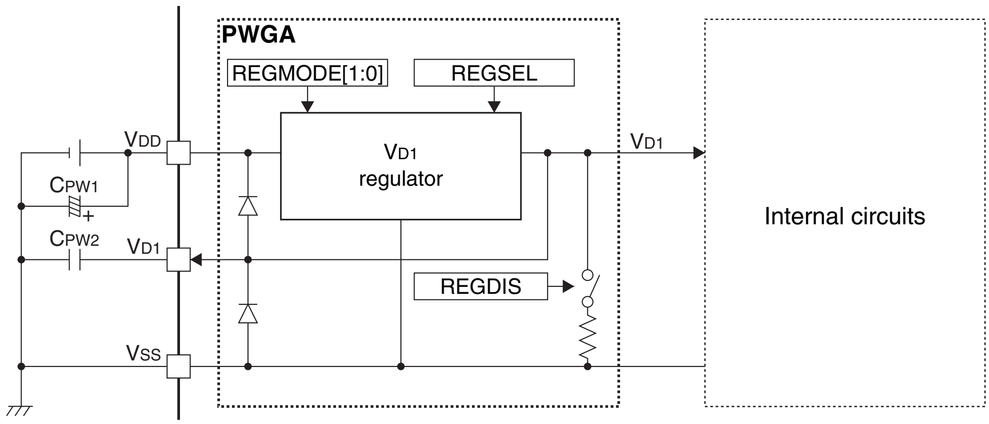


Figure 2.1.1.1 PWGA Configuration

2.1.2 Pins

Table 2.1.2.1 lists the PWGA pins.

Table 2.1.2.1 List of PWGA Pins

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

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.

2.1.4 V_{D1} Regulator Voltage Mode

The V_{D1} regulator supports two voltage modes, mode0 and mode1.
When the IC runs with a low-speed clock, setting the V_{D1} regulator into mode1 reduces power consumption.

When the voltage mode is switched, the system clock source automatically stops operating and it resumes operating after the voltage has stabilized. Table 2.1.4.1 shows the stop period of the system clock.

Table 2.1.4.1 System Clock Stop Period After Switching Voltage Mode

System clock	Stop period
IOSC	2,048 cycles
OSC1	Number of cycles set using the CLGOSC1.OSC1WT[1:0] bits

Procedure to switch from mode0 to mode1

1. Set the MODEN bits of the peripheral circuits to 0. (Stop using peripheral circuits)
2. Write 0x0096 to the SYSPROT.PROT[15:0] bits. (Remove system protection)
3. Switch the system clock to a low-speed clock (OSC1, IOSC 2 MHz or 1 MHz).
4. Stop OSC3 and EXOSC.
5. Configure the following PWGACTL register bits.
   - Set the PWGACTL.REGSEL bit to 0. (Switch to mode1)
   - Set the PWGACTL.REGDIS bit to 1. (Discharge)
   - Set the PWGACTL.REGMODE[1:0] bits to 0x2. (Set to normal mode)
6. Configure the following PWGACTL register bits after the system clock supply has resumed.
   - Set the PWGACTL.REGDIS bit to 0. (Stop discharging)
   - Set the PWGACTL.REGMODE[1:0] bits to 0x0. (Set to automatic mode)
7. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

Procedure to switch from mode1 to mode0

1. Set the MODEN bits of the peripheral circuits to 0. (Stop using peripheral circuits)
2. Write 0x0096 to the SYSPROT.PROT[15:0] bits. (Remove system protection)
3. Configure the following PWGACTL register bits.
   - Set the PWGACTL.REGSEL bit to 1. (Switch to mode0)
   - Set the PWGACTL.REGMODE[1:0] bits to 0x2. (Set to normal mode)
4. Set the PWGACTL.REGMODE[1:0] bits to 0x0 after the system clock supply has resumed. (Set to automatic mode)
5. Switch the system clock to a high-speed clock.
6. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

Note: After the voltage mode has been switched, correct the RTC, as the RTC operating clock is also stopped for the period set using the CLGOSC1.OSC1WT[1:0] bits.

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*
  - Reset request from the CPU
  - Key-entry reset*
  - Watchdog timer reset*
  - Supply voltage detector reset*
  - Peripheral circuit software reset (supports some peripheral circuits only)

  * SRC allows software to check the generated reset request after a reset has occurred.
- #RESET pin control function
  - External reset input can be enabled/disabled.
  - The pull-up resistor can be enabled/disabled.
- 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.

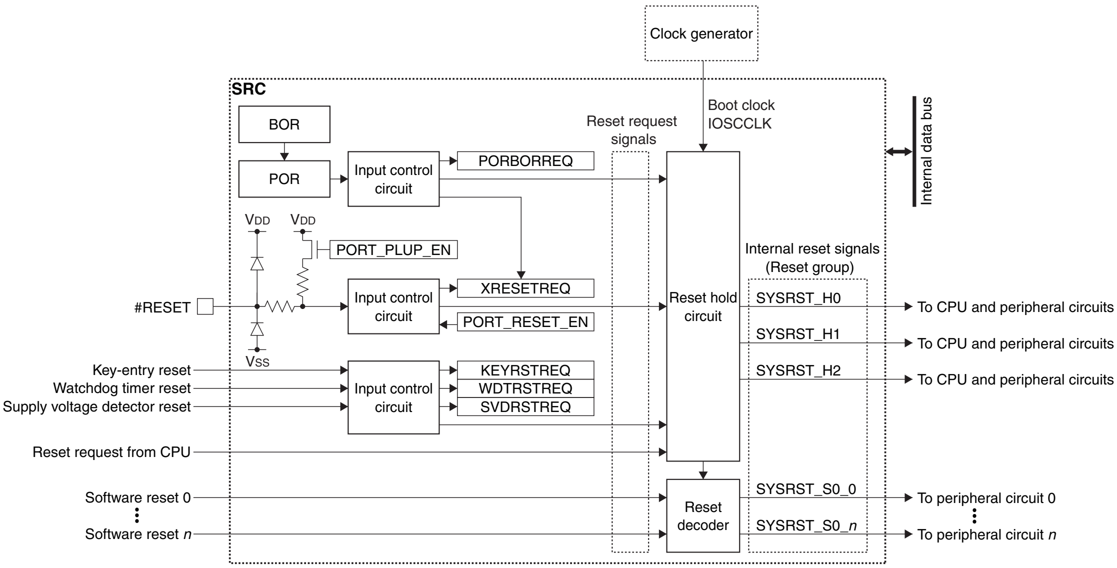


Figure 2.2.1.1 SRC Configuration

2.2.2 Input Pin

Table 2.2.2.1 shows the SRC input 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. The #RESET pin includes a pull-up resistor that can be enabled/disabled using the SRCRESETPCTL.PORT_PLUP_EN bit. 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}.

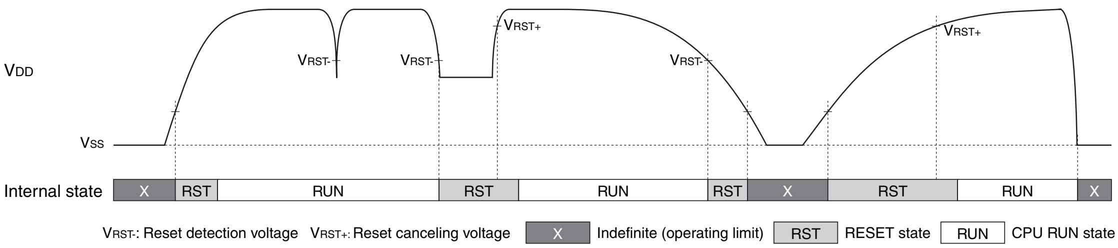


Figure 2.2.3.1 Example of Internal Reset by POR and BOR

For the POR and BOR electrical specifications, refer to "POR/BOR characteristics" in the "Electrical Characteristics" chapter.

Reset request from the CPU

The CPU issues a reset request by writing 1 to the AIRCR.SYSRESETREQ bit in the Cortex®-M0+ Application Interrupt and Reset Control Register. For more information, refer to the "ARM®v6-M Architecture Reference Manual."

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 Reset Request Flag

Table 2.2.4.1 Reset Request Flag

Reset source	Reset request flag	Set	Clear
POR/BOR	SRCRESETREQ.PORBORREQ	When a POR/BOR reset request is issued	Writing 1
#RESET pin	SRCRESETREQ.XRESETREQ	When a low level pulse is input to the #RESET pin or a POR/BOR reset request is issued	Writing 1
Watchdog timer reset	SRCRESETREQ.WDTRSTREQ	When the watchdog timer issues a reset request due to a counter overflow	Writing 1
Supply voltage detector reset	SRCRESETREQ.SVDRSTREQ	When the supply voltage detector issues a reset request due to a drop in the power supply voltage	Writing 1
Key-entry reset	SRCRESETREQ.KEYRSTREQ	When the I/O port issues a reset request due to a low level input to the specified ports	Writing 1

When a reset request is issued from a reset source, the corresponding reset request flag (SRCRESETREQ.*xxxx*REQ bit) is set to 1 (except for reset requests from the CPU and software reset requests from peripheral circuits). This makes it possible to determine the reset source, which issued a reset, after the MCU has rebooted by reading the reset request flag that has been set. The SRCRESETREQ.*xxxx*REQ bit, which has been set, is cleared by writing 1 via software.
Note that the reset request flag (SRCRESETREQ.XRESETREQ bit) of the #RESET pin is also set by a POR/BOR reset request (SRCRESETREQ.PORBORREQ bit = 1).When a reset has been issued by the #RESET pin, the SRCRESETREQ.XRESETREQ bit is set to 1 and the SRCRESETREQ.PORBORREQ bit is set to ０.

2.2.5 #RESET Pin Input Control

The #RESET pin functions as the external reset input pin by setting the SRCRESETPCTL.PORT_RESET_EN bit to 1. If the SRCRESETPCTL.PORT_RESET_EN bit = 0, the #RESET pin input is fixed at a high level to disable #RESET inputs.
The #RESET pin has a built-in pull-up resistor and it can be disabled by setting the SRCRESETPCTL.PORT_PLUP_EN bit to 0.

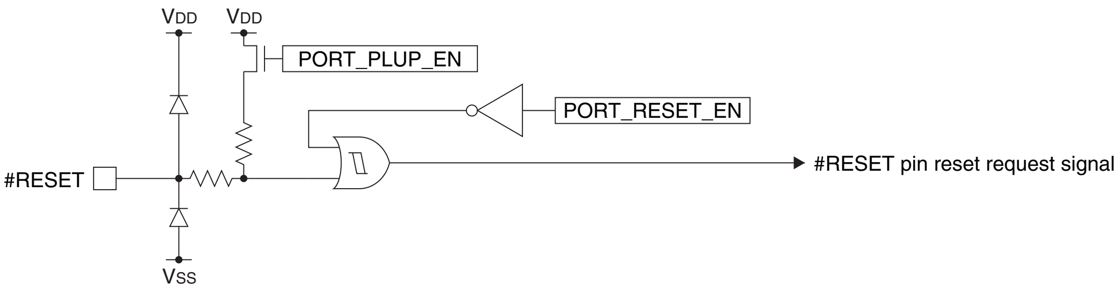


Figure 2.2.5.1 #RESET Pin Input Control Circuit

2.2.6 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 2.2.6.1 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 2.2.6.1 List of Reset Groups

Reset group	Reset source	Reset cancelation timing
H0	#RESET pin POR and BOR Reset request from the CPU 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 Reset request from the CPU	
H2	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.
- The 8 MHz clock 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.

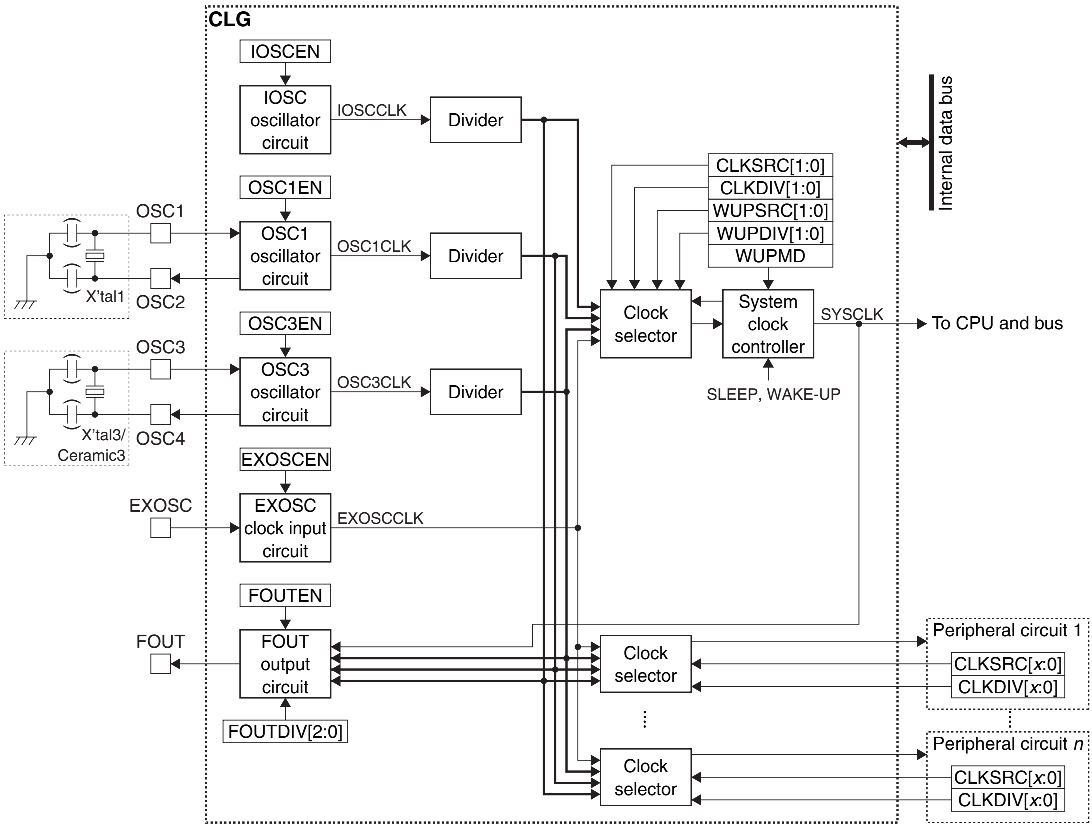


Figure 2.3.1.1 CLG Configuration

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.

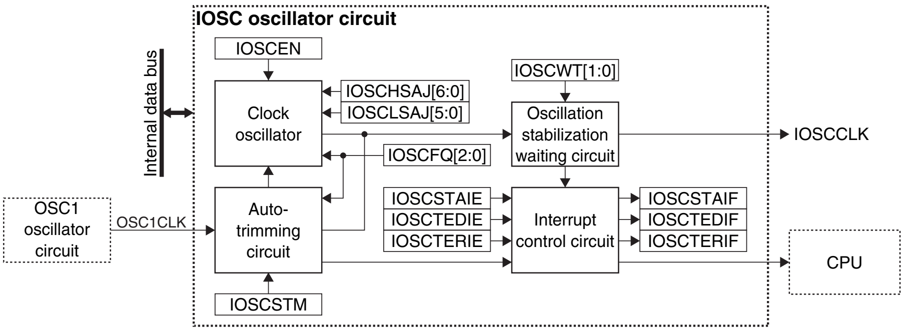


Figure 2.3.3.1 IOSC Oscillator Circuit Configuration

The IOSC oscillator circuit output clock IOSCCLK is used as SYSCLK at booting. The IOSCCLK frequency can be selected using the CLGIOSC.IOSCFQ[2:0] bits. The IOSC 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 and the oscillation characteristics, refer to "IOSC oscillation auto-trimming function" in this chapter and "IOSC oscillator circuit characteristics" in the "Electrical Characteristics" chapter, respectively.

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.

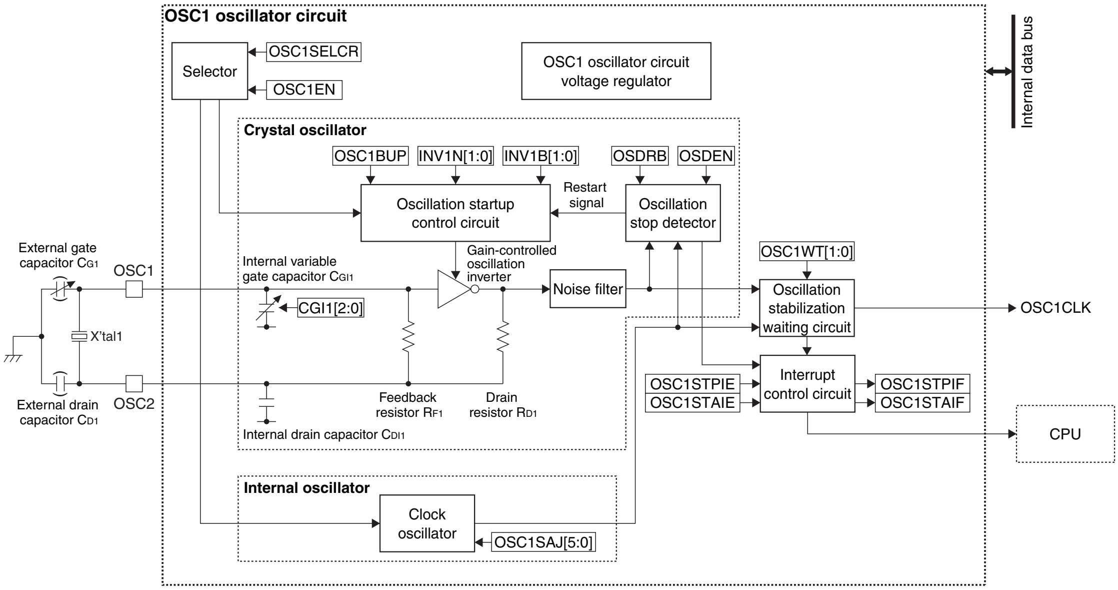


Figure 2.3.3.2 OSC1 Oscillator Circuit Configuration

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, set the OSC1 pin level to V_{SS} and leave the OSC2 pin open.

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 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.

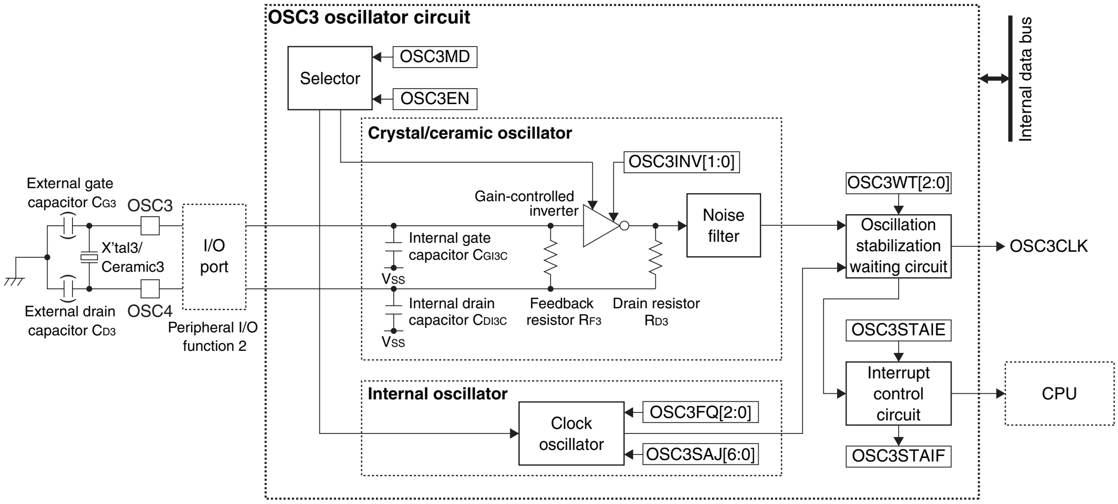


Figure 2.3.3.3 OSC3 Oscillator Circuit Configuration

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.

Note: Depending on the circuit board or the crystal resonator type used, an external gate capacitor C_{G3} and a drain capacitor C_{D3} may be required.

Internal oscillator

This oscillator circuit operates without any external parts. The OSC3CLK frequency can be selected using the CLGOSC3.OSC3FQ[2:0] bits.

For the recommended parts and the oscillation characteristics, refer to the "Basic External Connection Diagram" chapter and 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.

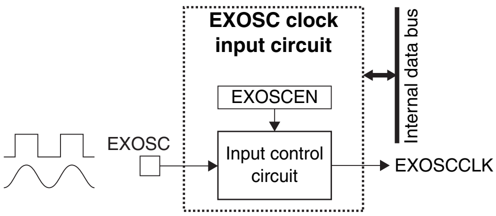


Figure 2.3.3.4 EXOSC Clock Input Circuit

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.

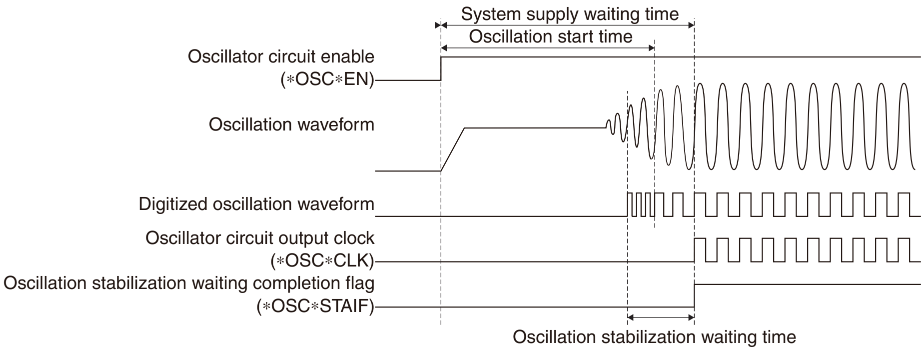


Figure 2.3.4.1 Oscillation Start Time and Oscillation Stabilization Waiting Time

The oscillation stabilization waiting times for the IOSC, OSC1, and OSC3 oscillator circuits can be set using the CLGIOSC.IOSCWT[1:0] bits, 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 when the IOSC frequency is 2 MHz or lower. When the frequency is 8 MHz or higher, set to 8 IOSCCLK clocks or more.

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.

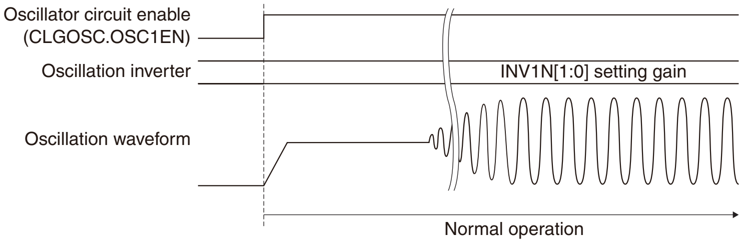
The oscillation stabilization waiting time for the OSC3 oscillator circuit should be set to 1,024 OSC3CLK clocks or more when crystal/ceramic oscillator is selected, or 16 OSC3CLK clocks or more when internal oscillator is selected.

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 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)

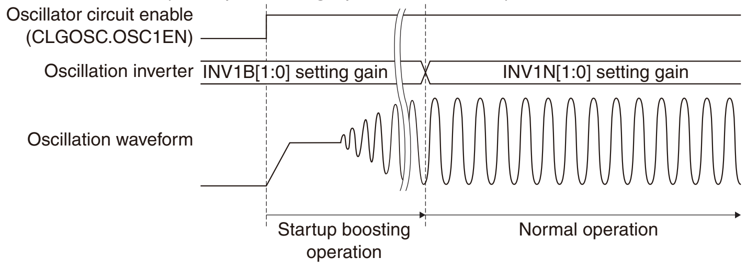


Figure 2.3.4.2 Operation Example when the Oscillation Startup Control Circuit is Used

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 0x0096 to the SYSPROT.PROT[15:0] bits. (Remove system protection)
4. Configure the following CLGIOSC register bits:
   - CLGIOSC.IOSCWT[1:0] bit (Set oscillation stabilization waiting time)
   - CLGIOSC.IOSCFQ[2:0] bit (Select frequency)
5. Set the CLGTRIM1.IOSCLSAJ[5:0] bits (f_{IOSC} = 2/1 MHz) or CLGTRIM1.IOSCHSAJ[6:0] bits (f_{IOSC} = 32/24/16/12/8 MHz) as necessary. (Finely adjust oscillation frequency)
6. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)
7. Write 1 to the CLGOSC.IOSCEN bit. (Start oscillation)
8. IOSCCLK can be used if the CLGINTF.IOSCSTAIF bit = 1 after an interrupt occurs.

The setting values of the CLGTRIM1.IOSCLSAJ[5:0] and CLGTRIM1.IOSCHSAJ[6:0] bits should be determined after performing evaluation using the populated circuit board.

Note: Make sure the CLGOSC.IOSCEN bit is set to 0 (while the IOSC oscillation is halted) when setting the CLGTRIM1.IOSCLSAJ[5:0] or CLGTRIM1.IOSCHSAJ[6:0] bits.

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 SYSPROT.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. When using the internal oscillator, set the CLGTRIM2.OSC1SAJ[5:0] bits as necessary. (Finely adjust oscillation frequency)
6. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)
7. Write 1 to the CLGOSC.OSC1EN bit. (Start oscillation)
8. 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], CLGOSC1.INV1B[1:0], and CLGTRIM2.OSC1SAJ[5:0] 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 OSC1 oscillation is halted) when setting the CLGTRIM2.OSC1SAJ[5:0] bits.

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 SYSPROT.PROT[15:0] bits. (Remove system protection)
4. Configure the following CLGOSC1 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[2:0] bits (Select frequency)
5. When using the internal oscillator, set the CLGTRIM2.OSC3SAJ[6:0] bits as necessary. (Finely adjust oscillation frequency)
6. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)
7. When using the crystal/ceramic oscillator, assign the OSC3 oscillator input/output functions to the ports. (Refer to the “I/O Ports” chapter.)
8. Write 1 to the CLGOSC.OSC3EN bit. (Start oscillation)
9. OSC3CLK can be used if the CLGINTF.OSC3STAIF bit = 1 after an interrupt occurs.

The setting values of the CLGOSC3.OSC3INV[1:0], CLGOSC3.OSC3WT[2:0], and CLGTRIM2.OSC3SAJ[6:0] 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 setting the CLGTRIM2.OSC3SAJ[6:0] bits.

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 SYSPROT.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

Whether the clock sources being operated are stopped or not when the CPU enters SLEEP mode (deep sleep mode) 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.

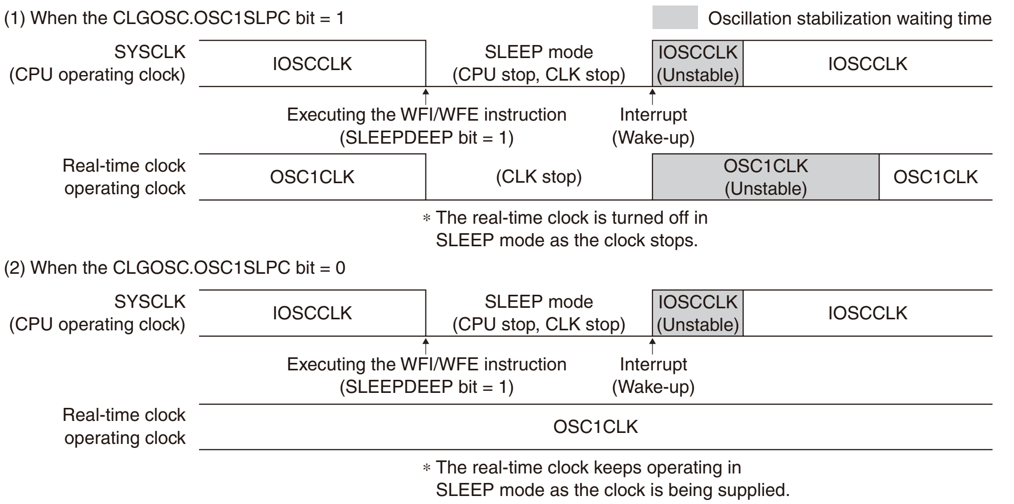


Figure 2.3.4.3 Clock Control Example in SLEEP Mode

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.

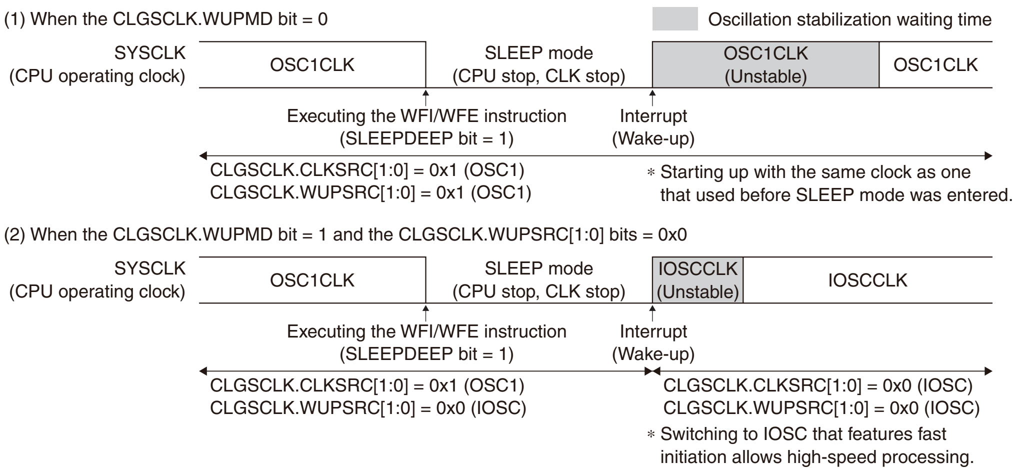


Figure 2.3.4.4 Clock Control Example at SLEEP Cancelation

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)

IOSC oscillation auto-trimming function

The auto-trimming function adjusts the IOSCCLK clock frequency selected using the CLGIOSC.IOSCFQ[2:0] bits by trimming the clock with reference to the high precision OSC1CLK clock generated by the OSC1 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 IOSC oscillation, check if the stabilized clock is supplied (CLGINTF.IOSCSTAIF bit = 1).
3. Write 0x0096 to the SYSPROT.PROT[15:0] bits. (Remove system protection)
4. If 1 or 2 MHz IOSC has been selected as the SYSCLK clock source, set the CLGSCLK.CLKSRC[1:0] bits to a value other than 0x0 (IOSC).
5. Configure the following CLGINTF register bits:
   - Write 1 to the CLGINTF.IOSCTEDIF bit. (Clear interrupt flag)
   - Write 1 to the CLGINTF.IOSCTERIF bit. (Clear interrupt flag)
6. Configure the following CLGINTF register bits:
   - Set the CLGINTE.IOSCTEDIE bit to 1. (Enable interrupt)
   - Set the CLGINTE.IOSCTERIE bit to 1. (Enable interrupt)
7. Write 1 to the CLGIOSC.IOSCSTM bit. (Enable IOSC oscillation auto-trimming)
8. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)
9. The trimmed IOSCCLK can be used if the CLGINTF.IOSCTEDIF bit = 1 after an interrupt occurs. If the CLGINTF.IOSCTERIF bit = 1, an error has occurred during the auto-trimming operation (the clock has not been adjusted).

After the trimming operation has completed, the CLGIOSC.IOSCSTM bit automatically reverts to 0. Although the trimming time depends on the temperature, an average of several 10 ms is required. When IOSCCLK 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 CLGINTE.OSC1STPIE bit. (Enable interrupt)
4. Write 0x0096 to the SYSPROT.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 SYSPROT.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.

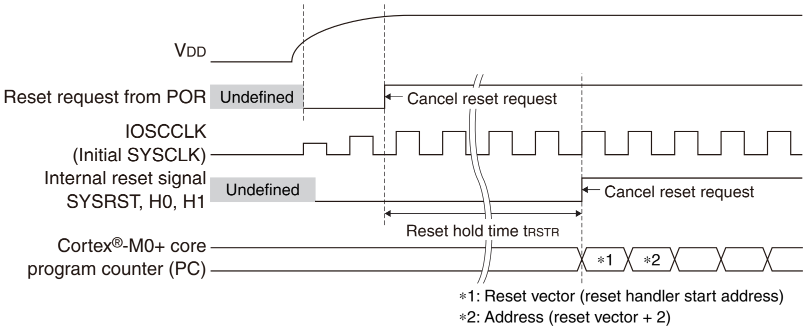


Figure 2.4.1.1 Initial Boot Sequence

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

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

2.4.2 Transition between Operating Modes

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

RUN mode

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

HALT mode

When the Cortex®-M0+ core executes the WFI or WFE instruction with the SLEEPDEEP bit of the Cortex®-M0+ System Control Register set to 0, it suspends program execution and stops operating. This state is referred to HALT mode in this IC. 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 Cortex®-M0+ core executes the WFI or WFE instruction with the SLEEPDEEP bit of the Cortex®-M0+ System Control Register set to 1, it suspends program execution and stops operating. This state is referred to SLEEP mode in this IC. In this mode, the clock sources stop operating as well.
However, the clock source in which the CLGOSC.IOSCSLPC/OSC1SLPC/OSC3SLPC/EXOSCSLPC bit is set to 0 keeps operating, so the peripheral circuits with the clock being supplied can also operate. By setting this mode when no software processing and peripheral circuit operations are required, power consumption can be less than HALT mode.
The RAM retains data even in SLEEP mode.

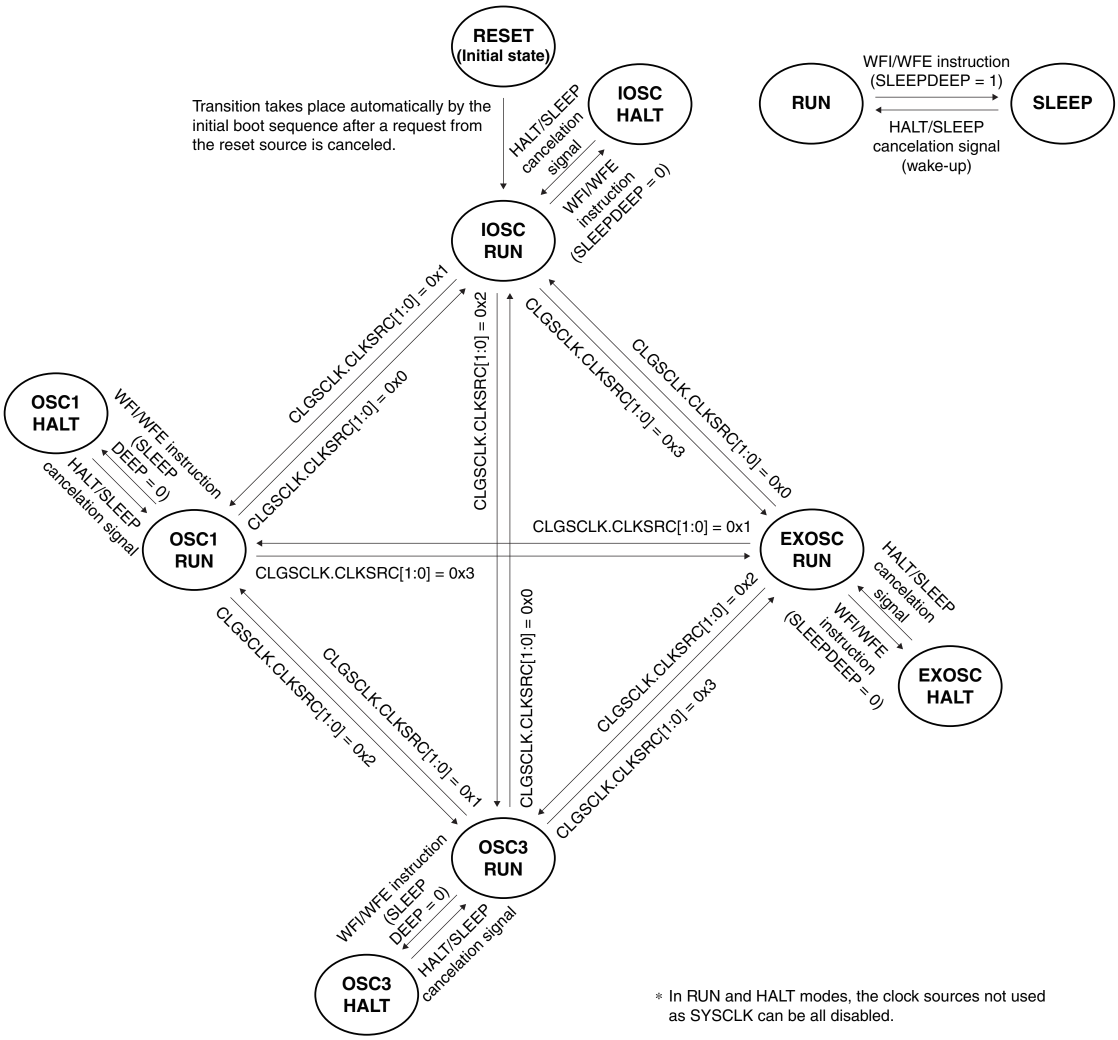


Figure 2.4.2.1 Operating Mode-to-Mode State Transition Diagram

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.

- Interrupt request from a peripheral circuit
- NMI from the watchdog timer
- 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
IOSC oscillation auto-trimming completion	CLGINTF.IOSCTEDIF	When the IOSC oscillation auto-trimming operation has completed	Writing 1
IOSC oscillation auto-trimming error	CLGINTF.IOSCTERIF	When the IOSC oscillation auto-trimming operation has terminated due to an error	Writing 1

CLG provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the CPU 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” chapter.

2.6 Control Registers

PWGA Control Register

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

Bits 15–6 Reserved

Bit 5 REGDIS

This bit enables the V_{D1} regulator discharge function.
1 (R/WP): Enable
0 (R/WP): Disable

Bit 4 REGSEL

This bit controls the V_{D1} regulator voltage mode.
1 (R/WP): mode0
0 (R/WP): mode1

Bits 3–2 Reserved

Bits 1–0 REGMODE[1:0]

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

Table 2.6.1 Internal Regulator Operating Mode

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

SRC Reset Request Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SRCRESETREQ	15–8	–	0x00	–	R	–
	7–5	–	0x0	–	R	
	4	PORBORREQ	1	H2	R/W	Cleared by writing 1.
	3	XRESETREQ	1	H2	R/W	
	2	WDTRSTREQ	0	H1	R/W	
	1	SVDRSTREQ	0	H1	R/W	
	0	KEYRSTREQ	0	H1	R/W	

Bits 15–5 Reserved

Bit 4 PORBORREQ
Bit 3 XRESETREQ
Bit 2 WDTRSTREQ
Bit 1 SVDRSTREQ
Bit 0 KEYRSTREQ

These bits indicate the reset request statuses.
1 (R): Reset requested
0 (R): No reset requested
1 (W): Clear flag
0 (W): Ineffective

Each bit corresponds to the reset source as follows:
SRCRESETREQ.PORBORREQ bit: POR and BOR
SRCRESETREQ.XRESETREQ bit: #RESET pin
SRCRESETREQ.WDTRSTREQ bit: Watchdog timer reset
SRCRESETREQ.SVDRSTREQ bit: Supply voltage detector reset
SRCRESETREQ.KEYRSTREQ bit: Key-entry reset

SRC #RESET Port Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SRCRESETPCTL	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6–4	(reserved)	0x0	H2	R/WP	Do not write 1.
	3	–	0	–	R	–
	2	(reserved)	0	H2	R/WP	Do not write 1.
	1	PORT_PLUP_EN	1	H2	R/WP	–
	0	PORT_RESET_EN	1	H2	R/WP	

Bits 15–2 Reserved

Bit 1 PORT_PLUP_EN

This bit enables the pull-up of the #RESET pin.
1 (R/WP): Enable pull-up
0 (R/WP): Disable pull-up

Bit 0 PORT_RESET_EN

This bit enables the #RESET pin to input the external reset signal.
1 (R/WP): Enable external reset input
0 (R/WP): Disable external reset input

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.

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	1/16	Reserved
0x2	1/4	Reserved	1/8	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 SYSCLK Clock Source and Division Ratio Settings

CLGSCLK. CLKDIV[1:0] bits	CLGSCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCCLK	OSC1CLK	OSC3CLK	EXOSCCLK
0x3	1/8	Reserved	1/16	Reserved
0x2	1/4	Reserved	1/8	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 IOSC Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGIOSC	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6–5	IOSCWT[1:0]	0x3	H0	R/WP	
	4	IOSCSTM	0	H0	R/WP	
	3	–	0	–	R	
	2–0	IOSCFQ[2:0]	0x3	H0	R/WP	

Bits 15–7 Reserved

Bits 6–5 IOSCWT[1:0]

These bits set the oscillation stabilization waiting time for the IOSC oscillator circuit (8 MHz–32 MHz).

Table 2.6.4 IOSC Oscillation Stabilization Waiting Time Setting

CLGIOSC.IOSCWT[1:0] bits	Oscillation stabilization waiting time
0x3	2,048 clocks
0x2	16 clocks
0x1	8 clocks
0x0	Setting prohibited

Bit 4 IOSCSTM

This bit controls the IOSCCLK 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 IOSCCLK as the system clock or peripheral circuit clocks while the CLGIOSC.IOSCSTM 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.
- Be sure to avoid altering the CLGIOSC.IOSCFQ[2:0] bits while the auto-trimming is being executed.

Bit 3 Reserved

Bits 2–0 IOSCFQ[2:0]

These bits select the IOSCCLK frequency.

Table 2.6.5 IOSCCLK Frequency Selection

CLGIOSC. IOSCFQ[2:0] bits	IOSCCLK frequency	
	V_{D1} voltage mode = mode0	V_{D1} voltage mode = mode1
0x7	32 MHz	Setting prohibited
0x6	24 MHz	
0x5	16 MHz	
0x4	12 MHz	
0x3	8 MHz	
0x2	Setting prohibited	
0x1	2 MHz	2 MHz
0x0	1 MHz	1 MHz

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 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 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 oscillator circuit.

Table 2.6.6 OSC1 Internal Gate Capacitance Setting

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

For more information, refer to “OSC1 oscillator circuit characteristics, Internal gate capacitance C_{GI1}” 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 oscillator circuit.

Table 2.6.7 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 oscillator circuit.

Table 2.6.8 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 2.6.9 OSC1 Oscillation Stabilization Waiting Time Setting

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

CLG OSC3 Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGOSC3	15–13	–	0x0	–	R	–
	12–10	OSC3FQ[2:0]	0x5	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	–	0	–	R	
	2–0	OSC3WT[2:0]	0x6	H0	R/WP	

Bits 15–13 Reserved

Bits 12–10 OSC3FQ[2:0]

These bits set the OSC3CLK frequency when internal oscillator is selected as the OSC3 oscillator type.

Table 2.6.10 OSC3CLK Frequency Setting (OSC3 Internal Oscillator)

CLGOSC3.OSC3FQ[2:0] bits	OSC3CLK frequency
0x7	32 MHz
0x6	24 MHz
0x5	16 MHz
0x4	12 MHz
0x3–0x0	8 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 when crystal/ceramic oscillator is selected as the OSC3 oscillator type.

Table 2.6.11 OSC3 Oscillation Inverter Gain Setting

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

Bit 3 Reserved

Bits 2–0 OSC3WT[2:0]

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

Table 2.6.12 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	15–9	–	0x00	–	R	–
	8	IOSCTERIF	0	H0	R/W	Cleared by writing 1.
	7	–	0	–	R	–
	6	(reserved)	0	H0	R	
	5	OSC1STPIF	0	H0	R/W	Cleared by writing 1.
	4	IOSCTEDIF	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 15–9, 7, 6, 3 Reserved

Bit 8 IOSCTERIF
Bit 5 OSC1STPIF
Bit 4 IOSCTEDIF
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.IOSCTERIF bit: IOSC oscillation auto-trimming error interrupt
CLGINTF.OSC1STPIF bit: OSC1 oscillation stop interrupt
CLGINTF.IOSCTEDIF bit: IOSC 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 IOSCCLK has already been stabilized.

CLG Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGINTE	15–9	–	0x00	–	R	–
	8	IOSCTERIE	0	H0	R/W	
	7	–	0	–	R	
	6	(reserved)	0	H0	R	
	5	OSC1STPIE	0	H0	R/W	
	4	IOSCTEDIE	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 15–9, 7, 6, 3 Reserved

Bit 8 IOSCTERIE
Bit 5 OSC1STPIE
Bit 4 IOSCTEDIE
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.IOSCTERIE bit: IOSC oscillation auto-trimming error interrupt
CLGINTE.OSC1STPIE bit: OSC1 oscillation stop interrupt
CLGINTE.IOSCTEDIE bit: IOSC 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

CLG FOUT Control Register

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

Bits 15–7 Reserved

Bits 6–4 FOUTDIV[2:0]

These bits set the FOUT clock division ratio.

Bits 3–2 FOUTSRC[1:0]

These bits select the FOUT clock source.

Table 2.6.13 FOUT Clock Source and Division Ratio Settings

CLGFOUT. FOUTDIV[2:0] bits	CLGFOUT.FOUTSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSCCLK	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.

CLG Oscillation Frequency Trimming Register 1

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

Bits 15–14 Reserved

Bits 13–8 IOSCLSAJ[5:0]

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

This setting affects the low-speed oscillation frequencies (1 MHz and 2 MHz).

Table 2.6.14 Low-Speed Oscillation Frequency Trimming Setting of IOSC Internal Oscillator Circuit

CLGTRIM1.IOSCLSAJ[5:0] bits	IOSC oscillation frequency (2/1 MHz)
0x3f	High
:	:
0x00	Low

Bit 7 Reserved

Bits 6–0 IOSCHSAJ[6:0]

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

This setting affects the high-speed oscillation frequencies (8 MHz to 32 MHz).

Table 2.6.15 High-Speed Oscillation Frequency Trimming Setting of IOSC Internal Oscillator Circuit

CLGTRIM1.IOSCHSAJ[6:0] bits	IOSC oscillation frequency (32/24/16/12/8 MHz)
0x7f	High
:	:
0x00	Low

Note: The initial values of the CLGTRIM1.IOSCLSAJ[5:0] and CLGTRIM1.IOSCHSAJ[6:0] bits were adjusted so that the IOSC oscillator circuit characteristics described in the “Electrical Characteristics” chapter can be guaranteed. Be aware that the frequency characteristics may not be satisfied when these settings are altered. When altering these settings, always make sure that the IOSC oscillator circuit is inactive.

CLG Oscillation Frequency Trimming Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
CLGTRIM2	15	–	0	–	R	–
	14–8	OSC3SAJ[6:0]	*	H0	R/WP	* Determined by factory adjustment.
	7–6	–	0x0	–	R	
	5–0	OSC1SAJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.

Bit 15 Reserved

Bits 14–8 OSC3SAJ[6:0]

These bits set the frequency trimming value for the OSC3 internal oscillator circuit.
This setting does not affect the OSC3 crystal/ceramic oscillation frequency.

Table 2.6.16 Oscillation Frequency Trimming Setting of OSC3 Internal Oscillator Circuit

CLGTRIM2.OSC3SAJ[6:0] bits	OSC3 internal oscillator frequency
0x7f	High
:	:
0x00	Low

Note: The initial value of the CLGTRIM2.OSC3SAJ[6:0] bits was adjusted so that the OSC3 oscillator circuit characteristics described in the “Electrical Characteristics” chapter can be guaranteed. Be aware that the frequency characteristic may not be satisfied when this setting is altered. When altering this setting, always make sure that the OSC3 oscillator circuit is inactive.

Bits 7–6 Reserved

Bits 5–0 OSC1SAJ[5:0]

These bits set the frequency trimming value for the OSC1 internal oscillator circuit.
This setting does not affect the OSC1 crystal oscillation frequency.

Table 2.6.17 Oscillation Frequency Trimming Setting of OSC1 Internal Oscillator Circuit

CLGTRIM2.OSC1SAJ[5:0] bits	OSC1 internal oscillator frequency
0x3f	High
:	:
0x00	Low

Note: The initial value of the CLGTRIM2.OSC1SAJ[5:0] bits was adjusted so that the OSC1 oscillator circuit characteristics described in the “Electrical Characteristics” chapter can be guaranteed. Be aware that the frequency characteristic may not be satisfied when this setting is altered. When altering this setting, always make sure that the OSC1 oscillator circuit is inactive.

3 CPU and Debugger

3.1 Overview

This IC incorporates a Cortex®-M0+ CPU manufactured by Arm Ltd.

3.2 CPU

The following shows the system configuration of the Cortex®-M0+ CPU embedded in this IC:

- Cortex®-M0+ core
- 32-bit single-cycle multiplier
- Nested vectored interrupt controller (NVIC)
- System timer (Systick)
- Serial-wire debug port (SW-DP)
- Micro trace buffer (MTB)
- Number of hardware break points: 4
- Number of watch points: 2

3.3 Debugger

This IC includes a serial-wire debug port (SW-DP).

3.3.1 List of Debugger Input/Output Pins

Table 3.3.3.1 lists the debug pins.

Table 3.3.1.1 List of Debug Pins

Pin name	I/O	Initial state	Function
SWCLK	O	O	On-chip debugger clock input pin Input a clock from a debugging tool.
SWD	I/O	I	On-chip debugger data input/output pin Used to input/output debugging data.

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.

3.3.2 External Connection

Figure 3.3.2.1 shows a connection example between this IC and a debugging tool when performing debugging.

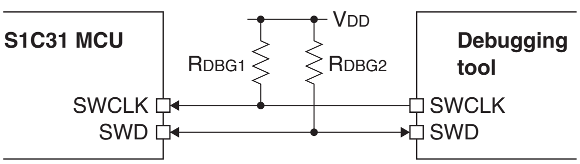


Figure 3.3.2.1 External Connection

For the recommended pull-up resistor value, refer to "Recommended Operating Conditions, Debug pin pull-up resistors R_{DBG1-2}" in the "Electrical Characteristics" chapter. R_{DBG1} and R_{DBG2} are not required when using the debug pins as general-purpose I/O port pins.

3.4 Reference Documents

Arm Ltd. provides various documents for developing a system with a Cortex®-M0+ CPU included. For detailed information on the Cortex®-M0+ CPU that are not described in this manual, refer to the following documents:

1. ARM®v6-M Architecture Reference Manual
2. Cortex®-M0+Technical Reference Manual
3. Cortex®-M0+ Devices Generic User Guide

These documents can be downloaded from the document site of Arm Ltd.

https://developer.arm.com/documentation

4 Memory and Bus

4.1 Overview

This IC supports up to 4G bytes of accessible memory space for both instructions and data.
The features are listed below.

- Embedded Flash memory that supports on-board programming
- Write-protect function to protect system control registers

Figure 4.1.1 shows the memory map.

0xffff ffff 0xf022 4000	Reserved
0xf022 3fff 0xf020 0000	MTB area (144K bytes) (Device size: 32 bits)
0xf01f ffff 0xf000 1000	Reserved
0xf000 0fff 0xf000 0000	System ROM table area (4K bytes) (Device size: 32 bits)
0xefff ffff 0xe000 0000	Reserved area for Cortex®-M0+ (256M bytes) (Device size: 32 bits)
0xdfff ffff 0x4000 3000	Reserved
0x4000 2fff 0x4000 1000	Peripheral circuit area (8K bytes) (Device size: 32 bits)
0x4000 0fff 0x4000 0000	Peripheral circuit area (4K bytes) (Device size: 16 bits)
0x3fff ffff 0x2020 0200	Reserved
0x2020 01ff 0x2020 0000	Display data RAM area (112 bytes) (Device size: 32 bits)
0x201f ffff 0x2000 4000	Reserved
0x2000 3fff 0x2000 0000	RAM area (16K bytes) (Device size: 32 bits)
0x1fff ffff 0x0002 0000	Reserved
0x0001 ffff 0x0000 0000	Flash area (128K bytes) (Device size: 32 bits)

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., LDR Rt, [Rn] → 32-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- or 16-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

4.3 Flash Memory

The Flash memory is used to store application programs and data. Address 0x0 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” 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

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 using a flash loader. The V_{PP} voltage can be supplied from the internal voltage booster. Be sure to connect C_{VPP} between the V_{SS} and V_{PP} pins for generating the voltage using the internal power supply.

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

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.

4.5 Display Data RAM

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

4.6 Peripheral Circuit Control Registers

The control registers for the peripheral circuits are located in the 12K-byte area beginning with address 0x4000 0000. Table 4.6.1 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 4.6.1 Peripheral Circuit Control Register Map

Peripheral circuit	Address	Register name	
System register (SYS)	0x4000 0000	SYSPROT	System Protect Register
Power generator (PWGA)	0x4000 0020	PWGACTL	PWGA Control Register
Clock generator(CLG)	0x4000 0040	CLGSCLK	CLG System Clock Control Register
	0x4000 0042	CLGOSC	CLG Oscillation Control Register
	0x4000 0044	CLGIOSC	CLG IOSC Control Register
	0x4000 0046	CLGOSC1	CLG OSC1 Control Register
	0x4000 0048	CLGOSC3	CLG OSC3 Control Register
	0x4000 004c	CLGINTF	CLG Interrupt Flag Register
	0x4000 004e	CLGINTE	CLG Interrupt Enable Register
	0x4000 0050	CLGFOUT	CLG FOUT Control Register
	0x4000 0052	CLGTRIM1	CLG Oscillation Frequency Trimming Register 1
	0x4000 0054	CLGTRIM2	CLG Oscillation Frequency Trimming Register 2
System reset controller (SRC)	0x4000 0060	SRCRESETREQ	SRC Reset Request Flag Register
	0x4000 0062	SRCRESETPCTL	SRC #RESET Port Control Register
Watchdog timer (WDT2)	0x4000 00a0	WDT2CLK	WDT2 Clock Control Register
	0x4000 00a2	WDT2CTL	WDT2 Control Register
	0x4000 00a4	WDT2CMP	WDT2 Counter Compare Match Register
Real-time clock (RTCA)	0x4000 00c0	RTCACTLL	RTCA Control Register (Low Byte)
	0x4000 00c1	RTCACTLH	RTCA Control Register (High Byte)
	0x4000 00c2	RTCAALM1	RTCA Second Alarm Register
	0x4000 00c4	RTCAALM2	RTCA Hour/Minute Alarm Register
	0x4000 00c6	RTCASWCTL	RTCA Stopwatch Control Register
	0x4000 00c8	RTCASEC	RTCA Second/1Hz Register
	0x4000 00ca	RTCAHUR	RTCA Hour/Minute Register
	0x4000 00cc	RTCAMON	RTCA Month/Day Register
	0x4000 00ce	RTCAYAR	RTCA Year/Week Register
	0x4000 00d0	RTCAINTF	RTCA Interrupt Flag Register
	0x4000 00d2	RTCAINTE	RTCA Interrupt Enable Register
Supply voltage detector (SVD4) Ch.0	0x4000 0100	SVD4_0CLK	SVD4 Ch.0 Clock Control Register
	0x4000 0102	SVD4_0CTL	SVD4 Ch.0 Control Register
	0x4000 0104	SVD4_0INTF	SVD4 Ch.0 Status and Interrupt Flag Register
	0x4000 0106	SVD4_0INTE	SVD4 Ch.0 Interrupt Enable Register
16-bit timer (T16) Ch.0	0x4000 0140	T16_0CLK	T16 Ch.0 Clock Control Register
	0x4000 0142	T16_0MOD	T16 Ch.0 Mode Register
	0x4000 0144	T16_0CTL	T16 Ch.0 Control Register
	0x4000 0146	T16_0TR	T16 Ch.0 Reload Data Register
	0x4000 0148	T16_0TC	T16 Ch.0 Counter Data Register
	0x4000 014a	T16_0INTF	T16 Ch.0 Interrupt Flag Register
	0x4000 014c	T16_0INTE	T16 Ch.0 Interrupt Enable Register
Flash controller (FLASHC)	0x4000 01b0	FLASHCWAIT	FLASHC Flash Read Cycle Register

Peripheral circuit	Address	Register name	
I/O ports (PPORT)	0x4000 0200	PPORTP0DAT	P0 Port Data Register
	0x4000 0202	PPORTP0IOEN	P0 Port Enable Register
	0x4000 0204	PPORTP0RCTL	P0 Port Pull-up/down Control Register
	0x4000 0206	PPORTP0INTF	P0 Port Interrupt Flag Register
	0x4000 0208	PPORTP0INTCTL	P0 Port Interrupt Control Register
	0x4000 020a	PPORTP0CHATEN	P0 Port Chattering Filter Enable Register
	0x4000 020c	PPORTP0MODSEL	P0 Port Mode Select Register
	0x4000 020e	PPORTP0FNCSEL	P0 Port Function Select Register
	0x4000 0210	PPORTP1DAT	P1 Port Data Register
	0x4000 0212	PPORTP1IOEN	P1 Port Enable Register
	0x4000 0214	PPORTP1RCTL	P1 Port Pull-up/down Control Register
	0x4000 0216	PPORTP1INTF	P1 Port Interrupt Flag Register
	0x4000 0218	PPORTP1INTCTL	P1 Port Interrupt Control Register
	0x4000 021a	PPORTP1CHATEN	P1 Port Chattering Filter Enable Register
	0x4000 021c	PPORTP1MODSEL	P1 Port Mode Select Register
	0x4000 021e	PPORTP1FNCSEL	P1 Port Function Select Register
	0x4000 0220	PPORTP2DAT	P2 Port Data Register
	0x4000 0222	PPORTP2IOEN	P2 Port Enable Register
	0x4000 0224	PPORTP2RCTL	P2 Port Pull-up/down Control Register
	0x4000 0226	PPORTP2INTF	P2 Port Interrupt Flag Register
	0x4000 0228	PPORTP2INTCTL	P2 Port Interrupt Control Register
	0x4000 022a	PPORTP2CHATEN	P2 Port Chattering Filter Enable Register
	0x4000 022c	PPORTP2MODSEL	P2 Port Mode Select Register
	0x4000 022e	PPORTP2FNCSEL	P2 Port Function Select Register
	0x4000 0230	PPORTP3DAT	P3 Port Data Register
	0x4000 0232	PPORTP3IOEN	P3 Port Enable Register
	0x4000 0234	PPORTP3RCTL	P3 Port Pull-up/down Control Register
	0x4000 0236	PPORTP3INTF	P3 Port Interrupt Flag Register
	0x4000 0238	PPORTP3INTCTL	P3 Port Interrupt Control Register
	0x4000 023a	PPORTP3CHATEN	P3 Port Chattering Filter Enable Register
	0x4000 023c	PPORTP3MODSEL	P3 Port Mode Select Register
	0x4000 023e	PPORTP3FNCSEL	P3 Port Function Select Register
	0x4000 0240	PPORTP4DAT	P4 Port Data Register
	0x4000 0242	PPORTP4IOEN	P4 Port Enable Register
	0x4000 0244	PPORTP4RCTL	P4 Port Pull-up/down Control Register
	0x4000 0246	PPORTP4INTF	P4 Port Interrupt Flag Register
	0x4000 0248	PPORTP4INTCTL	P4 Port Interrupt Control Register
	0x4000 024a	PPORTP4CHATEN	P4 Port Chattering Filter Enable Register
	0x4000 024c	PPORTP4MODSEL	P4 Port Mode Select Register
	0x4000 024e	PPORTP4FNCSEL	P4 Port Function Select Register
	0x4000 0250	PPORTP5DAT	P5 Port Data Register
	0x4000 0252	PPORTP5IOEN	P5 Port Enable Register
	0x4000 0254	PPORTP5RCTL	P5 Port Pull-up/down Control Register
	0x4000 0256	PPORTP5INTF	P5 Port Interrupt Flag Register
	0x4000 0258	PPORTP5INTCTL	P5 Port Interrupt Control Register
	0x4000 025a	PPORTP5CHATEN	P5 Port Chattering Filter Enable Register
	0x4000 025c	PPORTP5MODSEL	P5 Port Mode Select Register
	0x4000 025e	PPORTP5FNCSEL	P5 Port Function Select Register
	0x4000 0260	PPORTP6DAT	P6 Port Data Register
	0x4000 0262	PPORTP6IOEN	P6 Port Enable Register
	0x4000 0264	PPORTP6RCTL	P6 Port Pull-up/down Control Register
	0x4000 0266	PPORTP6INTF	P6 Port Interrupt Flag Register
	0x4000 0268	PPORTP6INTCTL	P6 Port Interrupt Control Register
	0x4000 026a	PPORTP6CHATEN	P6 Port Chattering Filter Enable Register
	0x4000 026c	PPORTP6MODSEL	P6 Port Mode Select Register
	0x4000 026e	PPORTP6FNCSEL	P6 Port Function Select Register
	0x4000 02d0	PPORTPDDAT	Pd Port Data Register
	0x4000 02d2	PPORTPDIOEN	Pd Port Enable Register
	0x4000 02d4	PPORTPDRCTL	Pd Port Pull-up/down Control Register
	0x4000 02dc	PPORTPDMODSEL	Pd Port Mode Select Register
	0x4000 02de	PPORTPDFNCSEL	Pd Port Function Select Register
	0x4000 02e0	PPORTCLK	P Port Clock Control Register
	0x4000 02e2	PPORTINTFGRP	P Port Interrupt Flag Group Register

Peripheral circuit	Address	Register name	
Universal port multiplexer (UPMUX)	0x4000 0300	UPMUXP0MUX0	P00–01 Universal Port Multiplexer Setting Register
	0x4000 0302	UPMUXP0MUX1	P02–03 Universal Port Multiplexer Setting Register
	0x4000 0304	UPMUXP0MUX2	P04–05 Universal Port Multiplexer Setting Register
	0x4000 0306	UPMUXP0MUX3	P06–07 Universal Port Multiplexer Setting Register
	0x4000 0308	UPMUXP1MUX0	P10–11 Universal Port Multiplexer Setting Register
	0x4000 030a	UPMUXP1MUX1	P12–13 Universal Port Multiplexer Setting Register
	0x4000 030c	UPMUXP1MUX2	P14–15 Universal Port Multiplexer Setting Register
	0x4000 030e	UPMUXP1MUX3	P16–17 Universal Port Multiplexer Setting Register
	0x4000 0310	UPMUXP2MUX0	P20–21 Universal Port Multiplexer Setting Register
	0x4000 0312	UPMUXP2MUX1	P22–23 Universal Port Multiplexer Setting Register
	0x4000 0314	UPMUXP2MUX2	P24–25 Universal Port Multiplexer Setting Register
	0x4000 0316	UPMUXP2MUX3	P26–27 Universal Port Multiplexer Setting Register
	0x4000 0318	UPMUXP3MUX0	P30–31 Universal Port Multiplexer Setting Register
	0x4000 031a	UPMUXP3MUX1	P32–33 Universal Port Multiplexer Setting Register
	0x4000 031c	UPMUXP3MUX2	P34–35 Universal Port Multiplexer Setting Register
	0x4000 031e	UPMUXP3MUX3	P36–37 Universal Port Multiplexer Setting Register
UART (UART3) Ch.0	0x4000 0380	UART3_0CLK	UART3 Ch.0 Clock Control Register
	0x4000 0382	UART3_0MOD	UART3 Ch.0 Mode Register
	0x4000 0384	UART3_0BR	UART3 Ch.0 Baud-Rate Register
	0x4000 0386	UART3_0CTL	UART3 Ch.0 Control Register
	0x4000 0388	UART3_0TXD	UART3 Ch.0 Transmit Data Register
	0x4000 038a	UART3_0RXD	UART3 Ch.0 Receive Data Register
	0x4000 038c	UART3_0INTF	UART3 Ch.0 Status and Interrupt Flag Register
	0x4000 038e	UART3_0INTE	UART3 Ch.0 Interrupt Enable Register
	0x4000 0390	UART3_0 TBEDMAEN	UART3 Ch.0 Transmit Buffer Empty DMA Request Enable Register
	0x4000 0392	UART3_0 RB1FDMAEN	UART3 Ch.0 Receive Buffer One Byte Full DMA Request Enable Register
	0x4000 0394	UART3_0CAWF	UART3 Ch.0 Carrier Waveform Register
16-bit timer (T16) Ch.1	0x4000 03a0	T16_1CLK	T16 Ch.1 Clock Control Register
	0x4000 03a2	T16_1MOD	T16 Ch.1 Mode Register
	0x4000 03a4	T16_1CTL	T16 Ch.1 Control Register
	0x4000 03a6	T16_1TR	T16 Ch.1 Reload Data Register
	0x4000 03a8	T16_1TC	T16 Ch.1 Counter Data Register
	0x4000 03aa	T16_1INTF	T16 Ch.1 Interrupt Flag Register
	0x4000 03ac	T16_1INTE	T16 Ch.1 Interrupt Enable Register
Synchronous serial interface (SPIA) Ch.0	0x4000 03b0	SPIA_0MOD	SPIA Ch.0 Mode Register
	0x4000 03b2	SPIA_0CTL	SPIA Ch.0 Control Register
	0x4000 03b4	SPIA_0TXD	SPIA Ch.0 Transmit Data Register
	0x4000 03b6	SPIA_0RXD	SPIA Ch.0 Receive Data Register
	0x4000 03b8	SPIA_0INTF	SPIA Ch.0 Interrupt Flag Register
	0x4000 03ba	SPIA_0INTE	SPIA Ch.0 Interrupt Enable Register
	0x4000 03bc	SPIA_0TBEDMAEN	SPIA Ch.0 Transmit Buffer Empty DMA Request Enable Register
	0x4000 03be	SPIA_0RBFDMAEN	SPIA Ch.0 Receive Buffer Full DMA Request Enable Register
I^2C (I2C) Ch.0	0x4000 03c0	I2C_0CLK	I2C Ch.0 Clock Control Register
	0x4000 03c2	I2C_0MOD	I2C Ch.0 Mode Register
	0x4000 03c4	I2C_0BR	I2C Ch.0 Baud-Rate Register
	0x4000 03c8	I2C_0OADR	I2C Ch.0 Own Address Register
	0x4000 03ca	I2C_0CTL	I2C Ch.0 Control Register
	0x4000 03cc	I2C_0TXD	I2C Ch.0 Transmit Data Register
	0x4000 03ce	I2C_0RXD	I2C Ch.0 Receive Data Register
	0x4000 03d0	I2C_0INTF	I2C Ch.0 Status and Interrupt Flag Register
	0x4000 03d2	I2C_0INTE	I2C Ch.0 Interrupt Enable Register
	0x4000 03d4	I2C_0TBEDMAEN	I2C Ch.0 Transmit Buffer Empty DMA Request Enable Register
	0x4000 03d6	I2C_0RBFDMAEN	I2C Ch.0 Receive Buffer Full DMA Request Enable Register

Peripheral circuit	Address	Register name	
16-bit PWM timer (T16B) Ch.0	0x4000 0400	T16B_0CLK	T16B Ch.0 Clock Control Register
	0x4000 0402	T16B_0CTL	T16B Ch.0 Counter Control Register
	0x4000 0404	T16B_0MC	T16B Ch.0 Max Counter Data Register
	0x4000 0406	T16B_0TC	T16B Ch.0 Timer Counter Data Register
	0x4000 0408	T16B_0CS	T16B Ch.0 Counter Status Register
	0x4000 040a	T16B_0INTF	T16B Ch.0 Interrupt Flag Register
	0x4000 040c	T16B_0INTE	T16B Ch.0 Interrupt Enable Register
	0x4000 040e	T16B_0MZDMAEN	T16B Ch.0 Counter Max/Zero DMA Request Enable Register
	0x4000 0410	T16B_0CCCTL0	T16B Ch.0 Compare/Capture 0 Control Register
	0x4000 0412	T16B_0CCR0	T16B Ch.0 Compare/Capture 0 Data Register
	0x4000 0414	T16B_0CC0DMAEN	T16B Ch.0 Compare/Capture 0 DMA Request Enable Register
	0x4000 0418	T16B_0CCCTL1	T16B Ch.0 Compare/Capture 1 Control Register
	0x4000 041a	T16B_0CCR1	T16B Ch.0 Compare/Capture 1 Data Register
	0x4000 041c	T16B_0CC1DMAEN	T16B Ch.0 Compare/Capture 1 DMA Request Enable Register
	0x4000 0420	T16B_0CCCTL2	T16B Ch.0 Compare/Capture 2 Control Register
	0x4000 0422	T16B_0CCR2	T16B Ch.0 Compare/Capture 2 Data Register
	0x4000 0424	T16B_0CC2DMAEN	T16B Ch.0 Compare/Capture 2 DMA Request Enable Register
	0x4000 0428	T16B_0CCCTL3	T16B Ch.0 Compare/Capture 3 Control Register
	0x4000 042a	T16B_0CCR3	T16B Ch.0 Compare/Capture 3 Data Register
	0x4000 042c	T16B_0CC3DMAEN	T16B Ch.0 Compare/Capture 3 DMA Request Enable Register
16-bit PWM timer (T16B) Ch.1	0x4000 0440	T16B_1CLK	T16B Ch.1 Clock Control Register
	0x4000 0442	T16B_1CTL	T16B Ch.1 Counter Control Register
	0x4000 0444	T16B_1MC	T16B Ch.1 Max Counter Data Register
	0x4000 0446	T16B_1TC	T16B Ch.1 Timer Counter Data Register
	0x4000 0448	T16B_1CS	T16B Ch.1 Counter Status Register
	0x4000 044a	T16B_1INTF	T16B Ch.1 Interrupt Flag Register
	0x4000 044c	T16B_1INTE	T16B Ch.1 Interrupt Enable Register
	0x4000 044e	T16B_1MZDMAEN	T16B Ch.1 Counter Max/Zero DMA Request Enable Register
	0x4000 0450	T16B_1CCCTL0	T16B Ch.1 Compare/Capture 0 Control Register
	0x4000 0452	T16B_1CCR0	T16B Ch.1 Compare/Capture 0 Data Register
	0x4000 0454	T16B_1CC0DMAEN	T16B Ch.1 Compare/Capture 0 DMA Request Enable Register
	0x4000 0458	T16B_1CCCTL1	T16B Ch.1 Compare/Capture 1 Control Register
	0x4000 045a	T16B_1CCR1	T16B Ch.1 Compare/Capture 1 Data Register
	0x4000 045c	T16B_1CC1DMAEN	T16B Ch.1 Compare/Capture 1 DMA Request Enable Register
	0x4000 0460	T16B_1CCCTL2	T16B Ch.1 Compare/Capture 2 Control Register
	0x4000 0462	T16B_1CCR2	T16B Ch.1 Compare/Capture 2 Data Register
	0x4000 0464	T16B_1CC2DMAEN	T16B Ch.1 Compare/Capture 2 DMA Request Enable Register
	0x4000 0468	T16B_1CCCTL3	T16B Ch.1 Compare/Capture 3 Control Register
	0x4000 046a	T16B_1CCR3	T16B Ch.1 Compare/Capture 3 Data Register
	0x4000 046c	T16B_1CC3DMAEN	T16B Ch.1 Compare/Capture 3 DMA Request Enable Register
16-bit timer (T16) Ch.3	0x4000 0480	T16_3CLK	T16 Ch.3 Clock Control Register
	0x4000 0482	T16_3MOD	T16 Ch.3 Mode Register
	0x4000 0484	T16_3CTL	T16 Ch.3 Control Register
	0x4000 0486	T16_3TR	T16 Ch.3 Reload Data Register
	0x4000 0488	T16_3TC	T16 Ch.3 Counter Data Register
	0x4000 048a	T16_3INTF	T16 Ch.3 Interrupt Flag Register
	0x4000 048c	T16_3INTE	T16 Ch.3 Interrupt Enable Register
16-bit timer (T16) Ch.4	0x4000 04a0	T16_4CLK	T16 Ch.4 Clock Control Register
	0x4000 04a2	T16_4MOD	T16 Ch.4 Mode Register
	0x4000 04a4	T16_4CTL	T16 Ch.4 Control Register
	0x4000 04a6	T16_4TR	T16 Ch.4 Reload Data Register
	0x4000 04a8	T16_4TC	T16 Ch.4 Counter Data Register
	0x4000 04aa	T16_4INTF	T16 Ch.4 Interrupt Flag Register
	0x4000 04ac	T16_4INTE	T16 Ch.4 Interrupt Enable Register

Peripheral circuit	Address	Register name	
16-bit timer (T16) Ch.5	0x4000 04c0	T16_5CLK	T16 Ch.5 Clock Control Register
	0x4000 04c2	T16_5MOD	T16 Ch.5 Mode Register
	0x4000 04c4	T16_5CTL	T16 Ch.5 Control Register
	0x4000 04c6	T16_5TR	T16 Ch.5 Reload Data Register
	0x4000 04c8	T16_5TC	T16 Ch.5 Counter Data Register
	0x4000 04ca	T16_5INTF	T16 Ch.5 Interrupt Flag Register
	0x4000 04cc	T16_5INTE	T16 Ch.5 Interrupt Enable Register
UART (UART3) Ch.1	0x4000 0600	UART3_1CLK	UART3 Ch.1 Clock Control Register
	0x4000 0602	UART3_1MOD	UART3 Ch.1 Mode Register
	0x4000 0604	UART3_1BR	UART3 Ch.1 Baud-Rate Register
	0x4000 0606	UART3_1CTL	UART3 Ch.1 Control Register
	0x4000 0608	UART3_1TXD	UART3 Ch.1 Transmit Data Register
	0x4000 060a	UART3_1RXD	UART3 Ch.1 Receive Data Register
	0x4000 060c	UART3_1INTF	UART3 Ch.1 Status and Interrupt Flag Register
	0x4000 060e	UART3_1INTE	UART3 Ch.1 Interrupt Enable Register
	0x4000 0610	UART3_1 TBEDMAEN	UART3 Ch.1 Transmit Buffer Empty DMA Request Enable Register
	0x4000 0612	UART3_1 RB1FDMAEN	UART3 Ch.1 Receive Buffer One Byte Full DMA Request Enable Register
	0x4000 0614	UART3_1CAWF	UART3 Ch.1 Carrier Waveform Register
16-bit timer (T16) Ch.6	0x4000 0660	T16_6CLK	T16 Ch.6 Clock Control Register
	0x4000 0662	T16_6MOD	T16 Ch.6 Mode Register
	0x4000 0664	T16_6CTL	T16 Ch.6 Control Register
	0x4000 0666	T16_6TR	T16 Ch.6 Reload Data Register
	0x4000 0668	T16_6TC	T16 Ch.6 Counter Data Register
	0x4000 066a	T16_6INTF	T16 Ch.6 Interrupt Flag Register
	0x4000 066c	T16_6INTE	T16 Ch.6 Interrupt Enable Register
Synchronous serial interface (SPIA) Ch.1	0x4000 0670	SPIA_1MOD	SPIA Ch.1 Mode Register
	0x4000 0672	SPIA_1CTL	SPIA Ch.1 Control Register
	0x4000 0674	SPIA_1TXD	SPIA Ch.1 Transmit Data Register
	0x4000 0676	SPIA_1RXD	SPIA Ch.1 Receive Data Register
	0x4000 0678	SPIA_1INTF	SPIA Ch.1 Interrupt Flag Register
	0x4000 067a	SPIA_1INTE	SPIA Ch.1 Interrupt Enable Register
	0x4000 067c	SPIA_1TBEDMAEN	SPIA Ch.1 Transmit Buffer Empty DMA Request Enable Register
	0x4000 067e	SPIA_1RBFDMAEN	SPIA Ch.1 Receive Buffer Full DMA Request Enable Register
16-bit timer (T16) Ch.2	0x4000 0680	T16_2CLK	T16 Ch.2 Clock Control Register
	0x4000 0682	T16_2MOD	T16 Ch.2 Mode Register
	0x4000 0684	T16_2CTL	T16 Ch.2 Control Register
	0x4000 0686	T16_2TR	T16 Ch.2 Reload Data Register
	0x4000 0688	T16_2TC	T16 Ch.2 Counter Data Register
	0x4000 068a	T16_2INTF	T16 Ch.2 Interrupt Flag Register
	0x4000 068c	T16_2INTE	T16 Ch.2 Interrupt Enable Register
I²C (I2C) Ch.1	0x4000 06c0	I2C_1CLK	I2C Ch.1 Clock Control Register
	0x4000 06c2	I2C_1MOD	I2C Ch.1 Mode Register
	0x4000 06c4	I2C_1BR	I2C Ch.1 Baud-Rate Register
	0x4000 06c8	I2C_1OADR	I2C Ch.1 Own Address Register
	0x4000 06ca	I2C_1CTL	I2C Ch.1 Control Register
	0x4000 06cc	I2C_1TXD	I2C Ch.1 Transmit Data Register
	0x4000 06ce	I2C_1RXD	I2C Ch.1 Receive Data Register
	0x4000 06d0	I2C_1INTF	I2C Ch.1 Status and Interrupt Flag Register
	0x4000 06d2	I2C_1INTE	I2C Ch.1 Interrupt Enable Register
	0x4000 06d4	I2C_1TBEDMAEN	I2C Ch.1 Transmit Buffer Empty DMA Request Enable Register
	0x4000 06d6	I2C_1RBFDMAEN	I2C Ch.1 Receive Buffer Full DMA Request Enable Register
Sound generator (SNDA)	0x4000 0700	SNDACLK	SNDA Clock Control Register
	0x4000 0702	SNDASEL	SNDA Select Register
	0x4000 0704	SNDACTL	SNDA Control Register
	0x4000 0706	SNDADAT	SNDA Data Register
	0x4000 0708	SNDAINTF	SNDA Interrupt Flag Register
	0x4000 070a	SNDAINTE	SNDA Interrupt Enable Register
	0x4000 070c	SNDAEMDMAEN	SNDA Sound Buffer Empty DMA Request Enable Register

Peripheral circuit	Address	Register name	
IR remote controller (REMC3)	0x4000 0720	REMC3CLK	REMC3 Clock Control Register
	0x4000 0722	REMC3DBCTL	REMC3 Data Bit Counter Control Register
	0x4000 0724	REMC3DBCNT	REMC3 Data Bit Counter Register
	0x4000 0726	REMC3APLEN	REMC3 Data Bit Active Pulse Length Register
	0x4000 0728	REMC3DBLEN	REMC3 Data Bit Length Register
	0x4000 072a	REMC3INTF	REMC3 Status and Interrupt Flag Register
	0x4000 072c	REMC3INTE	REMC3 Interrupt Enable Register
	0x4000 0730	REMC3CARR	REMC3 Carrier Waveform Register
	0x4000 0732	REMC3CCTL	REMC3 Carrier Modulation Control Register
16-bit PWM timer (T16B) Ch.2	0x4000 0740	T16B_2CLK	T16B Ch.2 Clock Control Register
	0x4000 0742	T16B_2CTL	T16B Ch.2 Counter Control Register
	0x4000 0744	T16B_2MC	T16B Ch.2 Max Counter Data Register
	0x4000 0746	T16B_2TC	T16B Ch.2 Timer Counter Data Register
	0x4000 0748	T16B_2CS	T16B Ch.2 Counter Status Register
	0x4000 074a	T16B_2INTF	T16B Ch.2 Interrupt Flag Register
	0x4000 074c	T16B_2INTE	T16B Ch.2 Interrupt Enable Register
	0x4000 074e	T16B_2MZDMAEN	T16B Ch.2 Counter Max/Zero DMA Request Enable Register
	0x4000 0750	T16B_2CCCTL0	T16B Ch.2 Compare/Capture 0 Control Register
	0x4000 0752	T16B_2CCR0	T16B Ch.2 Compare/Capture 0 Data Register
	0x4000 0754	T16B_2CC0DMAEN	T16B Ch.2 Compare/Capture 0 DMA Request Enable Register
	0x4000 0758	T16B_2CCCTL1	T16B Ch.2 Compare/Capture 1 Control Register
	0x4000 075a	T16B_2CCR1	T16B Ch.2 Compare/Capture 1 Data Register
	0x4000 075c	T16B_2CC1DMAEN	T16B Ch.2 Compare/Capture 1 DMA Request Enable Register
	0x4000 0760	T16B_2CCCTL2	T16B Ch.2 Compare/Capture 2 Control Register
	0x4000 0762	T16B_2CCR2	T16B Ch.2 Compare/Capture 2 Data Register
	0x4000 0764	T16B_2CC2DMAEN	T16B Ch.2 Compare/Capture 2 DMA Request Enable Register
	0x4000 0768	T16B_2CCCTL3	T16B Ch.2 Compare/Capture 3 Control Register
	0x4000 076a	T16B_2CCR3	T16B Ch.2 Compare/Capture 3 Data Register
16-bit timer (T16) Ch.7	0x4000 0780	T16_7CLK	T16 Ch.7 Clock Control Register
	0x4000 0782	T16_7MOD	T16 Ch.7 Mode Register
	0x4000 0784	T16_7CTL	T16 Ch.7 Control Register
	0x4000 0786	T16_7TR	T16 Ch.7 Reload Data Register
	0x4000 0788	T16_7TC	T16 Ch.7 Counter Data Register
	0x4000 078a	T16_7INTF	T16 Ch.7 Interrupt Flag Register
	0x4000 078c	T16_7INTE	T16 Ch.7 Interrupt Enable Register
12-bit A/D converter (ADC12A)	0x4000 07a2	ADC12A_0CTL	ADC12A Ch.0 Control Register
	0x4000 07a4	ADC12A_0TRG	ADC12A Ch.0 Trigger/Analog Input Select Register
	0x4000 07a6	ADC12A_0CFG	ADC12A Ch.0 Configuration Register
	0x4000 07a8	ADC12A_0INTF	ADC12A Ch.0 Interrupt Flag Register
	0x4000 07aa	ADC12A_0INTE	ADC12A Ch.0 Interrupt Enable Register
	0x4000 07ac	ADC12A_0DMAEN0	ADC12A Ch.0 DMA Request Enable Register 0
	0x4000 07ae	ADC12A_0DMAEN1	ADC12A Ch.0 DMA Request Enable Register 1
	0x4000 07b0	ADC12A_0DMAEN2	ADC12A Ch.0 DMA Request Enable Register 2
	0x4000 07b2	ADC12A_0DMAEN3	ADC12A Ch.0 DMA Request Enable Register 3
	0x4000 07b4	ADC12A_0DMAEN4	ADC12A Ch.0 DMA Request Enable Register 4
	0x4000 07b6	ADC12A_0DMAEN5	ADC12A Ch.0 DMA Request Enable Register 5
	0x4000 07b8	ADC12A_0DMAEN6	ADC12A Ch.0 DMA Request Enable Register 6
	0x4000 07ba	ADC12A_0DMAEN7	ADC12A Ch.0 DMA Request Enable Register 7
	0x4000 07bc	ADC12A_0ADD	ADC12A Ch.0 Result Register
Temperature sensor/reference voltage generator (TSRVR)	0x4000 07c0	TSRVR_0TCTL	TSRVR Ch.0 Temperature Sensor Control Register
	0x4000 07c2	TSRVR_0VCTL	TSRVR Ch.0 Reference Voltage Generator Control Register
LCD driver (LCD8D)	0x4000 0800	LCD8DCLK	LCD8D Clock Control Register
	0x4000 0802	LCD8DCTL	LCD8D Control Register
	0x4000 0804	LCD8DTIM1	LCD8D Timing Control Register 1
	0x4000 0806	LCD8DTIM2	LCD8D Timing Control Register 2
	0x4000 0808	LCD8DPWR	LCD8D Power Control Register
	0x4000 080a	LCD8DDSP	LCD8D Display Control Register
	0x4000 080c	LCD8DCOMC0	LCD8D COM Pin Control Register 0
	0x4000 0810	LCD8DINTF	LCD8D Interrupt Flag Register
	0x4000 0812	LCD8DINTE	LCD8D Interrupt Enable Register

Peripheral circuit	Address	Register name	
R/F converter (RFC) Ch.0	0x4000 0840	RFC_0CLK	RFC Ch.0 Clock Control Register
	0x4000 0842	RFC_0CTL	RFC Ch.0 Control Register
	0x4000 0844	RFC_0TRG	RFC Ch.0 Oscillation Trigger Register
	0x4000 0846	RFC_0MCL	RFC Ch.0 Measurement Counter Low Register
	0x4000 0848	RFC_0MCH	RFC Ch.0 Measurement Counter High Register
	0x4000 084a	RFC_0TCL	RFC Ch.0 Time Base Counter Low Register
	0x4000 084c	RFC_0TCH	RFC Ch.0 Time Base Counter High Register
	0x4000 084e	RFC_0INTF	RFC Ch.0 Interrupt Flag Register
	0x4000 0850	RFC_0INTE	RFC Ch.0 Interrupt Enable Register
DMA controller (DMAC)	0x4000 1000	DMACSTAT	DMAC Status Register
	0x4000 1004	DMACCFG	DMAC Configuration Register
	0x4000 1008	DMACCPTR	DMAC Control Data Base Pointer Register
	0x4000 100c	DMACACPTR	DMAC Alternate Control Data Base Pointer Register
	0x4000 1014	DMACSWREQ	DMAC Software Request Register
	0x4000 1020	DMACRMSET	DMAC Request Mask Set Register
	0x4000 1024	DMACRMCLR	DMAC Request Mask Clear Register
	0x4000 1028	DMACENSET	DMAC Enable Set Register
	0x4000 102c	DMACENCLR	DMAC Enable Clear Register
	0x4000 1030	DMACPASET	DMAC Primary-Alternate Set Register
	0x4000 1034	DMACPACLR	DMAC Primary-Alternate Clear Register
	0x4000 1038	DMACPRSET	DMAC Priority Set Register
	0x4000 103c	DMACPRCLR	DMAC Priority Clear Register
	0x4000 104c	DMACERRIF	DMAC Error Interrupt Flag Register
	0x4000 2000	DMACENDIF	DMAC Transfer Completion Interrupt Flag Register
	0x4000 2008	DMACENDIESET	DMAC Transfer Completion Interrupt Enable Set Register
	0x4000 200c	DMACENDIECLR	DMAC Transfer Completion Interrupt Enable Clear Register
	0x4000 2010	DMACERRIESET	DMAC Error Interrupt Enable Set Register
	0x4000 2014	DMACERRIECLR	DMAC Error Interrupt Enable Clear Register

4.6.1 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 SYSPROT.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 SYSPROT.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.

4.7 Control Registers

System Protect Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SYSPROT	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).

FLASHC Flash Read Cycle Register

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

Bits 15–2 Reserved

Bits 1–0 RDWAIT[1:0]

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

Table 4.7.1 Setting Number of Bus Access Cycles for Flash Read

FLASHCWAIT. RDWAIT[1:0] bits	Number of bus access cycles	System clock frequency	
		PWGACTL. REGSEL bit = 0	PWGACTL. REGSEL bit = 1
0x3	4	2.2 MHz (max.)	33 MHz (max.)
0x2	3		
0x1	2		
0x0	1	1.2 MHz (max.)	17.2 MHz (max.)

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

• When the FLASHCWAIT.RDWAIT[1:0] bit setting is altered from 0x2 to 0x1, add two NOP instructions immediately after that.
Program example:
```
FLASHC->WAIT_b.RDWAIT = 1;
asm("NOP");
asm("NOP");
CLG->OSC_b.IOSCEN = 0;
```

5 Interrupt

5.1 Overview

This IC includes a nested vectored interrupt controller (NVIC). For detailed information on the NVIC, refer to the documents introduced in Section 3.4, such as “ARM®v6-M Architecture Reference Manual.”
Figure 5.1.1 shows the configuration of the interrupt system.

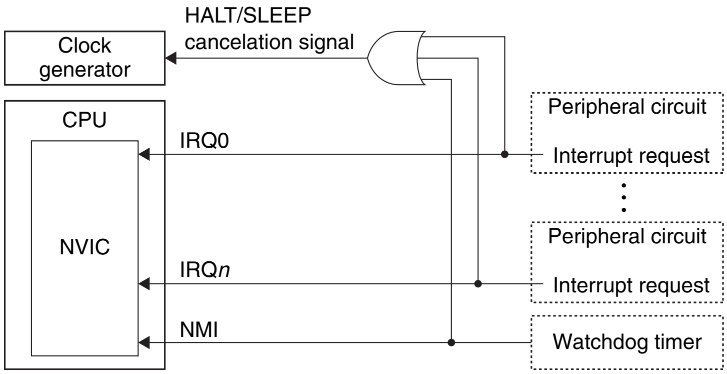


Figure 5.1.1 Configuration of Interrupt System

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

VTOR initial value = 0x0

Interrupt number	IRQ number	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
–	–	VTOR + 0x00	(Stack pointer initial value)	–	–
1	–	VTOR + 0x04	Reset	• Low input to the #RESET pin • Power-on reset • Key reset • Watchdog timer overflow *1 • Supply voltage detector reset	-3
2	-14	VTOR + 0x08	NMI	Watchdog timer overflow *1	-2
3	-13	VTOR + 0x0c	HardFault	• Bus error • Undefined instruction • Unaligned address etc.	-1
4–10	–	–	Reserved	–	–
11	-5	VTOR + 0x2c	SVCall	SVC instruction	Configurable
12–13	–	-	Reserved	–	–
14	-2	VTOR + 0x38	PendSV	–	Configurable
15	-1	VTOR + 0x3c	SysTick	SysTick timer underflow	
16	0	VTOR + 0x40	DMA controller interrupt	• DMA transfer completion • DMA transfer error	
17	1	VTOR + 0x44	Supply voltage detector Ch.0 interrupt	• Power supply voltage drop detection • Power supply voltage rise detection	
18	2	VTOR + 0x48	Port interrupt	Port input	
19	3	VTOR + 0x4c	Clock generator interrupt	• IOSC oscillation stabilization waiting completion • OSC1 oscillation stabilization waiting completion • OSC3 oscillation stabilization waiting completion • OSC1 oscillation stop • IOSC oscillation auto-trimming completion • IOSC oscillation auto-trimming error	

Interrupt number	IRQ number	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
20	4	VTOR + 0x50	Real-time clock interrupt	• 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	Configurable
21	5	VTOR + 0x54	16-bit timer Ch.0 interrupt	Underflow	
22	6	VTOR + 0x58	UART Ch.0 interrupt	• End of transmission • Framing error • Parity error • Overrun error • Receive buffer two bytes full • Receive buffer one byte full • Transmit buffer empty	
23	7	VTOR + 0x5c	16-bit timer Ch.1 interrupt	Underflow	
24	8	VTOR + 0x60	Synchronous serial interface Ch.0 interrupt	• End of transmission • Receive buffer full • Transmit buffer empty • Overrun error	
25	9	VTOR + 0x64	I^2C Ch.0 interrupt	• End of data transfer • General call address reception • NACK reception • STOP condition • START condition • Error detection • Receive buffer full • Transmit buffer empty	
26	10	VTOR + 0x68	16-bit PWM timer Ch.0 interrupt	• Capture overwrite • Compare/capture • Counter MAX • Counter zero	
27	11	VTOR + 0x6c	16-bit PWM timer Ch.1 interrupt	• Capture overwrite • Compare/capture • Counter MAX • Counter zero	
28	12	VTOR + 0x70	UART Ch.1 interrupt	• End of transmission • Framing error • Parity error • Overrun error • Receive buffer two bytes full • Receive buffer one byte full • Transmit buffer empty	
29	13	VTOR + 0x74	16-bit timer Ch.2 interrupt	Underflow	
30	–	–	Reserved	–	
31	15	VTOR + 0x7c	I^2C Ch.1 interrupt	• End of data transfer • General call address reception • NACK reception • STOP condition • START condition • Error detection • Receive buffer full • Transmit buffer empty	
32	16	VTOR + 0x80	IR remote controller interrupt	• Compare AP • Compare DB	
33	17	VTOR + 0x84	LCD driver interrupt	Frame	
34	18	VTOR + 0x88	16-bit timer Ch.3 interrupt	Underflow	
35	19	VTOR + 0x8c	16-bit PWM timer Ch.2 interrupt	• Capture overwrite • Compare/capture • Counter MAX • Counter zero	
36	20	VTOR + 0x90	Synchronous serial interface Ch.1 interrupt	• End of transmission • Receive buffer full • Transmit buffer empty • Overrun error	
37	21	VTOR + 0x94	Sound generator interrupt	• Sound buffer empty • Sound output completion	

Interrupt number	IRQ number	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
38	22	VTOR + 0x98	R/F converter Ch.0 interrupt	• Reference oscillation completion • Sensor A oscillation completion • Sensor B oscillation completion • Measurement counter overflow error • Time base counter overflow error	Configurable
39	23	VTOR + 0x9c	16-bit timer Ch.4 interrupt	Underflow	
40	24	VTOR + 0xa0	16-bit timer Ch.5 interrupt	Underflow	
41	25	VTOR + 0xa4	16-bit timer Ch.6 interrupt	Underflow	
42	26	VTOR + 0xa8	16-bit timer Ch.7 interrupt	Underflow	
43	27	VTOR + 0xac	12-bit A/D converter interrupt	• Analog input signal *m* A/D conversion completion • Analog input signal *m* A/D conversion result overwrite error	
44–47	–	–	Reserved	–	

∗1 Either reset or NMI can be selected as the watchdog timer interrupt via software.

5.2.1 Vector Table Offset Address (VTOR)

The Cortex®-M0+ Vector Table Offset Register (VTOR) is provided to set the offset (start) address of the vector table in which interrupt vectors are programmed. "VTOR" described in Table 5.2.1 means the value set to this register. After an initial reset, VTOR is set to address 0x0. Therefore, even when the vector table location is changed, it is necessary that at least the reset vector be written to this address. For more information on VTOR, refer to the documents introduced in Section 3.4, such as "Cortex®-M0+ Devices Generic User Guide."

5.2.2 Priority of Interrupts

The priorities of SVCall, PendSV, and SysTick are configurable to the desired levels using the Cortex®-M0+ System Handler Priority Registers (SHPR2 and SHPR3). The priorities of the interrupt number 16 or later are configurable to the desired levels using the Cortex®-M0+ Interrupt Priority Registers (NVIC_IPR0–7). The priority value can be set within a range of 0 to 192 (a lower value has a higher priority). The priorities of reset, NMI, and HardFault are fixed at the predefined values. For more information, refer to the documents introduced in Section 3.4, such as "Cortex®-M0+ Devices Generic User Guide."

5.3 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 CPU 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 CPU even if the interrupt flag is set to 1. An interrupt request is also sent to the CPU 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 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.

6 DMA Controller (DMAC)

6.1 Overview

The main features of the DMAC are outlined below.

- Supports byte, halfword, and word transfers.
- Each DMAC channel can be configured to different transfer conditions independently.
- Supports memory-to-memory, memory-to-peripheral circuit, and peripheral circuit-to-memory transfers.
- Supports hardware DMA requests from peripheral circuits and software DMA requests.
- Priority level for each channel is selectable from two levels.
- DMA transfers are allowed even if the CPU is placed into HALT mode.

Figure 6.1.1 shows the configuration of the DMAC.

Table 6.1.1 DMAC Channel Configuration of S1C31W65

Item	S1C31W65
Number of channels	4 channels (Ch.0 to Ch.3)
Transfer source memories	Internal Flash memory, external Flash memory, RAM, and display data RAM
Transfer destination memories	RAM and display data RAM
Transfer source peripheral circuits	UART3, SPIA, I2C, T16B, and ADC12A
Transfer destination peripheral circuits	UART3, SPIA, I2C, T16B, and SNDA

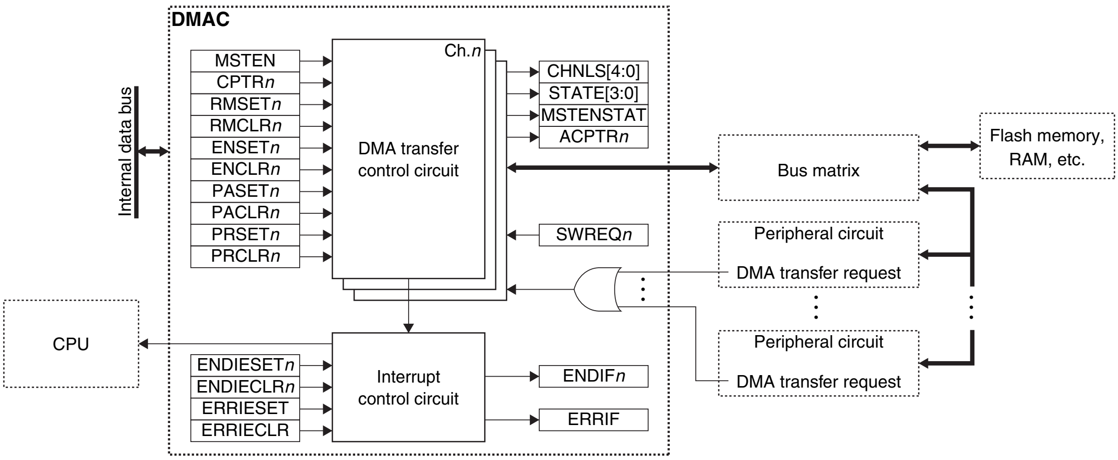


Figure 6.1.1 DMAC Configuration

6.2 Operations

6.2.1 Initialization

The DMAC should be initialized with the procedure shown below.

1. Set the data structure base address to the DMACCPTR register.
2. Configure the data structure for the channels to be used.
   - Set the control data.
   - Set the transfer source end pointer.
   - Set the transfer destination end pointer.
3. Set the DMACCFG.MSTEN bit to 1. (Enable DMAC)
4. Configure the DMACRMSET and DMACRMCLR registers. (Configure masks for DMA transfer requests from peripheral circuits)
5. Configure the DMACENSET and DMACENCLR registers. (Enable channels used)
6. Configure the DMACPASET and DMACPACLR registers. (Select data structure used)
7. Configure the DMACPRSET and DMACPRCLR registers. (Set priorities)
8. Set the following registers when using the interrupt:
   - Write 1 to the interrupt flags in the DMACENDIF and DMACERRIF registers. (Clear interrupt flags)
   - Configures the DMACENDIESET/DMACENDIECLR and DMACERRIESET/DMACERRIECLR registers. (Enable/disable interrupts)
9. Set the DMA request enable bits of the peripheral circuits that use DMA transfer to 1.
10. To issue a software DMA request to Ch.*n*, write 1 to the DMACSWREQ.SWREQ*n* bit.

6.3 Priority

If DMA requests are issued to two or more channels, the DMA transfers are performed in order from the highest-priority channel. The channel of which the priority level is set to 1 by the DMACPRSET.PRSET*n* bit has the highest priority. If two or more channels have been set to the same priority level, the smaller channel number takes precedence.

6.4 Data Structure

To perform DMA transfers, a data structure that contains basic transfer control information must be provided. The data structure consists of two blocks, primary data structure and alternate data structure, and one of them is used according to the DMA transfer mode.
The data structure can be located at an arbitrary address in the RAM area by setting the base address to the DMACCPTR.CPTR[31:0] bits.
The data structure for each channel consists of a transfer source end pointer, a transfer destination end pointer, and control data. An area of 16 bytes × 2 is allocated in the RAM for each channel.
The whole size of the data structure and the alternate data structure base address depend on the number of channels implemented.

Table 6.4.1 Data Structure Size According to Number of Channels Implemented

Number of channels implemented	Data structure size	Primary data structure base address	Alternate data structure base address
1	32 bytes	DMACCPTR.CPTR[31:0] (CPTR[4:0] = 0x00)	DMACCPTR.CPTR[31:0] + 0x010
2	64 bytes	DMACCPTR.CPTR[31:0] (CPTR[5:0] = 0x00)	DMACCPTR.CPTR[31:0] + 0x020
3 to 4	128 bytes	DMACCPTR.CPTR[31:0] (CPTR[6:0] = 0x00)	DMACCPTR.CPTR[31:0] + 0x040
5 to 8	256 bytes	DMACCPTR.CPTR[31:0] (CPTR[7:0] = 0x00)	DMACCPTR.CPTR[31:0] + 0x080
9 to 16	512 bytes	DMACCPTR.CPTR[31:0] (CPTR[8:0] = 0x000)	DMACCPTR.CPTR[31:0] + 0x100
17 to 32	1,024 bytes	DMACCPTR.CPTR[31:0] (CPTR[9:0] = 0x000)	DMACCPTR.CPTR[31:0] + 0x200

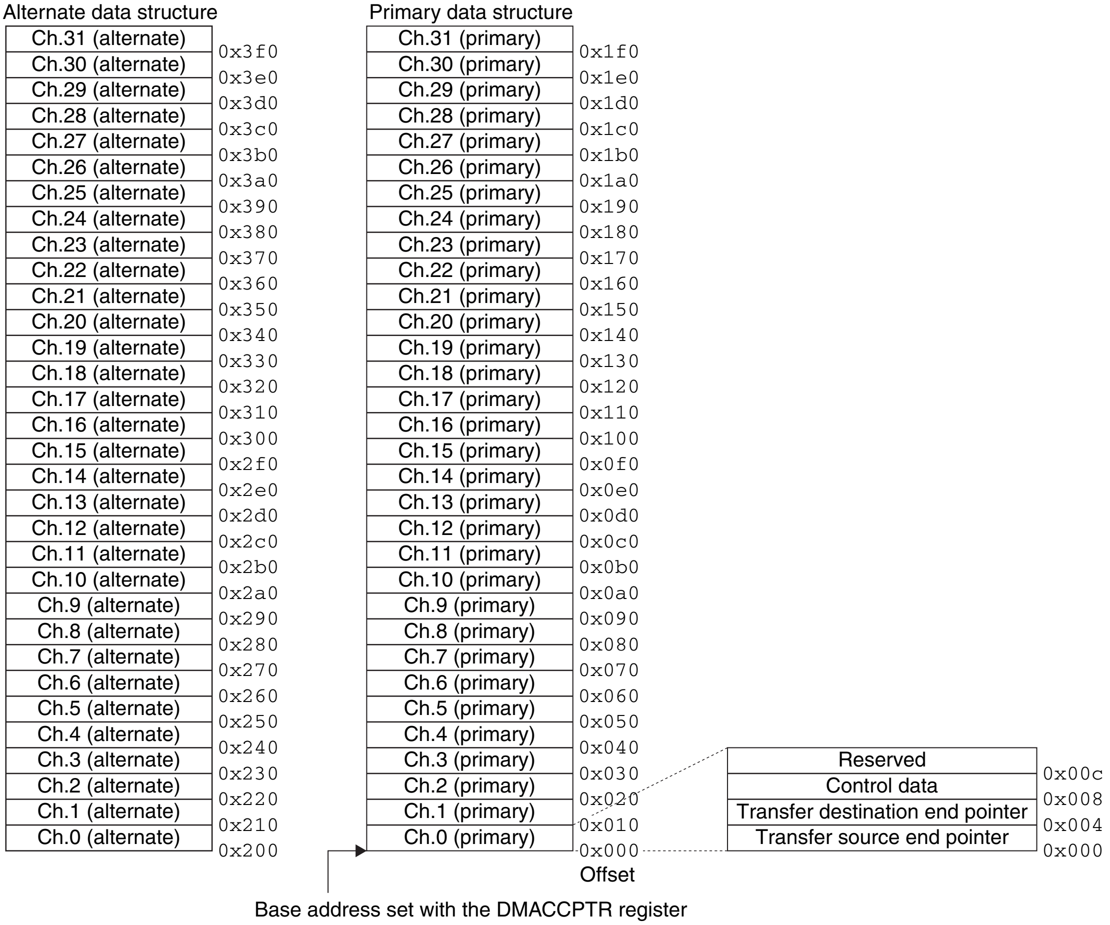


Figure 6.4.1 Data Structure Address Map (when 32 channels are implemented)

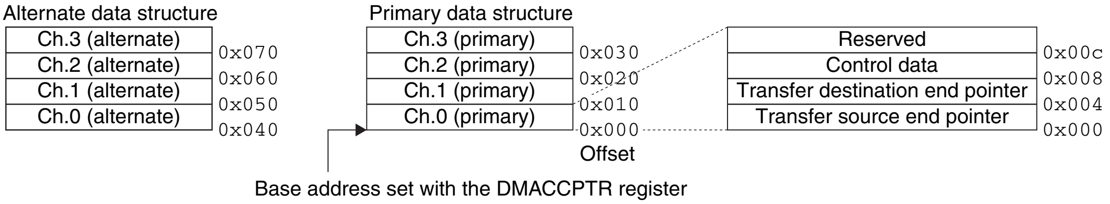


Figure 6.4.2 Data Structure Address Map (when 4 channels are implemented)

The alternate data structure base address can be determined from the DMACACPTR.ACPTR[31:0] bits.

6.4.1 Transfer Source End Pointer

Set the source data end address. The address of data to be transferred should be set as it is if the transfer source address is not incremented.

6.4.2 Transfer Destination End Pointer

Set the address to which the last transfer data is written. The address for writing transfer data should be set as it is if the transfer destination address is not incremented.

6.4.3 Control Data

Set the DMA transfer information. Figure 6.4.3.1 shows the constituent elements of the control data.

31–30	29–28	27–26	25–24	23–18	17–14	13–4	3	2–0
dst_inc	dst_size	src_inc	src_size	(Reserved)	R_power	n_minus_1	(Reserved)	cycle_ctrl

Reserved

Figure 6.4.3.1 Constituent Elements of Control Data

dst_inc

Set the increment value of the transfer destination address. The setting value must be equal to or larger than the transfer data size when the address is incremented.

Table 6.4.3.1 Increment Value of Transfer Destination Address

dst_inc	Increment value
0x3	No increment
0x2	+4
0x1	+2
0x0	+1

dst_size

Set the size of the data to be written to the transfer destination. It should be the same value as the src_size.

Table 6.4.3.2 Size of Data Written to Transfer Destination

dst_size	Data size
0x3	Reserved
0x2	Word
0x1	Halfword
0x0	Byte

src_inc

Set the increment value of the transfer source address. The setting value must be equal to or larger than the transfer data size when the address is incremented.

Table 6.4.3.3 Increment Value of Transfer Source Address

src_inc	Increment value
0x3	No increment
0x2	+4
0x1	+2
0x0	+1

src_size

Set the size of the data to be read from the transfer source. It should be the same value as the dst_size.

Table 6.4.3.4 Size of Data Read from Transfer Source

src_size	Data size
0x3	Reserved
0x2	Word
0x1	Halfword
0x0	Byte

R_power

Set the arbitration cycle during successive data transfer.

Arbitration cycle $(2^R) = 2^{R\_power}$

When the DMAC is performing a successive transfer, it suspends the data transfer at the cycle set with R_power. If DMA requests have been issued at that point, the DMAC re-arbitrates them according to their priorities and then performs a DMA transfer for the channel with the highest priority.

If the arbitration cycle setting value is larger than the number of successive data transfers, successive data transfers will not be suspended.

n_minus_1

Set the number of DMA transfers to be executed successively.

Number of successive transfers (N) = n_minus_1 + 1

When the set number of successive transfers has completed, a transfer completion interrupt occurs.

cycle_ctrl

Set the DMA transfer mode. For detailed information on each transfer mode, refer to Section 6.5, "DMA Transfer Mode."

Table 6.4.3.5 DMA Transfer Mode

cycle_ctrl	DMA transfer mode
0x7	Peripheral scatter-gather transfer (for alternate data structure)
0x6	Peripheral scatter-gather transfer (for primary data structure)
0x5	Memory scatter-gather transfer (for alternate data structure)
0x4	Memory scatter-gather transfer (for primary data structure)
0x3	Ping-pong transfer
0x2	Auto-request transfer
0x1	Basic transfer
0x0	Stop

6.5 DMA Transfer Mode

6.5.1 Basic Transfer

This is the basic DMA transfer mode. In this mode, DMA transfer starts when a DMA transfer request from a peripheral circuit or a software DMA request is issued, and it continues until it is completed for the set number of successive transfers or it is suspended at the arbitration cycle. To resume the DMA transfer suspended at the arbitration cycle, a DMA transfer request must be reissued.
When the set number of successive transfers has completed, a transfer completion interrupt occurs.

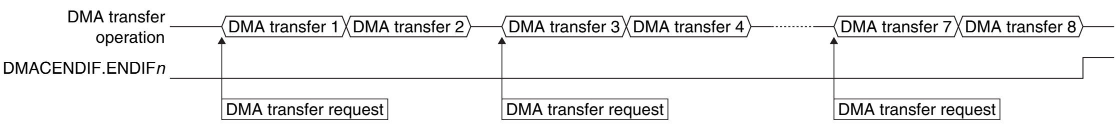


Figure 6.5.1.1 Basic Transfer Operation Example (N = 8, 2^R = 2)

6.5.2 Auto-Request Transfer

Similar to the basic transfer, DMA transfer starts when a DMA transfer request from a peripheral circuit or a software DMA request is issued, and it continues until it is completed for the set number of successive transfers or it is suspended at the arbitration cycle. The DMAC resumes the DMA transfer suspended at the arbitration cycle without a DMA transfer request being reissued.
When the set number of successive transfers has completed, a transfer completion interrupt occurs.

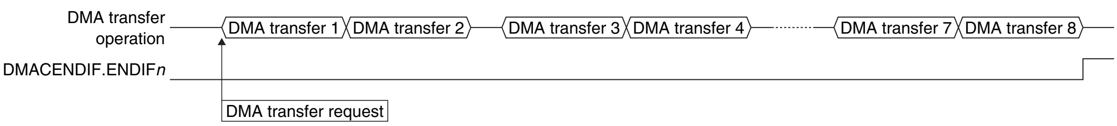


Figure 6.5.2.1 Auto-Request Transfer Operation Example (N = 8, 2^R = 2)

6.5.3 Ping-Pong Transfer

In ping-pong transfer mode, the DMAC performs basic transfers repeatedly while switching between the primary data structure and alternate data structure. The data structures are referred alternately, and DMA transfer is terminated when the control data with cycle_ctrl set to 0x0 is referred. A transfer completion interrupt occurs each time a transfer using a data structure is completed.

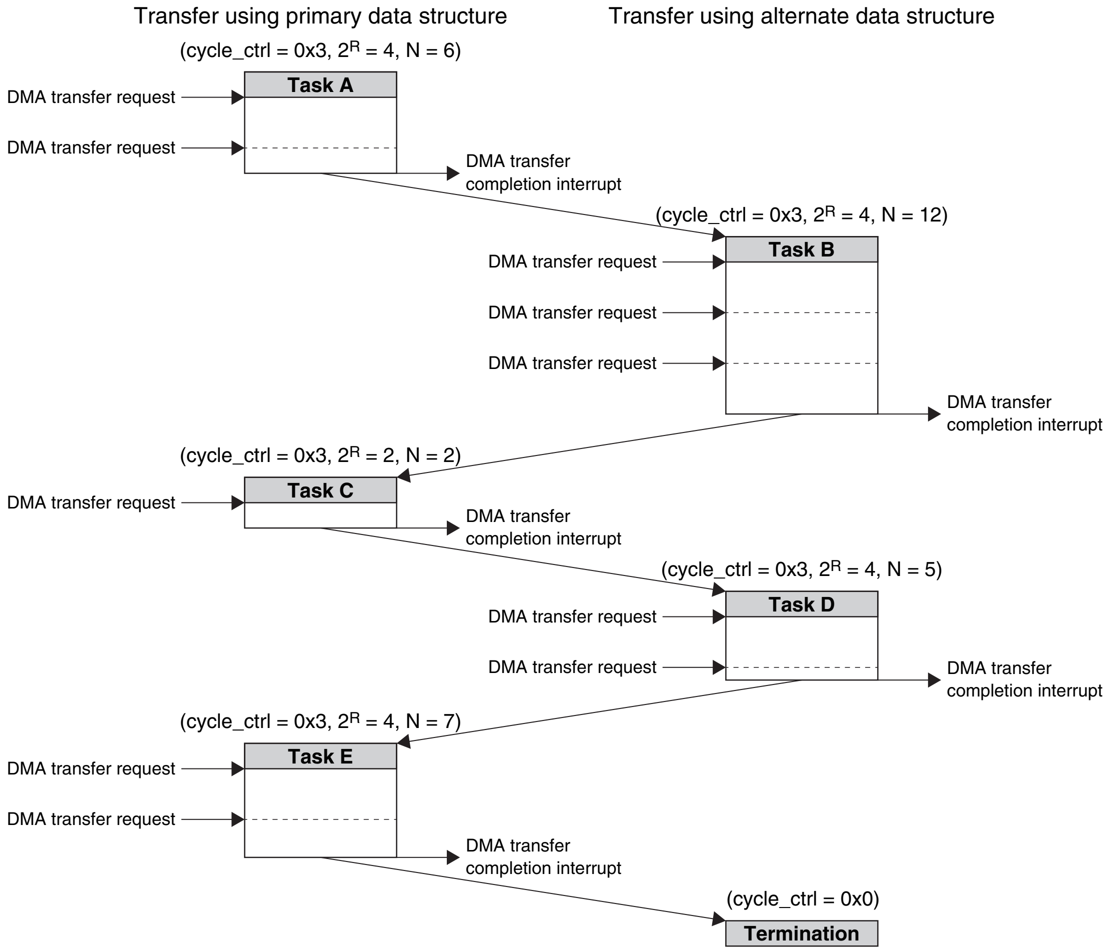


Figure 6.5.3.1 Ping-Pong Transfer Operation Example

DMA transfer procedure

1. Start data transfer by following the procedure shown in Section 6.2.1, “Initialization.” In Step 2 of the initialization procedure, set Task A and Task B to the primary data structure and the alternate data structure, respectively.
2. Set Task C to the primary data structure after a DMA transfer completion interrupt has occurred by Task A.
3. Set Task D to the alternate data structure when a DMA transfer completion interrupt has occurred by Task B.
4. Repeat Steps 2 and 3.
5. Set cycle_ctrl to 0x0 after a DMA transfer completion interrupt has occurred by the next to last task.
6. The DMA transfer is completed when a DMA transfer completion interrupt occurs by the last task.

6.5.4 Memory Scatter-Gather Transfer

In scatter-gather transfer mode, first the DMAC, using the primary data structure, copies a data structure from the data structure table, which has been prepared with multiple data structures included in advance, to the alternate data structure, and then it performs DMA transfer using the alternate data structure. The DMAC performs this operation repeatedly. By programming the transfer mode of the data structure located at the end of the table as a basic transfer, the DMA transfer can be terminated with a transfer completion interrupt. This mode requires a DMA transfer request only for starting the first data transfer. Subsequent data transfers are performed by auto-requests.

Reserved	Data structure for Task D
Control data	
Transfer destination end pointer	
Transfer source end pointer	
Reserved	Data structure for Task C
Control data	
Transfer destination end pointer	
Transfer source end pointer	
Reserved	Data structure for Task B
Control data	
Transfer destination end pointer	
Transfer source end pointer	
Reserved	Data structure for Task A
Control data	
Transfer destination end pointer	
Transfer source end pointer	

Figure 6.5.4.1 Example of Data Structure Table for Scatter-Gather Transfer

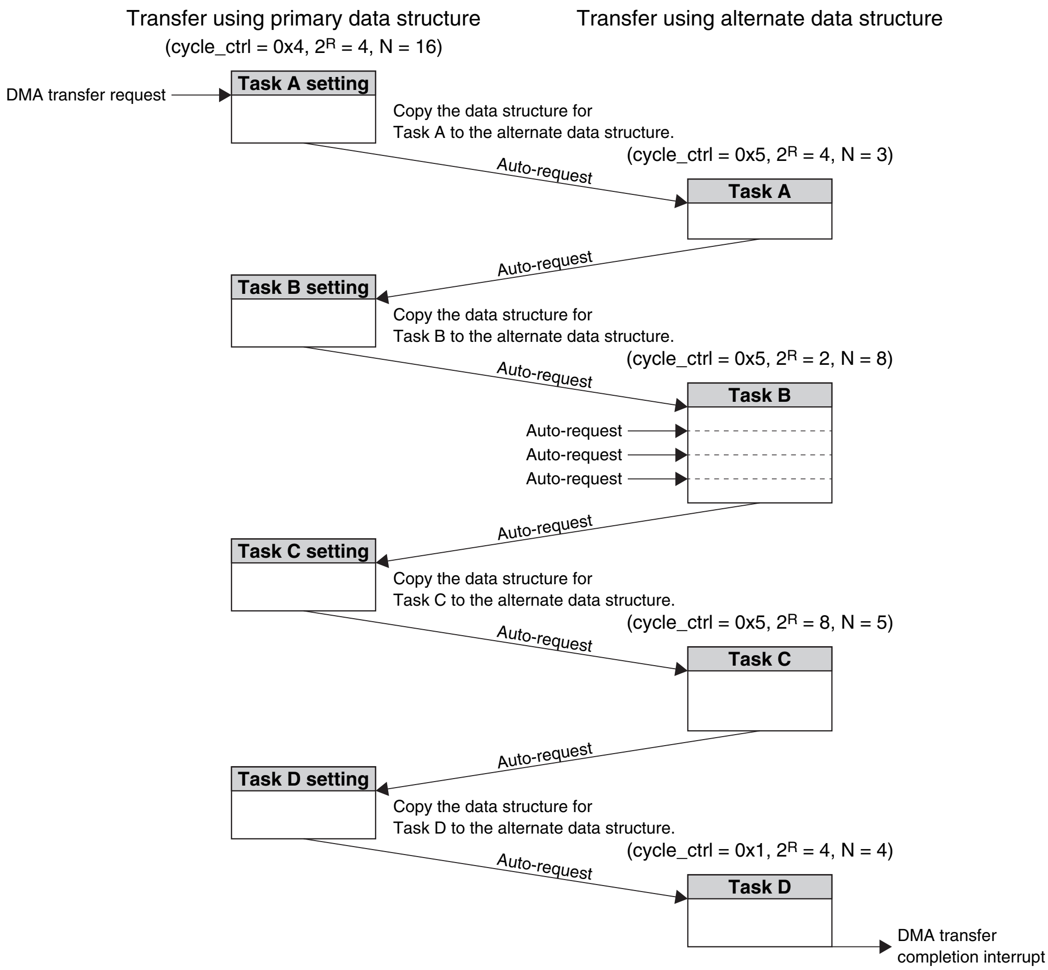


Figure 6.5.4.2 Memory Scatter-Gather Transfer Operation Example

DMA transfer procedure

1. Configure the data structure table for scatter-gather transfer.
   Set the cycle_ctrl for the last task to 0x1 and those for other tasks to 0x5.
2. Start data transfer by following the procedure shown in Section 6.2.1, "Initialization." In Step 2 of the initialization procedure, configure the primary data structure with the control data shown below.

   Transfer source end pointer = Data structure table end address
   Transfer destination end pointer = Alternate data structure end address
   dst_inc = 0x2
   dst_size = 0x2
   src_inc = 0x2
   src_size = 0x2
   R_power = 0x2
   n_minus_1 = Number of tasks × 4 - 1
   cycle_ctrl = 0x4
3. The DMA transfer is completed when a DMA transfer completion interrupt occurs.

6.5.5 Peripheral Scatter-Gather Transfer

In memory scatter-gather transfer mode, the second and subsequent DMA transfers are performed by auto-requests. On the other hand, in peripheral scatter-gather transfer mode, all DMA transfers are performed by a DMA transfer request issued by a peripheral circuit or a software DMA request.

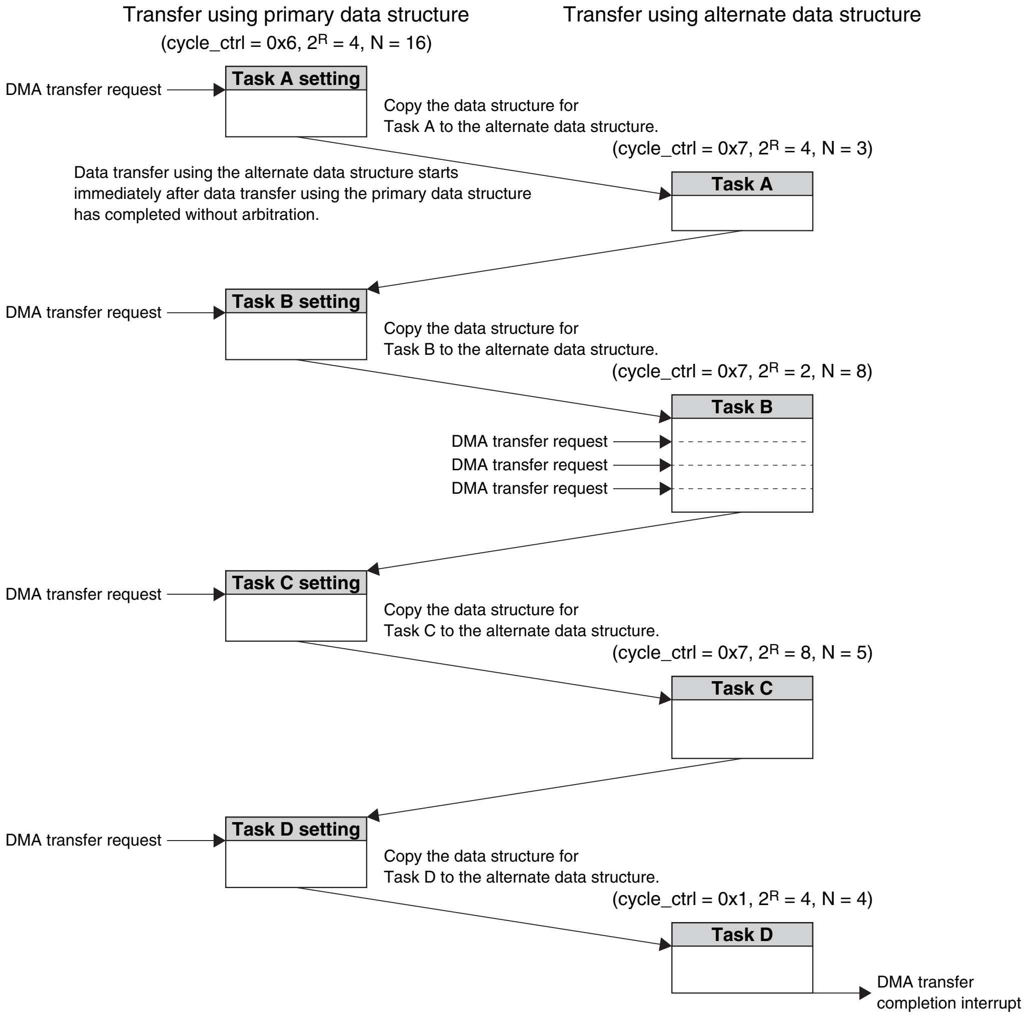


Figure 6.5.5.1 Peripheral Scatter-Gather Transfer Operation Example

DMA transfer procedure

1. Configure the data structure table for scatter-gather transfer.
   Set the cycle_ctrl for the last task to 0x1 and those for other tasks to 0x7.
2. Start data transfer by following the procedure shown in Section 6.2.1, "Initialization." In Step 2 of the initialization procedure, configure the primary data structure with the control data shown below.
   Transfer source end pointer = Data structure table end address
   Transfer destination end pointer = Alternate data structure end address
   dst_inc = 0x2
   dst_size = 0x2
   src_inc = 0x2
   src_size = 0x2
   R_power = 0x2
   n_minus_1 = Number of tasks × 4 - 1
   cycle_ctrl = 0x6
3. Issue a DMA transfer request in each task using a peripheral circuit or via software.
4. The DMA transfer is completed when a DMA transfer completion interrupt occurs.

6.6 DMA Transfer Cycle

A DMA transfer requires several clock cycles to execute. Figure 6.6.1 shows a detailed DAM transfer cycle. Note that the number of clock cycles for a DMA transfer may be increased due to a conflict with an access from the CPU or the Flash bus access cycle setting.

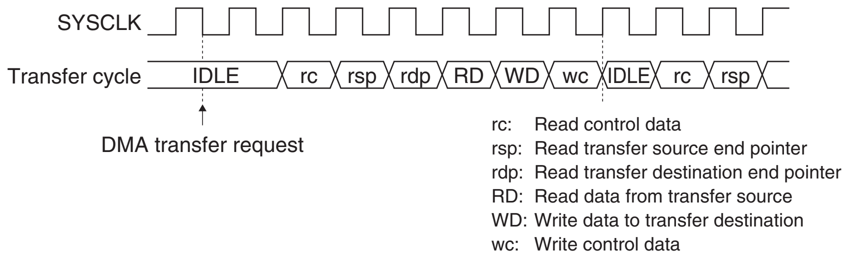


Figure 6.6.1 DMA Transfer Cycle

6.7 Interrupts

The DMAC has a function to generate the interrupts shown in Table 6.7.1.

Table 6.7.1 DMAC Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
DMA transfer completion	DMACENDIF.ENDIF*n*	When DMA transfers for a set number of successive transfers have completed	Writing 1
DMA transfer error	DMACERRIF.ERRIF	When an AHB bus error has occurred	Writing 1

The DMAC provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the CPU 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" chapter.

6.8 Control Registers

DMAC Status Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACSTAT	31–24	–	0x00	–	R	–
	23–21	–	0x0	–	R	
	20–16	CHNLS[4:0]	*	H0	R	* Number of channels implemented - 1
	15–8	–	0x00	–	R	–
	7–4	STATE[3:0]	0x0	H0	R	
	3–1	–	0x0	–	R	
	0	MSTENSTAT	0	H0	R	

Bits 31–21 Reserved

Bits 20–16 CHNLS[4:0]

These bits show the number of DMAC channels implemented in this IC.

Number of channels implemented = CHNLS + 1

Bits 15–8 Reserved

Bits 7–4 STATE[3:0]

These bits indicates the DMA transfer status.

Table 6.8.1 DMA Transfer Status

DMACSTAT.STATE[3:0] bits	DMA transfer status
0xf–0xbf	Reserved
0xa	Peripheral scatter-gather transfer is in progress.
0x9	Transfer has completed.
0x8	Transfer has been suspended.
0x7	Control data is being written.
0x6	Standby for transfer request to be cleared.
0x5	Transfer data is being written.
0x4	Transfer data is being read.
0x3	Transfer destination end pointer is being read.
0x2	Transfer source end pointer is being read.
0x1	Control data is being read.
0x0	Idle

Bits 3–1 Reserved

Bit 0 MSTENSTAT

This bit indicates the DMA controller status.

1 (R): DMA controller is operating.

0 (R): DMA controller is idle.

DMAC Configuration Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACCFG	31–24	–	0x00	–	R	–
	23–16	–	0x00	–	R	
	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	MSTEN	–	–	W	

Bits 31–1 Reserved

Bit 0 MSTEN

This bit enables the DMA controller.

1 (W): Enable

0 (W): Disable

DMAC Control Data Base Pointer Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACCPTR	31–0	CPTR[31:0]	0x0000 0000	H0	R/W	–

Bits 31–0 CPTR[31:0]

These bits set the leading address of the data structure.

Depending on the number of channels implemented, low-order bits are configured for read only.

Table 6.8.2 CPTR Writable/Read-Only Bits Depending On Number of Channel Implemented

Number of channel implemented	Writable bits	Read-only bits
1	CPTR[31:5]	CPTR[4:0]
2	CPTR[31:6]	CPTR[5:0]
3–4	CPTR[31:7]	CPTR[6:0]
5–8	CPTR[31:8]	CPTR[7:0]
9–16	CPTR[31:9]	CPTR[8:0]
17–32	CPTR[31:10]	CPTR[9:0]

DMAC Alternate Control Data Base Pointer Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACACPTR	31–0	ACPTR[31:0]	–	H0	R	–

Bits 31–0 ACPTR[31:0]

These bits show the alternate data structure base address.

DMAC Software Request Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACSWREQ	31–0	SWREQ[31:0]	–	–	W	–

Bits 31–0 SWREQ [31:0]

These bits issue a software DMA transfer request to each channel.

1 (W): Issue a software DMA transfer request

0 (W): Ineffective

Each bit corresponds to a DMAC channel (e.g. bit *n* corresponds to Ch.*n*). The high-order bits for the unimplemented channels are ineffective.

DMAC Request Mask Set Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACRMSET	31–0	RMSET[31:0]	0x0000 0000	H0	R/W	–

Bits 31–0 RMSET[31:0]

These bits mask DMA transfer requests from peripheral circuits.

1 (W): Mask DMA transfer requests from peripheral circuits

0 (W): Ineffective

1 (R): DMA transfer requests from peripheral circuits have been disabled.

0 (R): DMA transfer requests from peripheral circuits have been enabled.

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Request Mask Clear Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACRMCLR	31–0	RMCLR[31:0]	–	–	W	–

Bits 31–0 RMCLR[31:0]

These bits cancel the mask state of DMA transfer requests from peripheral circuits

1 (W): Cancel mask state of DMA transfer requests from peripheral circuits (The DMACRMSET register is cleared to 0.)

0 (W): Ineffective

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Enable Set Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACENSET	31–0	ENSET[31:0]	0x0000 0000	H0	R/W	–

Bits 31–0 ENSET[31:0]

These bits enable each DMAC channel.

1 (W): Enable DMAC channel

0 (W): Ineffective

1 (R): Enabled

0 (R): Disabled

These bits are cleared after the DMA transfer has completed.

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Enable Clear Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACENCLR	31–0	ENCLR[31:0]	–	–	W	–

Bits 31–0 ENCLR[31:0]

These bits disable each DMAC channel.

1 (W): Disable DMAC channel (The DMACENSET register is cleared to 0.)

0 (W): Ineffective

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Primary-Alternate Set Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACPASET	31–0	PASET[31:0]	0x0000 0000	H0	R/W	–

Bits 31–0 PASET[31:0]

These bits enable the alternate data structures.

1 (W): Enable alternate data structure

0 (W): Ineffective

1 (R): The alternate data structure has been enabled.

0 (R): The primary data structure has been enabled.

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Primary-Alternate Clear Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACPACLR	31–0	PACLR[31:0]	–	–	W	–

Bits 31–0 PACLR[31:0]

These bits disable the alternate data structures.

1 (W): Disable alternate data structure (The DMACPASET register is cleared to 0.)

0 (W): Ineffective

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Priority Set Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACPRSET	31–0	PRSET[31:0]	0x0000 0000	H0	R/W	–

Bits 31–0 PRSET[31:0]

These bits increase the priority of each channel.

1 (W): Increase priority

0 (W): Ineffective

1 (R): Priority = High

0 (R): Priority = Normal

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Priority Clear Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACPRCLR	31–0	PRCLR[31:0]	–	–	W	–

Bits 31–0 PRCLR[31:0]

These bits decrease the priority of each channel.

1(W): Decrease priority (The DMACPRSET register is cleared to 0.)

0 (W): Ineffective

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Error Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACERRIF	31–24	–	0x00	–	R	–
	23–16	–	0x00	–	R	
	15–8	–	0x00	–	R	
	7–1	–	0x00	–	R	
	0	ERRIF	0	H0	R/W	Cleared by writing 1.

Bits 31–1 Reserved

Bit 0 ERRIF

This bit indicates the DMAC error interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

DMAC Transfer Completion Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACENDIF	31–0	ENDIF[31:0]	0x0000 0000	H0	R/W	Cleared by writing 1.

Bits 31–0 ENDIF[31:0]

These bits indicate the DMA transfer completion interrupt cause occurrence status of each DMAC channel.

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

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Transfer Completion Interrupt Enable Set Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACENDIESET	31–0	ENDIESET[31:0]	0x0000 0000	H0	R/W	–

Bits 31–0 ENDIESET[31:0]

These bits enable DMA transfer completion interrupts to be generated from each DMAC channel.

1 (W): Enable interrupt
0 (W): Ineffective
1 (R): Interrupt has been enabled.
0 (R): Interrupt has been disabled.

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Transfer Completion Interrupt Enable Clear Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACENDIECLR	31–0	ENDIECLR[31:0]	–	–	W	–

Bits 31–0 ENDIECLR[31:0]

These bits disable DMA transfer completion interrupts to be generated from each DMAC channel.

1 (W): Disable interrupt (The DMACENDIESET register is cleared to 0.)
0 (W): Ineffective

Each bit corresponds to a DMAC channel. The high-order bits for the unimplemented channels are ineffective.

DMAC Error Interrupt Enable Set Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACERRIESET	31–24	–	0x00	–	R	–
	23–16	–	0x00	–	R	
	15–8	–	0x00	–	R	
	7–1	–	0x00	–	R	
	0	ERRIESET	0	H0	R/W	

Bits 31–1 Reserved

Bit 0 ERRIESET

This bit enables DMA error interrupts.

1 (W): Enable interrupt
0 (W): Ineffective
1 (R): Interrupt has been enabled.
0 (R): Interrupt has been disabled.

DMAC Error Interrupt Enable Clear Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
DMACERRIECLR	31–24	–	0x00	–	R	–
	23–16	–	0x00	–	R	
	15–8	–	0x00	–	R	
	7–1	–	0x00	–	R	
	0	ERRIECLR	–	–	W	

Bits 31–1 Reserved

Bit 0 ERRIECLR

This bit disables DMA error interrupts.

1 (W): Disable interrupt (The DMACERRIESET register is cleared to 0.)
0 (W): Ineffective

7 I/O Ports (PPORT)

7.1 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 P*xy*, 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 7.1.1 shows the configuration of PPORT.

Table 7.1.1 Port Configuration of S1C31W65

Item	S1C31W65
Port groups included	P0[7:0], P1[7:0], P2[7:0], P3[7:0], P4[7:0], P5[7:0], P6[7:0], Pd[7:0]
Ports with general-purpose I/O function (GPIO)	P0[7:0], P1[7:0], P2[7:0], P3[7:0], P4[7:0], P5[7:0], P6[7:0], Pd[7:0] (Pd4: output only)
Ports with interrupt function	P0[7:0], P1[7:0], P2[7:0], P3[7:0], P4[7:0], P5[7:0], P6[7:0]
Ports for debug function	Pd[1:0]
Key-entry reset function	Supported (P0[3:0])

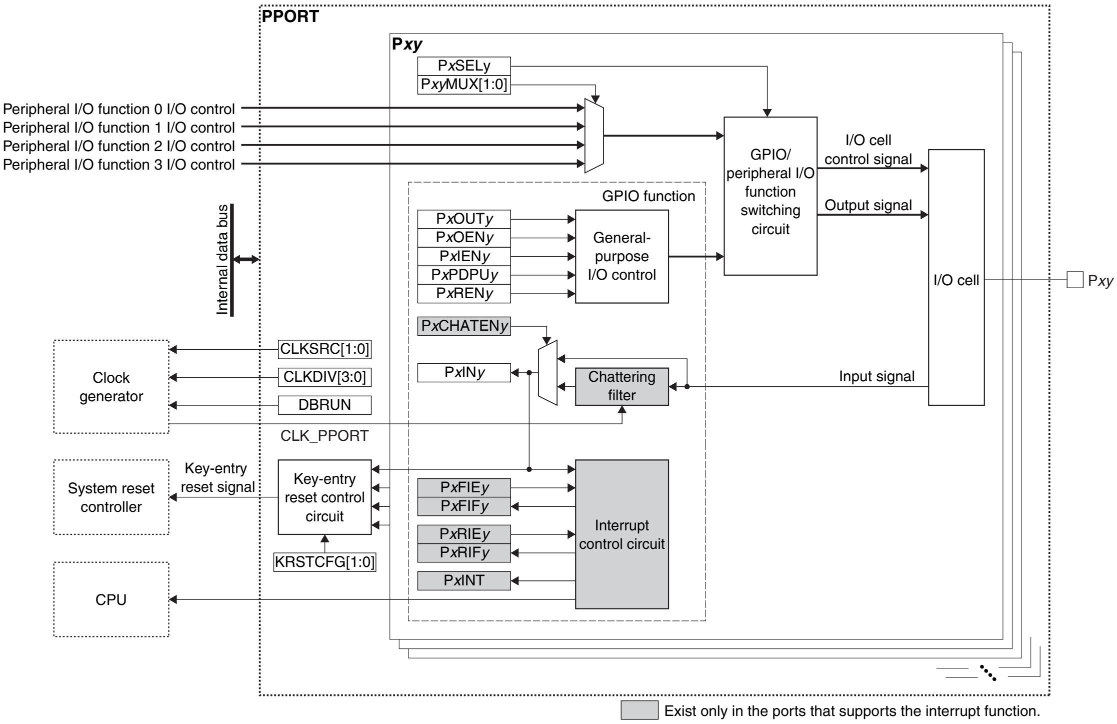


Figure 7.1.1 PPORT Configuration

7.2 I/O Cell Structure and Functions

Figure 7.2.1 shows the I/O cell Configuration.

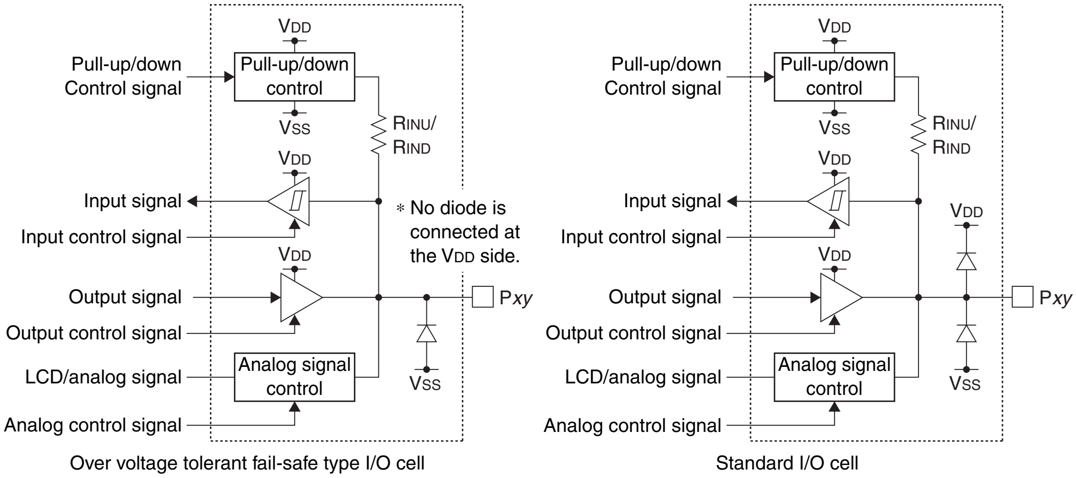


Figure 7.2.1 I/O Cell Configuration

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.

7.2.1 Schmitt Input

The input functions are all configured with the Schmitt interface level. When a port is set to input disable status (PPORTP*x*IOEN.P*x*IEN*y* bit = 0), unnecessary current is not consumed if the P*xy* pin is placed into floating status.

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

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

7.2.3 Pull-Up/Pull-Down

The GPIO port has a pull-up/pull-down function. Either pull-up or pull-down may be selected for each port individually. This function may also be disabled for the port that does not require pulling up/down.
When the port level is switched from low to high through the pull-up resistor included in the I/O cell or from high to low through the pull-down resistor, a delay will occur in the waveform rising/falling edge depending on the time constant by the pull-up/pull-down resistance and the pin load capacitance. The rising/falling time is commonly determined by the following equation:

$$t_{PR} = -R_{INU} \times (C_{IN} + C_{BOARD}) \times \ln(1 - V_{T+}/V_{DD}) \quad \text{(Eq. 7.1)}$$
$$t_{PF} = -R_{IND} \times (C_{IN} + C_{BOARD}) \times \ln(1 - V_{T-}/V_{DD})$$

Where

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

7.2.4 CMOS Output and High Impedance State

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

7.3 Clock Settings

7.3.1 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 SYSPROT.PROT[15:0] bits. (Remove system protection)
3. Set the following PPORTCLK register bits:
   - PPORTCLK.CLKSRC[1:0] bits (Clock source selection)
   - PPORTCLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

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

7.3.2 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 suppling by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_PPORT clock source.
If the CLGOSC.*xxxx*SLPC 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 PPORTP*x*CHATEN.P*x*CHATEN*y* bit setting (chattering filter enabled/disabled).

7.3.3 Clock Supply During Debugging

The CLK_PPORT supply during debugging should be controlled using the PPORTCLK.DBRUN bit.
The CLK_PPORT supply to PPORT is suspended when the CPU enters debug state if the PPORTCLK.DBRUN bit = 0. After the CPU returns to normal operation, 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 PPORTCLK.DBRUN bit = 1, the CLK_PPORT supply is not suspended and the chattering filter will keep operating in a debug state.

7.4 Operations

7.4.1 Initialization

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

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

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

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

When using the P*xy* port for a peripheral I/O function, perform the following software initial settings:

1. Set the following PPORTP*x*IOEN register bits:
   - Set the PPORTP*x*IOEN.P*x*IEN*y* bit to 0. (Disable input)
   - Set the PPORTP*x*IOEN.P*x*OEN*y* bit to 0. (Disable output)
2. Set the PPORTP*x*MODSEL.P*x*SEL*y* bit to 0. (Disable peripheral I/O function)
3. Initialize the peripheral circuit that uses the pin.
4. Set the PPORTP*x*FNCSEL.P*xy*MUX[1:0] bits. (Select peripheral I/O function)
5. Set the PPORTP*x*MODSEL.P*x*SEL*y* bit to 1. (Enable peripheral I/O function)

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

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

When using the P*xy* port pin as a general-purpose output pin, perform the following software initial settings:

1. Set the PPORTP*x*IOEN.P*x*OEN*y* bit to 1. (Enable output)
2. Set the PPORTP*x*MODSEL.P*x*SEL*y* bit to 0. (Enable GPIO function)

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

When using the P*xy* port pin as a general-purpose input pin, perform the following software initial settings:

1. Write 0 to the PPORTP*x*INTCTL.P*x*IE*y* bit. * (Disable interrupt)
2. When using the chattering filter, configure the PPORT operating clock (see "PPORT Operating Clock") and set the PPORTP*x*CHATEN.P*x*CHATEN*y* bit to 1. *

   When the chattering filter is not used, set the PPORTP*x*CHATEN.P*x*CHATEN*y* bit to 0 (supply of the PPORT operating clock is not required).
3. Configure the following PPORTP*x*RCTL register bits when pulling up/down the port using the internal pull-up or down resistor:
   - PPORTP*x*RCTL.P*x*PDPU*y* bit (Select pull-up or pull-down resistor)
   - Set the PPORTP*x*RCTL.P*x*REN*y* bit to 1. (Enable pull-up/down)

   Set the PPORTP*x*RCTL.P*x*REN*y* bit to 0 if the internal pull-up/down resistors are not used.
4. Set the PPORTP*x*MODSEL.P*x*SEL*y* bit to 0. (Enable GPIO function)
5. Configure the following bits when using the port input interrupt: *
   - Write 1 to the interrupt flags in the PPORTP*x*INTF register. (Clear interrupt flag)
   - Set the interrupt enable bits in the PPORTP*x*INTCTL register to 1. (Enable interrupt)
6. Set the following PPORTP*x*IOEN register bits:
   - Set the PPORTP*x*IOEN.P*x*OEN*y* bit to 0. (Disable output)
   - Set the PPORTP*x*IOEN.P*x*IEN*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 7.4.1.1 lists the port status according to the combination of data input/output control and pull-up/down control.

Table 7.4.1.1 GPIO Port Control List

PPORTP*x*IOEN. P*x*IEN*y* bit	PPORTP*x*IOEN. P*x*OEN*y* bit	PPORTP*x*RCTL. P*x*REN*y* bit	PPORTP*x*RCTL. P*x*PDPU*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 PPORTP*x*MODSEL.P*x*SEL*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.

7.4.2 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 PPORTP*x*DAT.P*x*OUT*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 PPORTP*x*DAT.P*x*IN*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 PPORTP*x*CHATEN.P*x*CHATEN*y* bit to 1. The input sampling time to remove chattering is determined by the CLK_PPORT frequency configured using the PPORTCLK register in common to all ports. The chattering filter removes pulses with a shorter width than the input sampling time.

$$\text{Input sampling time} = \frac{\text{2 to 3}}{\text{CLK\_PPORT frequency [Hz]}}\ \text{[second]} \qquad \text{(Eq. 7.2)}$$

Make sure the P*xy* port interrupt is disabled before altering the PPORTCLK register and PPORTP*x*CHATEN. P*x*CHATEN*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 PPORTCLK.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 PPORTCLK.KRSTCFG[1:0] bits.

PPORT issues a reset request immediately after all the input pins specified by the PPORTCLK.KRSTCFG[1:0] bits 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.

7.5 Interrupts

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

Table 7.5.1 Port Input Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Port input interrupt	PPORTP*x*INTF.P*x*FIF*y*	Falling edge of the input signal	Writing 1
	PPORTP*x*INTF.P*x*RIF*y*	Rising edge of the input signal	Writing 1
	PPORTINTFGRP.P*x*INT	Setting an interrupt flag in the port group	Clearing PPORTP*x*INTF register

Interrupt enable

PPORT provides interrupt enable bits (PPORTP*x*INTCTL.P*x*FIE*y* and P*x*RIE*y* bits) corresponding to each interrupt flag. An interrupt request is sent to the CPU only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. Setting the interrupt enable bits allows port input interrupts to be generated at the falling edge, rising edge, or both edges of the input signal. For more information on interrupt control, refer to the "Interrupt" chapter.

Interrupt check in port group unit

When interrupts are enabled in two or more port groups, check the PPORTINTFGRP.P*x*INT 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 interrupt flag in the PPORTP*x*INTF register of the port group that has been set to 1 to determine the port that has generated an interrupt. Clearing the PPORTP*x*INTF register also clears the PPORTINTFGRP.P*x*INT bit. If a port input interrupt is disabled by the interrupt enable bit in the PPORTP*x*INTCTL register, the PPORTINTFGRP.P*x*INT bit will not be set even if the corresponding interrupt flag is set to 1.

7.6 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."

P*x* Port Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP*x*DAT	15–8	P*x*OUT[7:0]	0x00	H0	R/W	–
	7–0	P*x*IN[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 P*x*OUT[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 (PPORTP*x*IOEN.P*x*OEN*y* bit = 1), the port pin outputs the data set here. Although data can be written when output is disabled (PPORTP*x*IOEN.P*x*OEN*y* bit = 0), it does not affect the pin status. These bits do not affect the outputs when the port is used as a peripheral I/O function.

Bits 7–0 P*x*IN[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 (PPORTP*x*IOEN.P*x*IEN*y* bit = 1). When input is disabled (PPORTP*x*IOEN.P*x*IEN*y* bit = 0), these bits are always read as 0.

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

P*x* Port Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP*x*IOEN	15–8	P*x*IEN[7:0]	0x00	H0	R/W	–
	7–0	P*x*OEN[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 P*x*IEN[7:0]

These bits enable/disable the GPIO port input.

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

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

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

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

Bits 7–0 P*x*OEN[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.

P*x* Port Pull-up/down Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP*x*RCTL	15–8	P*x*PDPU[7:0]	0x00	H0	R/W	–
	7–0	P*x*REN[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 P*x*PDPU[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 PPORTP*x*RCTL.P*x*REN*y* bit = 1.

Bits 7–0 P*x*REN[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 (PPORTP*x*IOEN.P*x*OEN*y* bit = 0). When output is enabled (PPORTP*x*IOEN.P*x*OEN*y* bit = 1), the PPORTP*x*RCTL.P*x*REN*y* bit setting is ineffective regardless of how the PPORTP*x*IOEN.P*x*IEN*y* bit is set and the port is not pulled up/down.

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

P*x* Port Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP*x*INTF	15–8	P*x*FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
	7–0	P*x*RIF[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 P*x*FIF[7:0]
Bits 7–0 P*x*RIF[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

The following shows the correspondence between the bit and interrupt:
PPORTP*x*INTF.P*x*FIF*y* bit: P*xy* port input falling edge interrupt
PPORTP*x*INTF.P*x*RIF*y* bit: P*xy* port input rising edge interrupt

P*x* Port Interrupt Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP*x*INTCTL	15–8	P*x*FIE[7:0]	0x00	H0	R/W	–
	7–0	P*x*RIE[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 P*x*FIE[7:0]
Bits 7–0 P*x*RIE[7:0]

These bits enable port input interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:
PPORTP*x*INTCTL.P*x*FIE*y* bit: P*xy* port input falling edge interrupt
PPORTP*x*INTCTL.P*x*RIE*y* bit: P*xy* port input rising edge interrupt

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

P*x* Port Chattering Filter Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP*x*CHATEN	15–8	–	0x00	–	R	–
	7–0	P*x*CHATEN[7:0]	0x00	H0	R/W	

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

Bits 15–8 Reserved

Bits 7–0 P*x*CHATEN[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.)

P*x* Port Mode Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP*x*MODSEL	15–8	–	0x00	–	R	–
	7–0	P*x*SEL[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 P*x*SEL[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

P*x* Port Function Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP*x*FNCSEL	15–14	P*x*7MUX[1:0]	0x0	H0	R/W	–
	13–12	P*x*6MUX[1:0]	0x0	H0	R/W	
	11–10	P*x*5MUX[1:0]	0x0	H0	R/W	
	9–8	P*x*4MUX[1:0]	0x0	H0	R/W	
	7–6	P*x*3MUX[1:0]	0x0	H0	R/W	
	5–4	P*x*2MUX[1:0]	0x0	H0	R/W	
	3–2	P*x*1MUX[1:0]	0x0	H0	R/W	
	1–0	P*x*0MUX[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 P*x*7MUX[1:0]

: :

Bits 1–0 P*x*0MUX[1:0]

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

Table 7.6.1 Selecting Peripheral I/O Function

PPORTP*x*FNCSEL.P*xy*MUX[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 PPORTP*x*MODSEL.P*x*SEL*y* bit = 1.

P Port Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTCLK	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 during debugging or not.

1 (R/WP): Clock supplied during debugging

0 (R/WP): No clock supplied during debugging

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 7.6.2 Key-Entry Reset Function Settings

PPORTCLK.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 PPORTCLK.CLKSRC[1:0] bits and the clock division ratio using the PPORTCLK.CLKDIV[3:0] bits as shown in Table 7.6.3. These settings determine the input sampling time of the chattering filter.

Table 7.6.3 Clock Source and Division Ratio Settings

PPORTCLK.CLKDIV[3:0] bits	PPORTCLK.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
PPORTINTFGRP	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 P*x*INT

These bits indicate that P*x* 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 PPORTINTFGRP.P*x*INT bit is cleared when the interrupt flag for the port that has generated an interrupt is cleared.

7.7 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.

7.7.1 P0 Port Group

The P0 port group supports the GPIO and interrupt functions.

Table 7.7.1.1 Control Registers for P0 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP0DAT (P0 Port Data Register)	15–8	P0OUT[7:0]	0x00	H0	R/W	–
	7–0	P0IN[7:0]	0x00	H0	R	
PPORTP0IOEN (P0 Port Enable Register)	15–8	P0IEN[7:0]	0x00	H0	R/W	–
	7–0	P0OEN[7:0]	0x00	H0	R/W	
PPORTP0RCTL (P0 Port Pull-up/down Control Register)	15–8	P0PDPU[7:0]	0x00	H0	R/W	–
	7–0	P0REN[7:0]	0x00	H0	R/W	
PPORTP0INTF (P0 Port Interrupt Flag Register)	15–8	P0FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
	7–0	P0RIF[7:0]	0x00	H0	R/W	
PPORTP0INTCTL (P0 Port Interrupt Control Register)	15–8	P0FIE[7:0]	0x00	H0	R/W	–
	7–0	P0RIE[7:0]	0x00	H0	R/W	
PPORTP0CHATEN (P0 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–0	P0CHATEN[7:0]	0x00	H0	R/W	
PPORTP0MODSEL (P0 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P0SEL[7:0]	0x00	H0	R/W	
PPORTP0FNCSEL (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 7.7.1.2 P0 Port Group Function Assignment

Port name	P0SEL*y* = 0	P0SEL*y* = 1							
	GPIO	P0*y*MUX = 0x0 (Function 0)		P0*y*MUX = 0x1 (Function 1)		P0*y*MUX = 0x2 (Function 2)		P0*y*MUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P00	P00	SPIA Ch.1	SDI1	UPMUX	∗1	–	–	–	–
P01	P01	SPIA Ch.1	SDO1	UPMUX	∗1	–	–	–	–
P02	P02	SPIA Ch.1	SPICLK1	UPMUX	∗1	–	–	–	–
P03	P03	SPIA Ch.1	#SPISS1	UPMUX	∗1	–	–	–	–
P04	P04	RFC Ch.0	SENB0	UPMUX	∗1	–	–	–	–
P05	P05	RFC Ch.0	SENA0	UPMUX	∗1	–	–	–	–
P06	P06	RFC Ch.0	REF0	UPMUX	∗1	–	–	–	–
P07	P07	RFC Ch.0	RFIN0	UPMUX	∗1	–	–	–	–

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

7.7.2 P1 Port Group

The P1 port group supports the GPIO and interrupt functions.

Table 7.7.2.1 Control Registers for P1 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP1DAT (P1 Port Data Register)	15–8	P1OUT[7:0]	0x00	H0	R/W	–
	7–0	P1IN[7:0]	0x00	H0	R	
PPORTP1IOEN (P1 Port Enable Register)	15–8	P1IEN[7:0]	0x00	H0	R/W	–
	7–0	P1OEN[7:0]	0x00	H0	R/W	
PPORTP1RCTL (P1 Port Pull-up/down Control Register)	15–8	P1PDPU[7:0]	0x00	H0	R/W	–
	7–0	P1REN[7:0]	0x00	H0	R/W	
PPORTP1INTF (P1 Port Interrupt Flag Register)	15–8	P1FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
	7–0	P1RIF[7:0]	0x00	H0	R/W	
PPORTP1INTCTL (P1 Port Interrupt Control Register)	15–8	P1FIE[7:0]	0x00	H0	R/W	–
	7–0	P1RIE[7:0]	0x00	H0	R/W	
PPORTP1CHATEN (P1 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–0	P1CHATEN[7:0]	0x00	H0	R/W	
PPORTP1MODSEL (P1 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P1SEL[7:0]	0x00	H0	R/W	
PPORTP1FNCSEL (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 7.7.2.2 P1 Port Group Function Assignment

Port name	P1SEL*y* = 0	P1SEL*y* = 1							
	GPIO	P1*y*MUX = 0x0 (Function 0)		P1*y*MUX = 0x1 (Function 1)		P1*y*MUX = 0x2 (Function 2)		P1*y*MUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P10	P10	–	–	UPMUX	*1	ADC12A	ADIN06	–	–
P11	P11	–	–	UPMUX	*1	ADC12A	ADIN05	–	–
P12	P12	–	–	UPMUX	*1	ADC12A	ADIN04	–	–
P13	P13	–	–	UPMUX	*1	ADC12A	ADIN03	–	–
P14	P14	–	–	UPMUX	*1	ADC12A	ADIN02	–	–
P15	P15	T16B Ch.0	EXCL00	UPMUX	*1	ADC12A	ADIN01	–	–
P16	P16	T16B Ch.1	EXCL10	UPMUX	*1	ADC12A	ADIN00	–	–
P17	P17	–	–	UPMUX	*1	ADC12A	VREFA0	–	–

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

7.7.3 P2 Port Group

The P2 port group supports the GPIO and interrupt functions.

Table 7.7.3.1 Control Registers for P2 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP2DAT (P2 Port Data Register)	15–8	P2OUT[7:0]	0x00	H0	R/W	–
	7–0	P2IN[7:0]	0x00	H0	R	
PPORTP2IOEN (P2 Port Enable Register)	15–8	P2IEN[7:0]	0x00	H0	R/W	–
	7–0	P2OEN[7:0]	0x00	H0	R/W	
PPORTP2RCTL (P2 Port Pull-up/down Control Register)	15–8	P2PDPU[7:0]	0x00	H0	R/W	–
	7–0	P2REN[7:0]	0x00	H0	R/W	
PPORTP2INTF (P2 Port Interrupt Flag Register)	15–8	P2FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
	7–0	P2RIF[7:0]	0x00	H0	R/W	
PPORTP2INTCTL (P2 Port Interrupt Control Register)	15–8	P2FIE[7:0]	0x00	H0	R/W	–
	7–0	P2RIE[7:0]	0x00	H0	R/W	
PPORTP2CHATEN (P2 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–0	P2CHATEN[7:0]	0x00	H0	R/W	
PPORTP2MODSEL (P2 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P2SEL[7:0]	0x00	H0	R/W	
PPORTP2FNCSEL (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 7.7.3.2 P2 Port Group Function Assignment

Port name	P2SEL*y* = 0	P2SEL*y* = 1							
	GPIO	P2*y*MUX = 0x0 (Function 0)		P2*y*MUX = 0x1 (Function 1)		P2*y*MUX = 0x2 (Function 2)		P2*y*MUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P20	P20	SNDA	BZOUT	UPMUX	*1	–	–	–	–
P21	P21	SNDA	#BZOUT	UPMUX	*1	–	–	–	–
P22	P22	REMC3	REMO	UPMUX	*1	–	–	–	–
P23	P23	REMC3	CLPLS	UPMUX	*1	–	–	–	–
P24	P24	T16 Ch.2	EXCL20	UPMUX	*1	–	–	–	–
P25	P25	T16 Ch.0	EXCL01	UPMUX	*1	–	–	LCD8D	SEG55
P26	P26	T16 Ch.1	EXCL11	UPMUX	*1	–	–	LCD8D	SEG54
P27	P27	T16 Ch.2	EXCL21	UPMUX	*1	–	–	LCD8D	SEG53

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

7.7.4 P3 Port Group

The P3 port group supports the GPIO and interrupt functions.

Table 7.7.4.1 Control Registers for P3 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP3DAT (P3 Port Data Register)	15–8	P3OUT[7:0]	0x00	H0	R/W	–
	7–0	P3IN[7:0]	0x00	H0	R	
PPORTP3IOEN (P3 Port Enable Register)	15–8	P3IEN[7:0]	0x00	H0	R/W	–
	7–0	P3OEN[7:0]	0x00	H0	R/W	
PPORTP3RCTL (P3 Port Pull-up/down Control Register)	15–8	P3PDPU[7:0]	0x00	H0	R/W	–
	7–0	P3REN[7:0]	0x00	H0	R/W	
PPORTP3INTF (P3 Port Interrupt Flag Register)	15–8	P3FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
	7–0	P3RIF[7:0]	0x00	H0	R/W	
PPORTP3INTCTL (P3 Port Interrupt Control Register)	15–8	P3FIE[7:0]	0x00	H0	R/W	–
	7–0	P3RIE[7:0]	0x00	H0	R/W	
PPORTP3CHATEN (P3 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–0	P3CHATEN[7:0]	0x00	H0	R/W	
PPORTP3MODSEL (P3 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P3SEL[7:0]	0x00	H0	R/W	
PPORTP3FNCSEL (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 7.7.4.2 P3 Port Group Function Assignment

Port name	P3SEL*y* = 0	P3SEL*y* = 1							
	GPIO	P3*y*MUX = 0x0 (Function 0)		P3*y*MUX = 0x1 (Function 1)		P3*y*MUX = 0x2 (Function 2)		P3*y*MUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P30	P30	CLG	EXOSC	UPMUX	*1	–	–	LCD8D	SEG47
P31	P31	ADC12A	#ADTRG0	UPMUX	*1	–	–	LCD8D	SEG46
P32	P32	RTCA	RTC1S	UPMUX	*1	–	–	LCD8D	SEG45
P33	P33	CLG	FOUT	UPMUX	*1	–	–	LCD8D	SEG44
P34	P34	LCD8D	LFRO	UPMUX	*1	–	–	LCD8D	SEG43
P35	P35	–	–	UPMUX	*1	SVD4 Ch.0	EXSVD00	LCD8D	SEG42
P36	P36	–	–	UPMUX	*1	SVD4 Ch.0	EXSVD01	LCD8D	SEG41
P37	P37	RFC Ch.0	RFCLKO0	UPMUX	*1	–	–	LCD8D	SEG40

*1: Refer to the "Universal Port Multiplexer" chapter.

7.7.5 P4 Port Group

The P4 port group supports the GPIO and interrupt functions.

Table 7.7.5.1 Control Registers for P4 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP4DAT (P4 Port Data Register)	15–8	P4OUT[7:0]	0x00	H0	R/W	–
	7–0	P4IN[7:0]	0x00	H0	R	
PPORTP4IOEN (P4 Port Enable Register)	15–8	P4IEN[7:0]	0x00	H0	R/W	–
	7–0	P4OEN[7:0]	0x00	H0	R/W	
PPORTP4RCTL (P4 Port Pull-up/down Control Register)	15–8	P4PDPU[7:0]	0x00	H0	R/W	–
	7–0	P4REN[7:0]	0x00	H0	R/W	
PPORTP4INTF (P4 Port Interrupt Flag Register)	15–8	P4FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
	7–0	P4RIF[7:0]	0x00	H0	R/W	
PPORTP4INTCTL (P4 Port Interrupt Control Register)	15–8	P4FIE[7:0]	0x00	H0	R/W	–
	7–0	P4RIE[7:0]	0x00	H0	R/W	
PPORTP4CHATEN (P4 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–0	P4CHATEN[7:0]	0x00	H0	R/W	
PPORTP4MODSEL (P4 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P4SEL[7:0]	0x00	H0	R/W	
PPORTP4FNCSEL (P4 Port Function Select Register)	15–14	P47MUX[1:0]	0x3	H0	R/W	–
	13–12	P46MUX[1:0]	0x3	H0	R/W	
	11–10	P45MUX[1:0]	0x3	H0	R/W	
	9–8	P44MUX[1:0]	0x3	H0	R/W	
	7–6	P43MUX[1:0]	0x3	H0	R/W	
	5–4	P42MUX[1:0]	0x3	H0	R/W	
	3–2	P41MUX[1:0]	0x3	H0	R/W	
	1–0	P40MUX[1:0]	0x3	H0	R/W	

Table 7.7.5.2 P4 Port Group Function Assignment

Port name	P4SEL*y* = 0	P4SEL*y* = 1							
	GPIO	P4*y*MUX = 0x0 (Function 0)		P4*y*MUX = 0x1 (Function 1)		P4*y*MUX = 0x2 (Function 2)		P4*y*MUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P40	P40	–	–	–	–	–	–	LCD8D	SEG39
P41	P41	–	–	–	–	–	–	LCD8D	SEG38
P42	P42	–	–	–	–	–	–	LCD8D	SEG37
P43	P43	–	–	–	–	–	–	LCD8D	SEG36
P44	P44	–	–	–	–	–	–	LCD8D	SEG35
P45	P45	–	–	–	–	–	–	LCD8D	SEG34
P46	P46	–	–	–	–	–	–	LCD8D	SEG33
P47	P47	–	–	–	–	–	–	LCD8D	SEG32

7.7.6 P5 Port Group

The P5 port group supports the GPIO and interrupt functions.

Table 7.7.6.1 Control Registers for P5 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP5DAT (P5 Port Data Register)	15–8	P5OUT[7:0]	0x00	H0	R/W	–
	7–0	P5IN[7:0]	0x00	H0	R	
PPORTP5IOEN (P5 Port Enable Register)	15–8	P5IEN[7:0]	0x00	H0	R/W	–
	7–0	P5OEN[7:0]	0x00	H0	R/W	
PPORTP5RCTL (P5 Port Pull-up/down Control Register)	15–8	P5PDPU[7:0]	0x00	H0	R/W	–
	7–0	P5REN[7:0]	0x00	H0	R/W	
PPORTP5INTF (P5 Port Interrupt Flag Register)	15–8	P5FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
	7–0	P5RIF[7:0]	0x00	H0	R/W	
PPORTP5INTCTL (P5 Port Interrupt Control Register)	15–8	P5FIE[7:0]	0x00	H0	R/W	–
	7–0	P5RIE[7:0]	0x00	H0	R/W	
PPORTP5CHATEN (P5 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–0	P5CHATEN[7:0]	0x00	H0	R/W	
PPORTP5MODSEL (P5 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P5SEL[7:0]	0x00	H0	R/W	
PPORTP5FNCSEL (P5 Port Function Select Register)	15–14	P57MUX[1:0]	0x3	H0	R/W	–
	13–12	P56MUX[1:0]	0x3	H0	R/W	
	11–10	P55MUX[1:0]	0x3	H0	R/W	
	9–8	P54MUX[1:0]	0x3	H0	R/W	
	7–6	P53MUX[1:0]	0x3	H0	R/W	
	5–4	P52MUX[1:0]	0x3	H0	R/W	
	3–2	P51MUX[1:0]	0x3	H0	R/W	
	1–0	P50MUX[1:0]	0x3	H0	R/W	

Table 7.7.6.2 P5 Port Group Function Assignment

Port name	P5SEL*y* = 0	P5SEL*y* = 1							
	GPIO	P5*y*MUX = 0x0 (Function 0)		P5*y*MUX = 0x1 (Function 1)		P5*y*MUX = 0x2 (Function 2)		P5*y*MUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P50	P50	–	–	–	–	–	–	LCD8D	SEG24
P51	P51	–	–	–	–	–	–	LCD8D	SEG23
P52	P52	–	–	–	–	–	–	LCD8D	SEG22
P53	P53	–	–	–	–	–	–	LCD8D	SEG21
P54	P54	–	–	–	–	–	–	LCD8D	SEG20
P55	P55	–	–	–	–	–	–	LCD8D	SEG19
P56	P56	–	–	–	–	–	–	LCD8D	SEG18
P57	P57	–	–	–	–	–	–	LCD8D	SEG17

7.7.7 P6 Port Group

The P6 port group supports the GPIO and interrupt functions.

Table 7.7.7.1 Control Registers for P6 Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTP6DAT (P6 Port Data Register)	15–8	P6OUT[7:0]	0x00	H0	R/W	–
	7–0	P6IN[7:0]	0x00	H0	R	
PPORTP6IOEN (P6 Port Enable Register)	15–8	P6IEN[7:0]	0x00	H0	R/W	–
	7–0	P6OEN[7:0]	0x00	H0	R/W	
PPORTP6RCTL (P6 Port Pull-up/down Control Register)	15–8	P6PDPU[7:0]	0x00	H0	R/W	–
	7–0	P6REN[7:0]	0x00	H0	R/W	
PPORTP6INTF (P6 Port Interrupt Flag Register)	15–8	P6FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
	7–0	P6RIF[7:0]	0x00	H0	R/W	
PPORTP6INTCTL (P6 Port Interrupt Control Register)	15–8	P6FIE[7:0]	0x00	H0	R/W	–
	7–0	P6RIE[7:0]	0x00	H0	R/W	
PPORTP6CHATEN (P6 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
	7–0	P6CHATEN[7:0]	0x00	H0	R/W	
PPORTP6MODSEL (P6 Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	P6SEL[7:0]	0x00	H0	R/W	
PPORTP6FNCSEL (P6 Port Function Select Register)	15–14	P67MUX[1:0]	0x3	H0	R/W	–
	13–12	P66MUX[1:0]	0x3	H0	R/W	
	11–10	P65MUX[1:0]	0x3	H0	R/W	
	9–8	P64MUX[1:0]	0x3	H0	R/W	
	7–6	P63MUX[1:0]	0x3	H0	R/W	
	5–4	P62MUX[1:0]	0x3	H0	R/W	
	3–2	P61MUX[1:0]	0x3	H0	R/W	
	1–0	P60MUX[1:0]	0x3	H0	R/W	

Table 7.7.7.2 P6 Port Group Function Assignment

Port name	P6SEL*y* = 0	P6SEL*y* = 1							
	GPIO	P6*y*MUX = 0x0 (Function 0)		P6*y*MUX = 0x1 (Function 1)		P6*y*MUX = 0x2 (Function 2)		P6*y*MUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
P60	P60	–	–	–	–	–	–	LCD8D	SEG16
P61	P61	–	–	–	–	–	–	LCD8D	SEG15
P62	P62	–	–	–	–	–	–	LCD8D	SEG14
P63	P63	–	–	–	–	–	–	LCD8D	SEG13
P64	P64	–	–	–	–	–	–	LCD8D	COM7/SEG3
P65	P65	–	–	–	–	–	–	LCD8D	COM6/SEG2
P66	P66	–	–	–	–	–	–	LCD8D	COM5/SEG1
P67	P67	–	–	–	–	–	–	LCD8D	COM4/SEG0

7.7.8 Pd Port Group

The Pd port group supports the GPIO function. Two ports Pd0–Pd1 are configured as debugging function ports at initialization.

Table 7.7.8.1 Control Registers for Pd Port Group

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTPDDAT (Pd Port Data Register)	15–8	PDOUT[7:0]	0x00	H0	R/W	–
	7–5	PDIN[7:5]	0x0	H0	R	
	4	(reserved)	0	–	R	
	3–0	PDIN[3:0]	0x00	H0	R	
PPORTPDIOEN (Pd Port Enable Register)	15–13	PDIEN[7:5]	0x0	H0	R/W	–
	12	(reserved)	0	H0	R/W	
	11–8	PDIEN[3:0]	0x0	H0	R/W	
	7–0	PDOEN[7:0]	0x10	H0	R/W	
PPORTPDRCTL (Pd Port Pull-up/down Control Register)	15–13	PDPDPU[7:5]	0x0	H0	R/W	–
	12	(reserved)	0	H0	R/W	
	11–8	PDPDPU[3:0]	0x0	H0	R/W	
	7–5	PDREN[7:5]	0x0	H0	R/W	
	4	(reserved)	0	H0	R/W	
	3–0	PDREN[3:0]	0x0	H0	R/W	
PPORTPDINTF PPORTPDINTCTL PPORTPDCHATEN	15–0	–	0x0000	–	R	–
PPORTPDMODSEL (Pd Port Mode Select Register)	15–8	–	0x00	–	R	–
	7–0	PDSEL[7:0]	0x23	H0	R/W	
PPORTPDFNCSEL (Pd Port Function Select Register)	15–14	PD7MUX[1:0]	0x0	H0	R/W	–
	13–12	PD6MUX[1:0]	0x0	H0	R/W	
	11–10	PD5MUX[1:0]	0x2	H0	R/W	
	9–8	(reserved)	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 7.7.8.2 Pd Port Group Function Assignment

Port name	PdSEL*y* = 0	PdSEL*y* = 1							
	GPIO	Pd*y*MUX = 0x0 (Function 0)		Pd*y*MUX = 0x1 (Function 1)		Pd*y*MUX = 0x2 (Function 2)		Pd*y*MUX = 0x3 (Function 3)	
		Peripheral	Pin	Peripheral	Pin	Peripheral	Pin	Peripheral	Pin
Pd0	Pd0	CPU	SWCLK	–	–	–	–	–	–
Pd1	Pd1	CPU	SWD	–	–	–	–	–	–
Pd2	Pd2	–	–	–	–	CLG	OSC3	–	–
Pd3	Pd3	–	–	–	–	CLG	OSC4	–	–
Pd4	Pd4	–	–	–	–	–	–	–	–
Pd5	Pd5	–	–	–	–	FLASHC	V_{PP}	–	–
Pd6	Pd6	–	–	–	–	LCD8D	C_{P1}	–	–
Pd7	Pd7	–	–	–	–	LCD8D	C_{P2}	–	–

7.7.9 Common Registers between Port Groups

Table 7.7.9.1 Control Registers for Common Use with Port Groups

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
PPORTCLK (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	
PPORTINTFGRP (P Port Interrupt Flag Group Register)	15–8	–	0x00	–	R	–
	7	–	0	–	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	

8 Universal Port Multiplexer (UPMUX)

8.1 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 I[2]C, SPI, 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 PPORTP*x*FNCSEL.P*xy*MUX[1:0] bits to 0x1.

Note: '*x*', which is used in the port names P*xy*, register names, and bit names, refers to a port group (*x* = 0, 1, 3) and '*y*' refers to a port number (*y* = 0, 1, 2, ··· , 7).

Figure 8.1.1 shows the configuration of UPMUX.

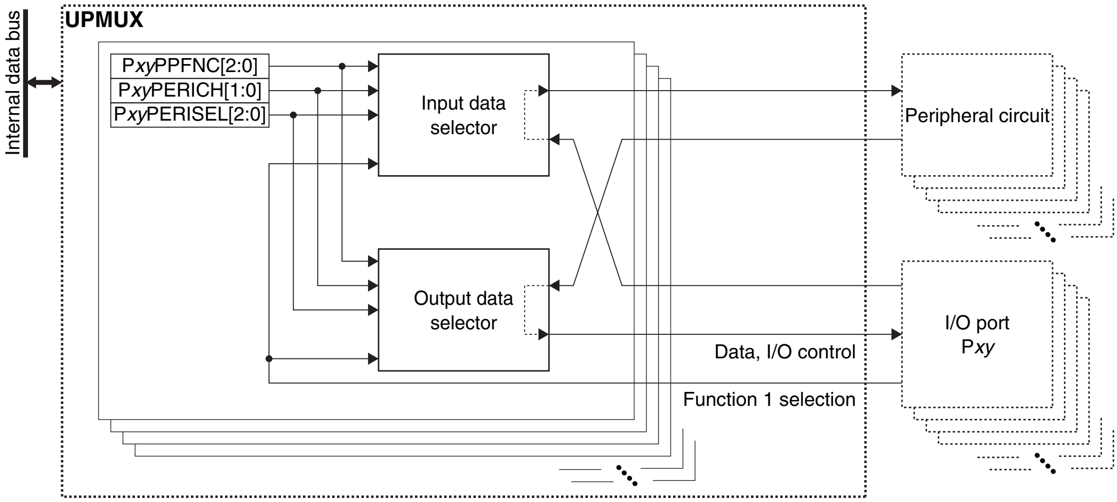


Figure 8.1.1 UPMUX Configuration

8.2 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 PPORTP*x*IOEN register of the I/O port.
   - Set the PPORTP*x*IOEN.P*x*IEN*y* bit to 0. (Disable input)
   - Set the PPORTP*x*IOEN.P*x*OEN*y* bit to 0. (Disable output)
2. Set the PPORTP*x*MODSEL.P*x*SEL*y* bit of the I/O port to 0. (Disable peripheral I/O function)
3. Set the following UPMUXP*x*MUX*n* register bits (*n* = 0 to 3).
   - UPMUXP*x*MUX*n*.P*xy*PERISEL[2:0] bits (Select peripheral circuit)
   - UPMUXP*x*MUX*n*.P*xy*PERICH[1:0] bits (Select peripheral circuit channel)
   - UPMUXP*x*MUX*n*.P*xy*PPFNC[2:0] bits (Select function to assign)
4. Initialize the peripheral circuit.
5. Set the PPORTP*x*FNCSEL.P*xy*MUX[1:0] bits of the I/O port to 0x1. (Select peripheral I/O function 1)
6. Set the PPORTP*x*MODSEL.P*x*SEL*y* bit of the I/O port to 1. (Enable peripheral I/O function)

8.3 Control Registers

P*xy*–*xz* Universal Port Multiplexer Setting Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UPMUXP*x*MUX*n*	15–13	P*xz*PPFNC[2:0]	0x0	H0	R/W	–
	12–11	P*xz*PERICH[1:0]	0x0	H0	R/W	
	10–8	P*xz*PERISEL[2:0]	0x0	H0	R/W	
	7–5	P*xy*PPFNC[2:0]	0x0	H0	R/W	
	4–3	P*xy*PERICH[1:0]	0x0	H0	R/W	
	2–0	P*xy*PERISEL[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 P*xz*PPFNC[2:0]
Bits 7–5 P*xy*PPFNC[2:0]

These bits specify the peripheral I/O function to be assigned to the port. (See Table 8.3.1.)

Bits 12–11 P*xz*PERICH[1:0]
Bits 4–3 P*xy*PERICH[1:0]

These bits specify a peripheral circuit channel number. (See Table 8.3.1.)

Bits 10–8 P*xz*PERISEL[2:0]
Bits 2–0 P*xy*PERISEL[2:0]

These bits specify a peripheral circuit. (See Table 8.3.1.)

Table 8.3.1 Peripheral I/O Function Selections

<table>
<tr><th rowspan="6">UPMUXPxMUXn.PxyPPFNC[2:0] bits (Peripheral I/O function)</th><th colspan="8">UPMUXPxMUXn.PxyPERISEL[2:0] bits (Peripheral circuit)</th></tr>
<tr><td>0x0</td><td>0x1</td><td>0x2</td><td>0x3</td><td>0x4</td><td>0x5</td><td>0x6</td><td>0x7</td></tr>
<tr><td>None ∗</td><td>I2C</td><td>SPIA</td><td>UART3</td><td>T16B</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><th colspan="8">UPMUXPxMUXn.PxyPERICH[1:0] bits (Peripheral circuit channel)</th></tr>
<tr><td>–</td><td>0x0–0x1</td><td>0x0</td><td>0x0–0x1</td><td>0x0–0x2</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>–</td><td>Ch.0–1</td><td>Ch.0</td><td>Ch.0–1</td><td>Ch.0–2</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>0x0</td><td>None ∗</td><td>None ∗</td><td>None ∗</td><td>None ∗</td><td>None ∗</td><td>None ∗</td><td>None ∗</td><td>None ∗</td></tr>
<tr><td>0x1</td><td rowspan="7">Reserved</td><td>SCLn</td><td>SDIn</td><td>USINn</td><td>TOUTn0/
CAPn0</td><td rowspan="7">Reserved</td><td rowspan="7">Reserved</td><td rowspan="7">Reserved</td></tr>
<tr><td>0x2</td><td>SDAn</td><td>SDOn</td><td>USOUTn</td><td>TOUTn1/
CAPn1</td></tr>
<tr><td>0x3</td><td rowspan="5">Reserved</td><td>SPICLKn</td><td rowspan="5">Reserved</td><td>TOUTn2/
CAPn2</td></tr>
<tr><td>0x4</td><td>#SPISSn</td><td>TOUTn3/
CAPn3</td></tr>
<tr><td>0x5</td><td rowspan="3">Reserved</td><td rowspan="3">Reserved</td></tr>
<tr><td>0x6</td></tr>
<tr><td>0x7</td></tr>
</table>

∗ “None” means no assignment. Selecting this will put the P*xy* 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.

9 Watchdog Timer (WDT2)

9.1 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 9.1.1 shows the configuration of WDT2.

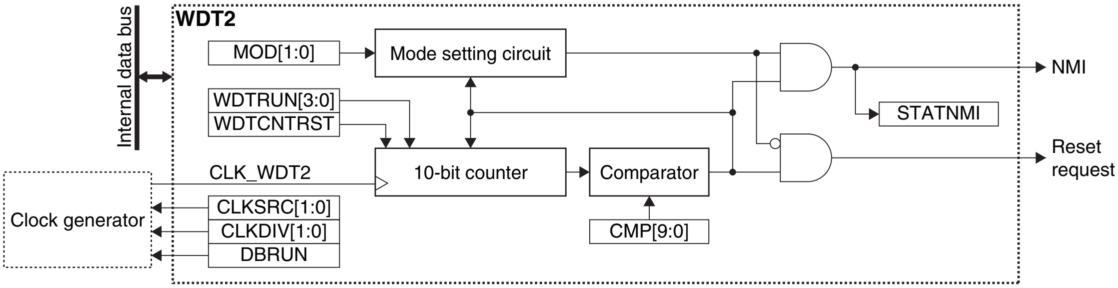


Figure 9.1.1 WDT2 Configuration

9.2 Clock Settings

9.2.1 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 SYSPROT.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 WDT2CLK register bits:
   WDT2CLK.CLKSRC[1:0] bits (Clock source selection)
   WDT2CLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

9.2.2 Clock Supply in DEBUG Mode

The CLK_WDT2 supply during DEBUG mode should be controlled using the WDT2CLK.DBRUN bit.
The CLK_WDT2 supply to WDT2 is suspended when the CPU enters DEBUG mode if the WDT2CLK.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 WDT2CLK.DBRUN bit = 1, the CLK_WDT2 supply is not suspended and WDT2 will keep operating in DEBUG mode.

9.3 Operations

9.3.1 WDT2 Control

Activating WDT2

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

1. Write 0x0096 to the SYSPROT.PROT[15:0] bits. (Remove system protection)
2. Configure the WDT2 operating clock.
3. Set the WDT2CTL.MOD[1:0] bits. (Select WDT2 operating mode)
4. Set the WDT2CMP.CMP[9:0] bits. (Set NMI/reset generation cycle)
5. Write 1 to the WDT2CTL.WDTCNTRST bit. (Reset WDT2 counter)
6. Write a value other than 0xa to the WDT2CTL.WDTRUN[3:0] bits. (Start up WDT2)
7. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

NMI/reset generation cycle

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

$$t_{WDT} = \frac{CMP + 1}{CLK\_WDT2} \quad \text{(Eq. 9.1)}$$

Where

t_{WDT}: NMI/reset generation cycle [second]
CLK_WDT2: WDT2 operating clock frequency [Hz]
CMP: Setting value of the WDT2CMP.CMP[9:0] bits

Example) t_{WDT} = 2.5 seconds when CLK_WDT2 = 256 Hz and the WDT2CMP.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 SYSPROT.PROT[15:0] bits. (Remove system protection)
2. Write 1 to the WDT2CTL.WDTCNTRST bit. (Reset WDT2 counter)
3. Write a value other than 0x0096 to the SYSPROT.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 WDT2CMP.CMP[9:0] bits, a compare match occurs to cause WDT2 to issue an NMI or reset according to the setting of the WDT2CTL.MOD[1:0] bits.
If an NMI is issued, the WDT2CTL.STATNMI bit is set to 1. This bit can be cleared to 0 by writing 1 to the WDT2CTL.WDTCNTRST bit. Be sure to clear the WDT2CTL.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 SYSPROT.PROT[15:0] bits. (Remove system protection)
2. Write 0xa to the WDT2CTL.WDTRUN[3:0] bits. (Stop WDT2)
3. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

9.3.2 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 WDT2CTL.WDTRUN[3:0] bits before setting to HALT mode. 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 WDT2CTL.WDTRUN[3:0] bits before setting to SLEEP mode.
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 setting to SLEEP mode. WDT2 should also be stopped as required using the WDT2CTL.WDTRUN[3:0] bits.

9.4 Control Registers

WDT2 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
WDT2CLK	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 9.4.1 Clock Source and Division Ratio Settings

WDT2CLK.CLKDIV[1:0] bits	WDT2CLK.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
WDT2CTL	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 9.4.2 Operating Mode Setting

WDT2CTL. MOD[1:0] bits	Operating mode	Description
0x3	Reserved	–
0x2	RESET after NMI mode	If the WDT2CTL.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 WDT2CTL.STATNMI bit set to 1 is cleared to 0 by writing 1 to the WDT2CTL.WDTCNTRST bit.

Bits 7–5 Reserved

Bit 4 WDTCNTRST

This bit resets the 10-bit counter and the WDT2CTL.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
WDT2CMP	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.

10 Real-Time Clock (RTCA)

10.1 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 10.1.1 shows the configuration of RTCA.

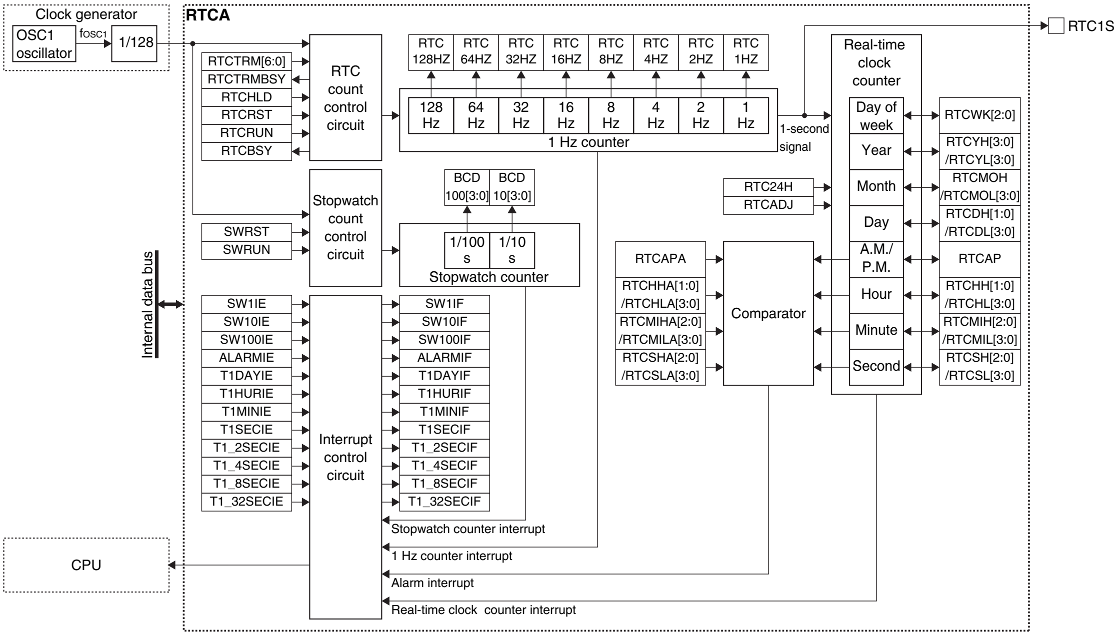


Figure 10.1.1 RTCA Configuration

10.2 Output Pin and External Connection

10.2.1 Output Pin

Table 10.2.1.1 shows the RTCA pin.

Table 10.2.1.1 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.

10.3 Clock Settings

10.3.1 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.

10.3.2 Theoretical Regulation Function

The time-of-day clock loses accuracy if the OSC1 frequency f_{OSC1} 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 f_{OSC1} and calculate the frequency tolerance correction value
   "m [ppm] = -{(f_{OSC1} - 32,768 [Hz]) / 32,768 [Hz]} × 10^6."
2. Determine the theoretical regulation execution cycle time "n seconds."
3. Determine the value to be written to the RTCACTLH.RTCTRM[6:0] bits from the results in Steps 1 and 2.
4. Write the value determined in Step 3 to the RTCACTLH.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 RTCACTLH.RTCTRM[6:0] bits as a two's-complement number. Use Eq. 10.1 to calculate the correction value.

$$\text{RTCTRM[6:0]} = \frac{m}{10^6} \times 256 \times n \quad \text{(However, RTCTRM[6:0] is an integer after rounding off to -64 to +63.)} \qquad \text{(Eq. 10.1)}$$

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

Figure 10.3.2.1 shows the RTC1S signal waveform.

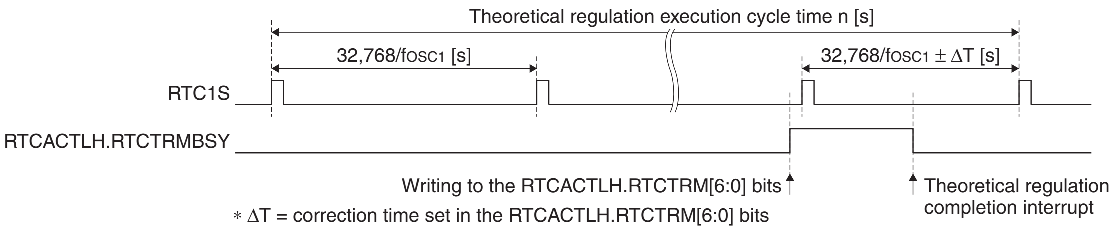


Figure 10.3.2.1 RTC1S Signal Waveform

Table 10.3.2.1 lists the frequency tolerance correction rates when the theoretical regulation execution cycle time n is 4,096 seconds as an example.

Table 10.3.2.1 Correction Rates when Theoretical Regulation Execution Cycle Time n = 4,096 Seconds

RTCACTLH.RTCTRM[6:0] bits (two's-complement)	Correction value (decimal)	Correction rate [ppm]	RTCACTLH.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 RTCACTLH.RTCTRM[6:0] bits, 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.

10.4 Operations

10.4.1 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 RTCACTLL.RTC24H bit.
2. Write 1 to the RTCACTLL.RTCRUN bit to enable for the real-time clock counter to start counting up.
3. Check to see if the RTCACTLL.RTCBSY bit = 0 that indicates the counter is ready to rewrite. If the RTCACTLL.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.
   RTCASEC.RTCSH[2:0]/RTCSL[3:0] bits (second)
   RTCAHUR.RTCMIH[2:0]/RTCMIL[3:0] bits (minute)
   RTCAHUR.RTCHH[1:0]/RTCHL[3:0] bits (hour)
   RTCAHUR.RTCAP bit (AM/PM) (effective when RTCACTLL.RTC24H bit = 0)
   RTCAMON.RTCDH[1:0]/RTCDL[3:0] bits (day)
   RTCAMON.RTCMOH/RTCMOL[3:0] bits (month)
   RTCAYAR.RTCYH[3:0]/RTCYL[3:0] bits (year)
   RTCAYAR.RTCWK[2:0] bits (day of the week)
5 Write 1 to the RTCACTLL.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 RTCAINTF register to clear them.
7. Write 1 to the interrupt enable bits in the RTCAINTE register to enable real-time clock counter interrupts.

Time read

1. Check to see if the RTCACTLL.RTCBSY bit = 0. If the RTCACTLL.RTCBSY bit = 1, wait until it is set to 0.
2. Write 1 to the RTCACTLL.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 RTCACTLL.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 RTCAINTE.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).
   RTCAALM1.RTCSHA[2:0]/RTCSLA[3:0] bits (second)
   RTCAALM2.RTCMIHA[2:0]/RTCMILA[3:0] bits (minute)
   RTCAALM2.RTCHHA[1:0]/RTCHLA[3:0] bits (hour)
   RTCAALM2.RTCAPA bit (AM/PM) (effective when RTCACTLL.RTC24H bit = 0)
3. Write 1 to the RTCAINTF.ALARMIF bit to clear the alarm interrupt flag.
4. Write 1 to the RTCAINTE.ALARMIE bit to enable alarm interrupts.
   When the real-time clock counter reaches the alarm time set in Step 2, an alarm interrupt occurs.

10.4.2 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 RTCAMON.RTCMOL[3:0] bits to 0x0 if the RTCAMON.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 RTCACTLL.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 RTCACTLL.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 RTCACTLL.RTCHLD bit.

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

10.4.3 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 RTCASWCTL.SWRST bit to reset the stopwatch counter.
2. Write 1 to the stopwatch interrupt flags in the RTCAINTF register to clear them.
3. Write 1 to the interrupt enable bits in the RTCAINTE register to enable stopwatch interrupts.
4. Write 1 to the RTCASWCTL.SWRUN bit to start stopwatch count up operation.

Counter read

1. Read the count value from the RTCASWCTL.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.

10.4.4 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 10.4.4.1.

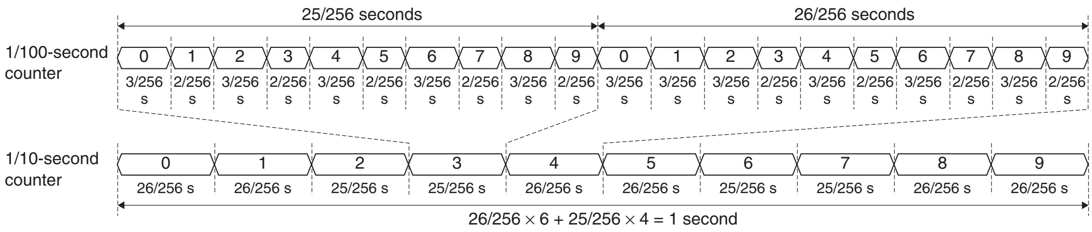


Figure 10.4.4.1 Stopwatch Count-Up Patterns

10.5 Interrupts

RTCA has a function to generate the interrupts shown in Table 10.5.1.

Table 10.5.1 RTCA Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Alarm	RTCAINTF.ALARMIF	Matching between the RTCAALM1–2 register contents and the real-time clock counter contents	Writing 1
1-day	RTCAINTF.T1DAYIF	Day counter count up	Writing 1
1-hour	RTCAINTF.T1HURIF	Hour counter count up	Writing 1
1-minute	RTCAINTF.T1MINIF	Minute counter count up	Writing 1
1-second	RTCAINTF.T1SECIF	Second counter count up	Writing 1
1/2-second	RTCAINTF.T1_2SECIF	See Figure 10.5.1.	Writing 1
1/4-second	RTCAINTF.T1_4SECIF	See Figure 10.5.1.	Writing 1
1/8-second	RTCAINTF.T1_8SECIF	See Figure 10.5.1.	Writing 1
1/32-second	RTCAINTF.T1_32SECIF	See Figure 10.5.1.	Writing 1
Stopwatch 1 Hz	RTCAINTF.SW1IF	1/10-second counter overflow	Writing 1
Stopwatch 10 Hz	RTCAINTF.SW10IF	1/10-second counter count up	Writing 1
Stopwatch 100 Hz	RTCAINTF.SW100IF	1/100-second counter count up	Writing 1
Theoretical regulation completion	RTCAINTF.RTCTRMIF	At the end of theoretical regulation operation	Writing 1

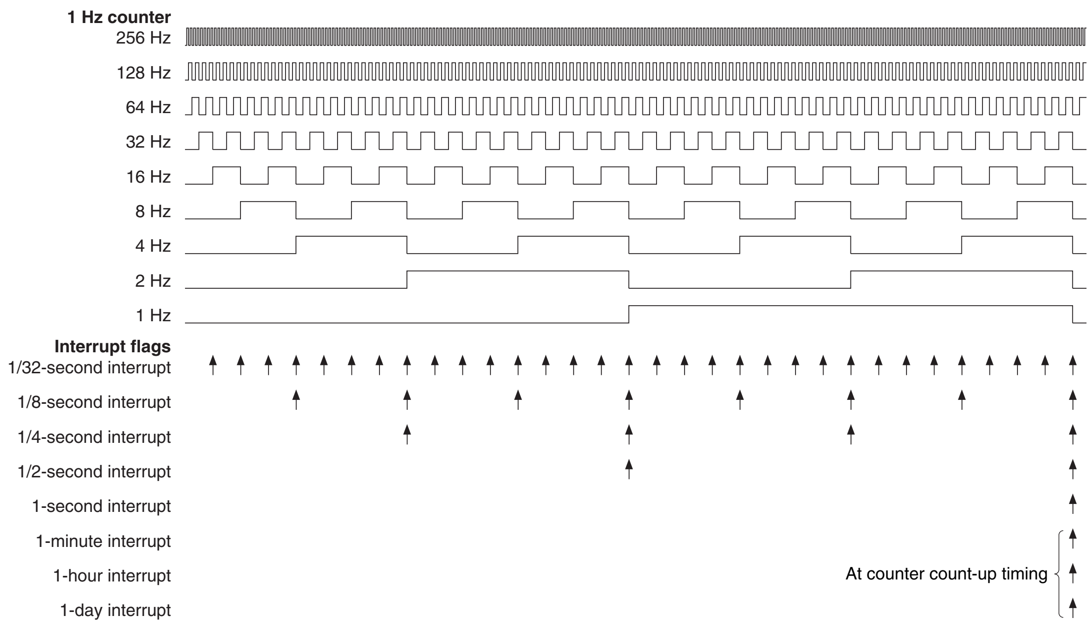


Figure 10.5.1 RTCA Interrupt Timings

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 CPU 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" chapter.

10.6 Control Registers

RTCA Control Register (Low Byte)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCACTLL	7	–	0	–	R	–
	6	RTCBSY	0	H0	R	
	5	RTCHLD	0	H0	R/W	Cleared by setting the RTCACTLL.RTCRST bit to 1.
	4	RTC24H	0	H0	R/W	–
	3	–	0	–	R	
	2	RTCADJ	0	H0	R/W	Cleared by setting the RTCACTLL.RTCRST bit to 1.
	1	RTCRST	0	H0	R/W	–
	0	RTCRUN	0	H0	R/W	

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 RTCACTLH.RTCTRMBSY bit = 1, the RTCACTLL.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 RTCACTLL.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 RTCACTLL.RTCRUN bit = 0. The correction takes up to 2/256 seconds. The RTCACTLL.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 RTCACTLL.RTCBSY bit = 1.

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

Bit 1 RTCRST

This bit resets the 1 Hz counter, the RTCACTLL.RTCADJ bit, and the RTCACTLL.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.

RTCA Control Register (High Byte)

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCACTLH	7	RTCTRMBSY	0	H0	R	–
	6–0	RTCTRM[6:0]	0x00	H0	W	Read as 0x00.

Bit 7 RTCTRMBSY

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 RTCACTLH.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 6–0 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 RTCACTLH.RTCTRMBSY bit = 1, the RTCACTLH.RTCTRM[6:0] bits cannot be rewritten.

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

RTCA Second Alarm Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCAALM1	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 RTCAALM1.RTCSHA[2:0] bits and the RTCAALM1.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 10.6.1.

Table 10.6.1 Setting Examples in BCD Code

Setting value in BCD code		Alarm (second) setting
RTCAALM1.RTCSHA[2:0] bits	RTCAALM1.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

RTCA Hour/Minute Alarm Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCAALM2	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 (RTCACTLL.RTC24H bit = 0).
1 (R/W): P.M.
0 (R/W): A.M.

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

Bits 13–12 RTCHHA[1:0]
Bits 11–8 RTCHLA[3:0]

The RTCAALM2.RTCHHA[1:0] bits and the RTCAALM2.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 RTCAALM2.RTCMIHA[2:0] bits and the RTCAALM2.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.

RTCA Stopwatch Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCASWCTL	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 RTCASWCTL.BCD10[3:0] bits and the RTCASWCTL.BCD100[3:0] bits, respectively.

Note: The counter value may not be read correctly while the stopwatch counter is running. The RTCASWCTL.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 RTCASWCTL.SWRUN bit. Therefore, the counter value may be incremented (+1) from the value at writing 0.

RTCA Second/1Hz Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCASEC	15	–	0	–	R	–
	14–12	RTCSH[2:0]	0x0	H0	R/W	
	11–8	RTCSL[3:0]	0x0	H0	R/W	
	7	RTC1HZ	0	H0	R	Cleared by setting the RTCACTLL.RTCRST bit to 1.
	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 RTCASEC.RTCSH[2:0] bits and the RTCASEC.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 RTCASEC.RTCSH[2:0]/RTCSL[3:0] bits while the RTCACTLL.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:
RTCASEC.RTC1HZ bit: 1 Hz
RTCASEC.RTC2HZ bit: 2 Hz
RTCASEC.RTC4HZ bit: 4 Hz
RTCASEC.RTC8HZ bit: 8 Hz
RTCASEC.RTC16HZ bit: 16 Hz
RTCASEC.RTC32HZ bit: 32 Hz
RTCASEC.RTC64HZ bit: 64 Hz
RTCASEC.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.

RTCA Hour/Minute Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCAHUR	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 (RTCACTLL.RTC24H bit = 0).
1 (R/W): P.M.
0 (R/W): A.M.

In 24H mode (RTCACTLL.RTC24H bit = 1), this bit is fixed at 0 and writing 1 is ignored. However, if the RTCAHUR.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 RTCAHUR.RTCHH[1:0] bits and the RTCAHUR.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 RTCAHUR.RTCHH[1:0]/RTCHL[3:0] bits while the RTCACTLL.RTCBSY bit = 1.

Bit 7 Reserved

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

The RTCAHUR.RTCMIH[2:0] bits and the RTCAHUR.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 RTCAHUR.RTCMIH[2:0]/RTCMIL[3:0] bits while the RTCACTLL.RTCBSY bit = 1.

RTCA Month/Day Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCAMON	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 RTCAMON.RTCMOH bit and the RTCAMON.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 RTCAMON.RTCMOH/RTCMOL[3:0] bits while the RTCACTLL.RTCBSY bit = 1.
- Be sure to avoid setting the RTCAMON.RTCMOH/RTCMOL[3:0] bits to 0x00.

Bits 7–6 Reserved

Bits 5–4 RTCDH[1:0]
Bits 3–0 RTCDL[3:0]

The RTCAMON.RTCDH[1:0] bits and the RTCAMON.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 RTCAMON.RTCDH[1:0]/RTCDL[3:0] bits while the RTCACTLL.RTCBSY bit = 1.

RTCA Year/Week Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCAYAR	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 10.6.2 lists the correspondence between the count value and day of the week.

Table 10.6.2 Correspondence between the count value and day of the week

RTCAYAR.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 RTCAYAR.RTCWK[2:0] bits while the RTCACTLL.RTCBSY bit = 1.

Bits 7–4 RTCYH[3:0]
Bits 3–0 RTCYL[3:0]

The RTCAYAR.RTCYH[3:0] bits and the RTCAYAR.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 RTCAYAR.RTCYH[3:0]/RTCYL[3:0] bits while the RTCACTLL. RTCBSY bit = 1.

RTCA Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCAINTF	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	T1DAYIF	0	H0	R/W	
	6	T1HURIF	0	H0	R/W	
	5	T1MINIF	0	H0	R/W	
	4	T1SECIF	0	H0	R/W	
	3	T1_2SECIF	0	H0	R/W	
	2	T1_4SECIF	0	H0	R/W	
	1	T1_8SECIF	0	H0	R/W	
	0	T1_32SECIF	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:

RTCAINTF.RTCTRMIF bit: Theoretical regulation completion interrupt
RTCAINTF.SW1IF bit: Stopwatch 1 Hz interrupt
RTCAINTF.SW10IF bit: Stopwatch 10 Hz interrupt
RTCAINTF.SW100IF bit: Stopwatch 100 Hz interrupt

Bits 11–9 Reserved

Bit 8 ALARMIF
Bit 7 T1DAYIF
Bit 6 T1HURIF
Bit 5 T1MINIF
Bit 4 T1SECIF
Bit 3 T1_2SECIF
Bit 2 T1_4SECIF
Bit 1 T1_8SECIF
Bit 0 T1_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:
RTCAINTF. ALARMIF bit: Alarm interrupt
RTCAINTF.T1DAYIF bit: 1-day interrupt
RTCAINTF.T1HURIF bit: 1-hour interrupt
RTCAINTF.T1MINIF bit: 1-minute interrupt
RTCAINTF.T1SECIF bit: 1-second interrupt
RTCAINTF.T1_2SECIF bit: 1/2-second interrupt
RTCAINTF.T1_4SECIF bit: 1/4-second interrupt
RTCAINTF.T1_8SECIF bit: 1/8-second interrupt
RTCAINTF.T1_32SECIF bit: 1/32-second interrupt

RTCA Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RTCAINTE	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	T1DAYIE	0	H0	R/W	
	6	T1HURIE	0	H0	R/W	
	5	T1MINIE	0	H0	R/W	
	4	T1SECIE	0	H0	R/W	
	3	T1_2SECIE	0	H0	R/W	
	2	T1_4SECIE	0	H0	R/W	
	1	T1_8SECIE	0	H0	R/W	
	0	T1_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:
RTCAINTE.RTCTRMIE bit: Theoretical regulation completion interrupt
RTCAINTE.SW1IE bit: Stopwatch 1 Hz interrupt
RTCAINTE.SW10IE bit: Stopwatch 10 Hz interrupt
RTCAINTE.SW100IE bit: Stopwatch 100 Hz interrupt

Bits 11–9 Reserved

Bit 8 ALARMIE
Bit 7 T1DAYIE
Bit 6 T1HURIE
Bit 5 T1MINIE
Bit 4 T1SECIE
Bit 3 T1_2SECIE
Bit 2 T1_4SECIE
Bit 1 T1_8SECIE
Bit 0 T1_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:
RTCAINTE.ALARMIE bit: Alarm interrupt
RTCAINTE.T1DAYIE bit: 1-day interrupt
RTCAINTE.T1HURIE bit: 1-hour interrupt
RTCAINTE.T1MINIE bit: 1-minute interrupt
RTCAINTE.T1SECIE bit: 1-second interrupt
RTCAINTE.T1_2SECIE bit: 1/2-second interrupt
RTCAINTE.T1_4SECIE bit: 1/4-second interrupt
RTCAINTE.T1_8SECIE bit: 1/8-second interrupt
RTCAINTE.T1_32SECIE bit: 1/32-second interrupt

11 Supply Voltage Detector (SVD4)

11.1 Overview

SVD4 is a supply voltage detector to monitor the V_{DD} voltage, or an external voltage detection input pin. The main features are listed below.

- Power supply voltage to be detected: Selectable from V_{DD} and external power sources (EXSVD*n*0, EXSVD*n*1) (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 11.1.1 shows the configuration of SVD4.

Table 11.1.1 SVD4 Configuration of S1C31W65

Item	S1C31W65
Number of channels	1 channel (Ch.0)
Power supply voltage to be detected	V_{DD} and two externally input voltages (EXSVD00, EXSVD01)
Detectable voltage level	V_{DD}: 32 levels (1.7 to 5.0 V)/external voltage: 32 levels (1.7 to 5.0 V)

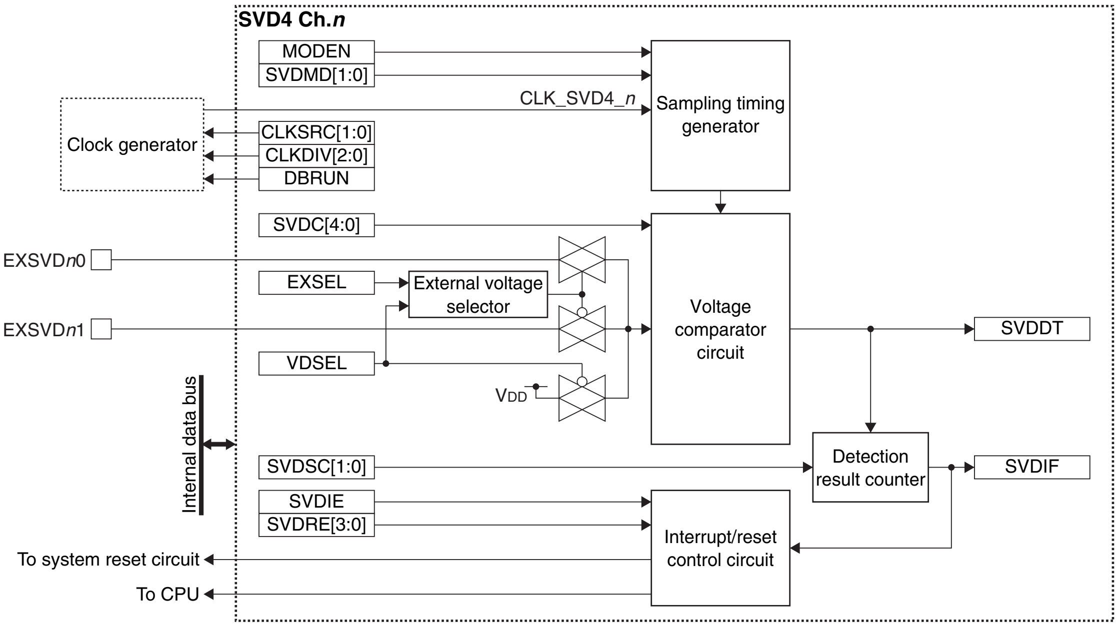


Figure 11.1.1 SVD4 Configuration

11.2 Input Pins and External Connection

11.2.1 Input Pins

Table 11.2.1.1 shows the SVD4 input pins.

Table 11.2.1.1 SVD4 Input Pins

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

* Indicates the status when the pin is configured for SVD4.

If the port is shared with the EXSVD*nx* pin and other functions, the EXSVD*nx* function must be assigned to the port before SVD4 Ch.*n* can be activated. For more information, refer to the "I/O Ports" chapter.

11.2.2 External Connection

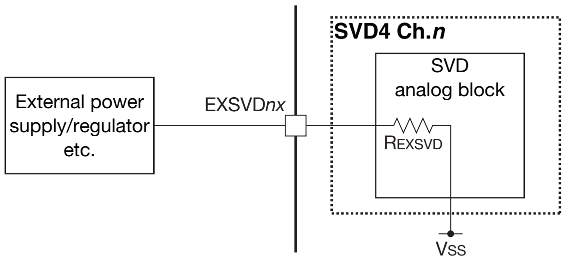


Figure 11.2.2.1 Connection between EXSVD*n* Pin and External Power Supply

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

11.3 Clock Settings

11.3.1 SVD4 Operating Clock

When using SVD4 Ch.*n*, the SVD4 operating clock CLK_SVD4_*n* must be supplied to SVD4 Ch.*n* from the clock generator.
The CLK_SVD4_*n* supply should be controlled as in the procedure shown below.

1. Write 0x0096 to the SYSPROT.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 SVD4_*n*CLK register bits:
   - SVD4_*n*CLK.CLKSRC[1:0] bits (Clock source selection)
   - SVD4_*n*CLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)
4. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

The CLK_SVD4_*n* frequency should be set to around 32 kHz.

11.3.2 Clock Supply in SLEEP Mode

When using SVD4 Ch.*n* during SLEEP mode, the SVD4 operating clock CLK_SVD4_*n* must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_SVD4_*n* clock source.
If the CLGOSC.*xxxx*SLPC bit for the CLK_SVD4_*n* clock source is 1, the CLK_SVD4_*n* clock source is deactivated during SLEEP mode and SVD4 Ch.*n* stops with the register settings maintained at those before entering SLEEP mode. After the CPU returns to normal mode, CLK_SVD4_*n* is supplied and the SVD4 Ch.*n* operation resumes.

11.3.3 Clock Supply in DEBUG Mode

The CLK_SVD4_*n* supply during DEBUG mode should be controlled using the SVD4_*n*CLK.DBRUN bit.
The CLK_SVD4_*n* supply to SVD4 Ch.*n* is suspended when the CPU enters DEBUG mode if the SVD4_*n*CLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_SVD4_*n* supply resumes. Although SVD4 Ch.*n* stops operating when the CLK_SVD4_*n* supply is suspended, the registers retain the status before DEBUG mode was entered. If the SVD4_*n*CLK.DBRUN bit = 1, the CLK_SVD4_*n* supply is not suspended and SVD4 Ch.*n* will keep operating in DEBUG mode.

11.4 Operations

11.4.1 SVD4 Control

Starting detection

SVD4 Ch.*n* should be initialized and activated with the procedure listed below.

1. Write 0x0096 to the SYSPROT.PROT[15:0] bits. (Remove system protection)
2. Configure the operating clock using the SVD4_*n*CLK.CLKSRC[1:0] and SVD4_*n*CLK.CLKDIV[2:0] bits.
3. Set the following SVD4_*n*CTL register bits:
   - SVD4_*n*CTL.VDSEL and SVD4_*n*CTL.EXSEL bits (Select detection voltage (V_{DD}, EXSVD*nx*))
   - SVD4_*n*CTL.SVDSC[1:0] bits (Set low power supply voltage detection counter)
   - SVD4_*n*CTL.SVDC[4:0] bits (Set SVD detection voltage V_{SVD}/EXSVD detection voltage V_{SVD_EXT})
   - SVD4_*n*CTL.SVDRE[3:0] bits (Select reset/interrupt mode)
   - SVD4_*n*CTL.SVDMD[1:0] bits (Set intermittent operation mode)
4. Set the following bits when using the interrupt:
   - Write 1 to the SVD4_*n*INTF.SVDIF bit. (Clear interrupt flag)
   - Set the SVD4_*n*INTE.SVDIE bit to 1. (Enable SVD4 Ch.*n* interrupt)
5. Set the SVD4_*n*CTL.MODEN bit to 1. (Enable SVD4 Ch.*n* detection)
6. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

Terminating detection

Follow the procedure shown below to stop SVD4 operation.

1. Write 0x0096 to the SYSPROT.PROT[15:0] bits. (Remove system protection)
2. Write 0 to the SVD4_*n*CTL.MODEN bit. (Disable SVD4 Ch.*n* detection)
3. Write a value other than 0x0096 to the SYSPROT.PROT[15:0] bits. (Set system protection)

Reading detection results

The following two detection results can be obtained by reading the SVD4_*n*INTF.SVDDT bit:

- When SVD4_*n*INTF.SVDDT bit = 0
  Power supply voltage (V_{DD}, EXSVD*nx*) ≥ SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}
- When SVD4_*n*INTF.SVDDT bit = 1
  Power supply voltage (V_{DD}, EXSVD*nx*) < SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}

Before reading the SVD4_*n*INTF.SVDDT bit, wait for at least SVD circuit enable response time after 1 is written to the SVD4_*n*CTL.MODEN bit (refer to "Supply Voltage Detector Characteristics, SVD circuit enable response time t_{SVDEN}" in the "Electrical Characteristics" chapter).
After the SVD4_*n*CTL.SVDC[4:0] bits setting value is altered to change the SVD detection voltage V_{SVD}/EXSVD detection voltage V_{SVD_EXT} when the SVD4_*n*CTL.MODEN bit = 1, wait for at least SVD circuit response time before reading the SVD4_*n*INTF.SVDDT bit (refer to "Supply Voltage Detector Characteristics, SVD circuit response time t_{SVD}" in the "Electrical Characteristics" chapter).

11.4.2 SVD4 Operations

Continuous operation mode

SVD4 Ch.*n* operates in continuous operation mode by default (SVD4_*n*CTL.SVDMD[1:0] bits = 0x0). In this mode, SVD4 Ch.*n* operates continuously while the SVD4_*n*CTL.MODEN bit is set to 1 and it keeps loading the detection results to the SVD4_*n*INTF.SVDDT bit. During this period, the current detection results can be obtained by reading the SVD4_*n*INTF.SVDDT bit as necessary. Furthermore, an interrupt (if the SVD4_*n*CTL.SVDRE[3:0] bits ≠ 0xa) or a reset (if the SVD4_*n*CTL.SVDRE[3:0] bits = 0xa) can be generated when the SVD4_*n*INTF.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

SVD4 Ch.*n* operates in intermittent operation mode when the SVD4_*n*CTL.SVDMD[1:0] bits are set to 0x1 to 0x3. In this mode, SVD4 Ch.*n* turns on at an interval set using the SVD4_*n*CTL.SVDMD[1:0] bits to perform detection operation and then it turns off while the SVD4_*n*CTL.MODEN bit is set to 1. During this period, the latest detection results can be obtained by reading the SVD4_*n*INTF.SVDDT bit as necessary. Furthermore, an interrupt or a reset can be generated when SVD4 Ch.*n* has successively detected low power supply voltage the number of times specified by the SVD4_*n*CTL.SVDSC[1:0] bits.

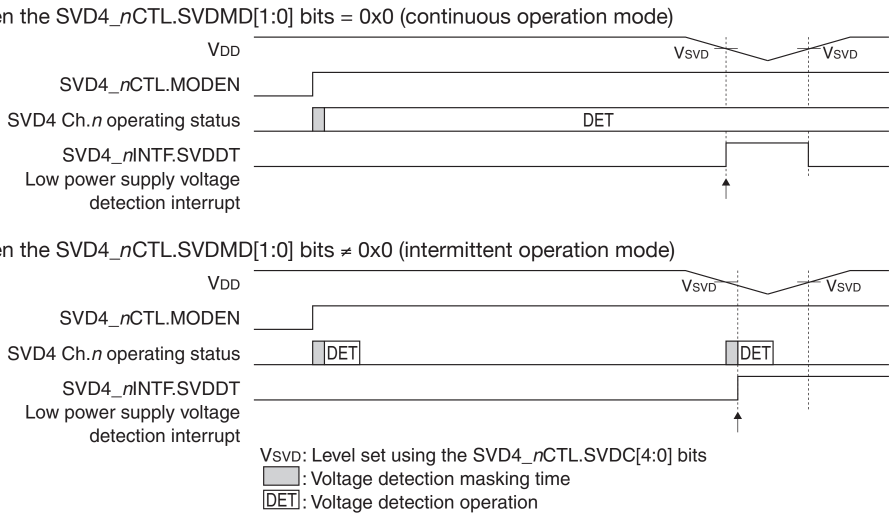


Figure 11.4.2.1 SVD4 Operations

11.5 SVD4 Interrupt and Reset

11.5.1 SVD4 Interrupt

Setting the SVD4_*n*CTL.SVDRE[3:0] bits to a value other than 0xa allows use of the low power supply voltage detection interrupt function.

Table 11.5.1.1 Low Power Supply Voltage Detection Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Low power supply voltage detection	SVD4_*n*INTF.SVDIF	In continuous operation mode When the SVD4_*n*INTF.SVDDT bit is 1 In intermittent operation mode When low power supply voltage is successively detected the specified number of times	Writing 1

SVD4 provides the interrupt enable bit (SVD4_*n*INTE.SVDIE bit) corresponding to the interrupt flag (SVD4_*n*INTF.SVDIF bit). An interrupt request is sent to the CPU only when the SVD4_*n*INTF.SVDIF bit is set while the interrupt is enabled by the SVD4_*n*INTE.SVDIE bit. For more information on interrupt control, refer to the "Interrupt" chapter.

Once the SVD4_*n*INTF.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 SVD4_*n*INTF.SVDDT bit in the interrupt handler routine.

11.5.2 SVD Reset

Setting the SVD4_*n*CTL.SVDRE[3:0] bits to 0xa allows use of the SVD reset issuance function.
The reset issuing timing is the same as that of the SVD4_*n*INTF.SVDIF bit being set when a low voltage is detected. After a reset has been issued, SVD4 Ch.*n* enters continuous operation mode even if it was operating in intermittent operation mode, and continues operating. Issuing an SVD reset initializes the port assignment. However, when EXSVD*nx* is being detected, the input of the port for the EXSVD*nx* pin is sent to SVD4 Ch.*n* so that SVD4 Ch.*n* will continue the EXSVD*nx* detection operation. If the power supply voltage reverts to the normal level, the SVD4_*n*INTF.SVDDT bit goes 0 and the reset state is canceled. After that, SVD4 Ch.*n* resumes operating in the operation mode set previously via the initialization routine. During reset state, the SVD4 Ch.*n* control bits are set as shown in Table 11.5.2.1.

Table 11.5.2.1 SVD4 Control Bits During Reset State

Control register	Control bit	Setting
SVD4_*n*CLK	DBRUN	Reset to the initial values.
	CLKDIV[2:0]	
	CLKSRC[1:0]	
SVD4_*n*CTL	VDSEL	The set value is retained.
	SVDSC[1:0]	Cleared to 0. (The set value becomes invalid as SVD4 Ch.*n* 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.
SVD4_*n*INTF	SVDIF	The status (1) before being reset is retained.
SVD4_*n*INTE	SVDIE	Cleared to 0.

11.6 Control Registers

SVD4 Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVD4_*n*CLK	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 operating clock is supplied to SVD4 Ch.*n* 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 SVD4 Ch.*n* operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of SVD4 Ch.*n*.

Table 11.6.1 Clock Source and Division Ratio Settings

SVD4_*n*CLK. CLKDIV[2:0] bits	SVD4_*n*CLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x7	Reserved	1/1	Reserved	1/1
0x6	1/1,024		1/1,024	
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.

SVD4 Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVD4_*n*CTL	15	VDSEL	0	H1	R/WP	–
	14–13	SVDSC[1:0]	0x0	H0	R/WP	Writing takes effect when the SVD4_ *n*CTL.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	EXSEL	0	H1	R/WP	
	2–1	SVDMD[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 SVD4 Ch.*n*.

1 (R/WP): Voltage applied to the EXSVD*nx* 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 (SVD4_*n*CTL.SVDMD[1:0] bits = 0x1 to 0x3).

Table 11.6.2 Interrupt/Reset Generating Condition in Intermittent Operation Mode

SVD4_*n*CTL.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 (SVD4_*n*CTL.SVDMD[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 11.6.3 Setting of SVD Detection Voltage V_{SVD}/EXSVD Detection Voltage V_{SVD_EXT}

SVD4_*n*CTL.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 SVD reset issuance function, refer to "SVD Reset."

Bit 3 **EXSEL**

This bit selects the voltage to be detected when the SVD4_*n*CTL.VDSEL bit = 1.

1 (R/WP): EXSVD*n*1

0 (R/WP): EXSVD*n*0

Bits 2–1 **SVDMD[1:0]**

These bits select intermittent operation mode and its detection cycle.

Table 11.6.4 Intermittent Operation Mode Detection Cycle Selection

SVD4_*n*CTL.SVDMD[1:0] bits	Operation mode (detection cycle)
0x3	Intermittent operation mode (CLK_SVD4_*n*/512)
0x2	Intermittent operation mode (CLK_SVD4_*n*/256)
0x1	Intermittent operation mode (CLK_SVD4_*n*/128)
0x0	Continuous operation mode

For more information on intermittent and continuous operation modes, refer to "SVD4 Operations."

Bit 0 **MODEN**

This bit enables/disables for the SVD4 Ch.*n* 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 SVD4_*n*CTL.MODEN bit resets the SVD4 Ch.*n* hardware. However, the register values set and the interrupt flag are not cleared. The SVD4_*n*CTL.MODEN bit is actually set to 0 after this processing has finished. If 1 is written to the SVD4_*n*CTL.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 SVD4 Ch.*n* internal circuit is initialized if the SVD4_*n*CTL.SVDSC[1:0] bits, SVD4_*n*CTL.SVDRE[3:0] bits, or SVD4_*n*CTL.SVDMD[1:0] bits are altered while SVD4 Ch.*n* is in operation after 1 is written to the SVD4_*n*CTL.MODEN bit.

SVD4 Ch.*n* Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVD4_*n*INTF	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}, EXSVD*nx*) < SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}

0 (R): Power supply voltage (V_{DD}, EXSVD*nx*) ≥ SVD detection voltage V_{SVD} or EXSVD detection voltage V_{SVD_EXT}

Bits 7–1 **Reserved**

Bit 0 SVDIF

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

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

Note: The SVD4 Ch.*n* internal circuit is initialized if the interrupt flag is cleared while SVD4 Ch.*n* is in operation after 1 is written to the SVD4_*n*CTL.MODEN bit.

SVD4 Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SVD4_*n*INTE	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 SVD4_*n*CTL.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.

12 16-bit Timers (T16)

12.1 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 12.1.1 shows the configuration of a T16 channel.

Table 12.1.1 T16 Channel Configuration of S1C31W65

Item	S1C31W65
Number of channels	8 channels (Ch.0–Ch.7)
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.6 → Synchronous serial interface Ch.1 master clock Ch.7 → 12-bit A/D converter trigger signal

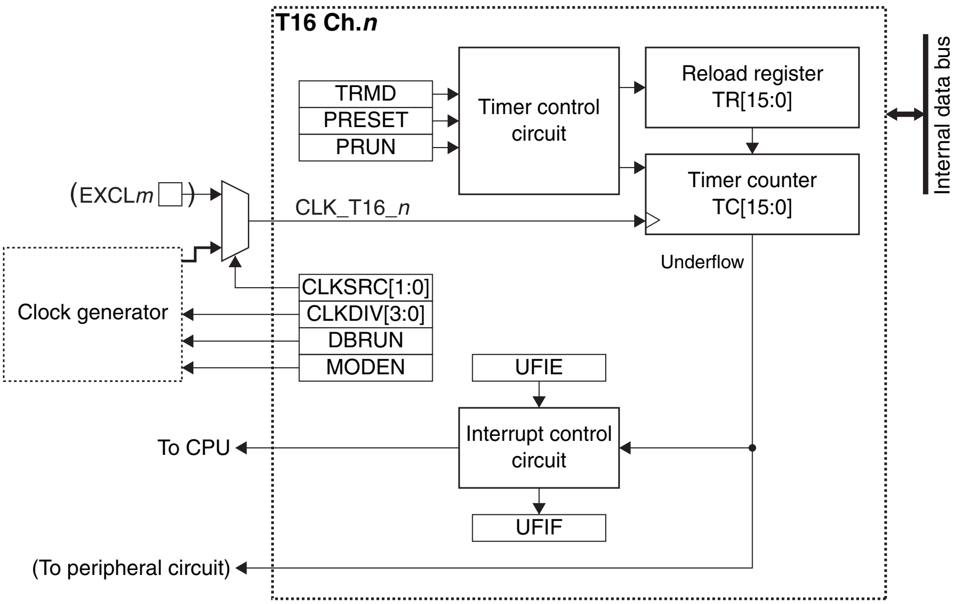


Figure 12.1.1 Configuration of a T16 Channel

12.2 Input Pin

Table 12.2.1 shows the T16 input pin.

Table 12.2.1 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. The EXCL*m* signal can be input through the chattering filter. For more information, refer to the “I/O Ports” chapter.

12.3 Clock Settings

12.3.1 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)

12.3.2 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.*xxxx*SLPC bit for the CLK_T16_*n* clock source.
If the CLGOSC.*xxxx*SLPC 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.

12.3.3 Clock Supply During Debugging

The CLK_T16_*n* supply during debugging 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 state if the T16_*n*CLK.DBRUN bit = 0. After the CPU returns to normal operation, 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 the debug state 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 a debug state.

12.3.4 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.

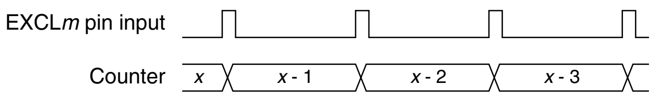


Figure 12.3.4.1 Count Down Timing

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

12.4 Operations

12.4.1 Initialization

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

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

6. Set the following T16_*n*CTL register bits:
   - Set the T16_*n*CTL.PRESET bit to 1. (Preset reload data to counter)
   - Set the T16_*n*CTL.PRUN bit to 1. (Start counting)

12.4.2 Counter Underflow

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

$$T = \frac{TR + 1}{f_{CLK\_T16\_n}} \qquad f_T = \frac{f_{CLK\_T16\_n}}{TR + 1} \qquad \text{(Eq. 12.1)}$$

Where
- T: Underflow cycle [s]
- f_T: Underflow frequency [Hz]
- TR: T16_*n*TR register setting
- $f_{CLK\_T16\_n}$: T16 Ch.*n* operating clock frequency [Hz]

12.4.3 Operations in Repeat Mode

T16 Ch.*n* enters repeat mode by setting the T16_*n*MOD.TRMD bit to 0.
In repeat mode, the count operation starts by writing 1 to the T16_*n*CTL.PRUN bit and continues until 0 is written. A counter underflow presets the T16_*n*TR register value to the counter, so underflow occurs periodically. Select this mode to generate periodic underflow interrupts or when using the timer to output a trigger/clock to the peripheral circuit.

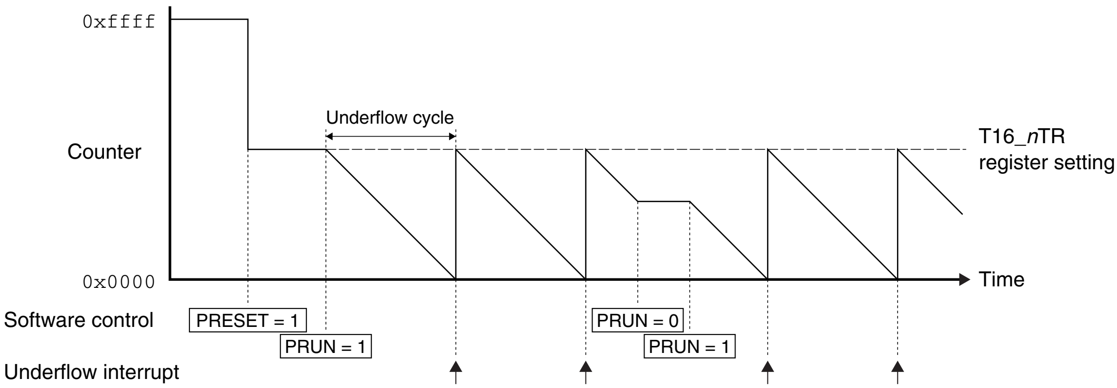


Figure 12.4.3.1 Count Operations in Repeat Mode

12.4.4 Operations in One-shot Mode

T16 Ch.*n* enters one-shot mode by setting the T16_*n*MOD.TRMD bit to 1.
In one-shot mode, the count operation starts by writing 1 to the T16_*n*CTL.PRUN bit and stops after the T16_*n*TR register value is preset to the counter when an underflow has occurred. At the same time the counter stops, the T16_*n*CTL.PRUN bit is cleared automatically. Select this mode to stop the counter after an interrupt has occurred once, such as for checking a specific lapse of time.

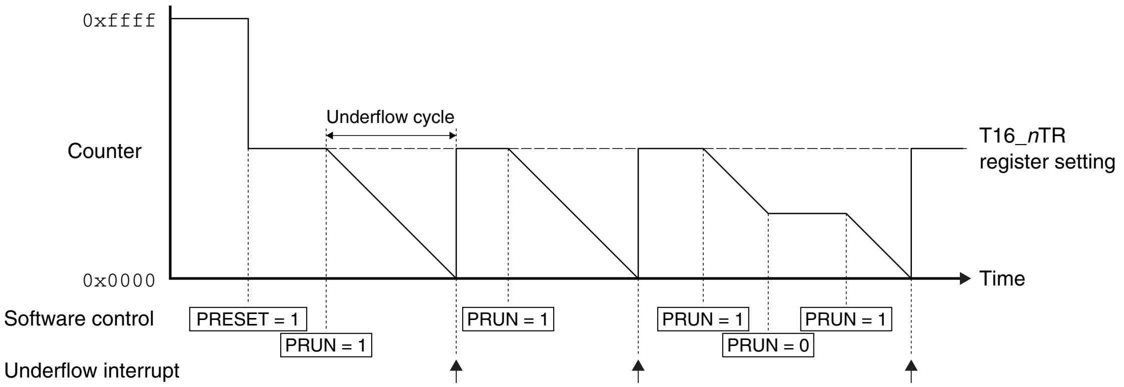


Figure 12.4.4.1 Count Operations in One-shot Mode

12.4.5 Counter Value Read

The counter value can be read out from the T16_*n*TC.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.

12.5 Interrupt

Each T16 channel has a function to generate the interrupt shown in Table 12.5.1.

Table 12.5.1 T16 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Underflow	T16_*n*INTF.UFIF	When the counter underflows	Writing 1

T16 provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the CPU 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” chapter.

12.6 Control Registers

T16 Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*CLK	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 during debugging or not.

1 (R/W): Clock supplied during debugging

0 (R/W): No clock supplied during debugging

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 12.6.1 Clock Source and Division Ratio Settings

T16_*n*CLK.CLKDIV[3:0] bits	T16_*n*CLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC/EXCL*m*
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_*n*CLK.CLKSRC[1:0] bits are set to 0x3, EXCL*m* is selected for the channel with an event counter function or EXOSC is selected for other channels.

T16 Ch.*n* Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16_*n*MOD	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_*n*CTL	15–9	–	0x00	–	R	–
	8	PRUN	0	H0	R/W	
	7–2	–	0x00	–	R	
	1	PRESET	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 PRUN

This bit starts/stops the timer.

1 (W): Start timer

0 (W): Stop timer

1 (R): Timer is running

0 (R): Timer is idle

By writing 1 to this bit, the timer starts count operations. However, the T16_*n*CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance. While the timer is running, writing 0 to 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_*n*TR 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_*n*TR register value to the counter. However, the T16_*n*CTL.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_*n*TR	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_*n*CTL.PRESET bit or when the counter underflows.

Notes:
- The T16_*n*TR register cannot be altered while the timer is running (T16_*n*CTL.PRUN bit = 1), as an incorrect initial value may be preset to the counter.
- When one-shot mode is set, the T16_*n*TR.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_*n*TC	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_*n*INTF	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_*n*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.

13 UART (UART3)

13.1 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.
- Can issue a DMA transfer request when a receive buffer one byte full or a transmit buffer empty occurs.
- 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 13.1.1 shows the UART3 configuration.

Table 13.1.1 UART3 Channel Configuration of S1C31W65

Item	S1C31W65
Number of channels	2 channels (Ch.0 and Ch.1)

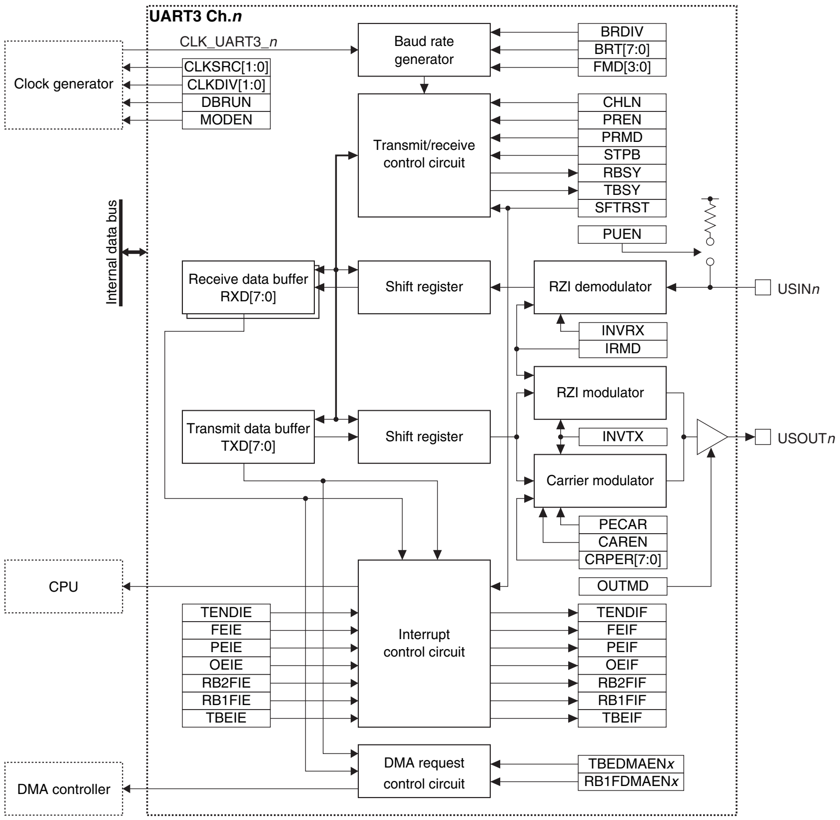


Figure 13.1.1 UART3 Configuration

13.2 Input/Output Pins and External Connections

13.2.1 List of Input/Output Pins

Table 13.2.1.1 lists the UART3 pins.

Table 13.2.1.1 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.

13.2.2 External Connections

Figure 13.2.2.1 shows a connection diagram between the UART3 in this IC and an external UART device.

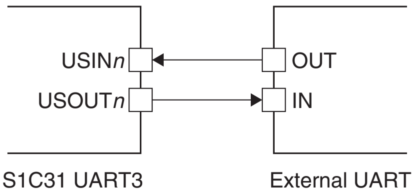


Figure 13.2.2.1 Connections between UART3 and an External UART Device

13.2.3 Input Pin Pull-Up Function

The UART3 includes a pull-up resistor for the USIN*n* pin. Setting the UART3_*n*MOD.PUEN bit to 1 enables the resistor to pull up the USIN*n* pin.

13.2.4 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 UART3_*n*MOD.OUTMD bit to 1.

13.2.5 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 UART3_*n*MOD.INVRX bit and the UART3_*n*MOD.INVTX bit, respectively, to 1.

Note: Unless otherwise specified, this chapter shows input/output signals with non-inverted waveforms (UART3_*n*MOD.INVRX bit = 0, UART3_*n*MOD.INVTX bit =0).

13.3 Clock Settings

13.3.1 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 UART3_*n*CLK register bits:
   - UART3_*n*CLK.CLKSRC[1:0] bits (Clock source selection)
   - UART3_*n*CLK.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.

13.3.2 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.*xxxx*SLPC bit for the CLK_UART3_*n* clock source.

13.3.3 Clock Supply During Debugging

The CLK_UART3_*n* supply during debugging should be controlled using the UART3_*n*CLK.DBRUN bit.
The CLK_UART3_*n* supply to the UART3 Ch.*n* is suspended when the CPU enters debug state if the UART3_*n*CLK.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 the debug state was entered. If the UART3_*n*CLK.DBRUN bit = 1, the CLK_UART3_*n* supply is not suspended and the UART3 Ch.*n* will keep operating in a debug state.

13.3.4 Baud Rate Generator

The UART3 includes a baud rate generator to generate the transfer (sampling) clock. The transfer rate is determined by the UART3_*n*MOD.BRDIV, UART3_*n*BR.BRT[7:0], and UART3_*n*BR.FMD[3:0] bit settings. Use the following equations to calculate the setting values for obtaining the desired transfer rate.

$$\mathrm{bps} = \frac{\mathrm{CLK\_UART3}}{\frac{\mathrm{BRT}+1}{\mathrm{BRDIV}} + \mathrm{FMD}} \qquad \mathrm{BRT} = \mathrm{BRDIV} \times \left(\frac{\mathrm{CLK\_UART3}}{\mathrm{bps}} - \mathrm{FMD}\right) - 1 \qquad \text{(Eq. 13.1)}$$

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 UART3_*n*MOD.BRDIV bit
BRT: UART3_*n*BR.BRT[7:0] setting value (0 to 255)
FMD: UART3_*n*BR.FMD[3:0] setting value (0 to 15)

For the transfer rate range configurable in the UART3, refer to "UART Characteristics, Transfer baud rates U_{BRT1} and U_{BRT2}" in the "Electrical Characteristics" chapter.

13.4 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 UART3_*n*MOD.CHLN bit, the data length can be set to seven bits (UART3_*n*MOD.CHLN bit = 0) or eight bits (UART3_*n*MOD.CHLN bit = 1).

Stop bit length

With the UART3_*n*MOD.STPB bit, the stop bit length can be set to one bit (UART3_*n*MOD.STPB bit = 0) or two bits (UART3_*n*MOD.STPB bit = 1).

Parity function

The parity function is configured using the UART3_*n*MOD.PREN and UART3_*n*MOD.PRMD bits.

Table 13.4.1 Parity Function Setting

UART3_*n*MOD.PREN bit	UART3_*n*MOD.PRMD bit	Parity function
1	1	Odd parity
1	0	Even parity
0	*	Non parity

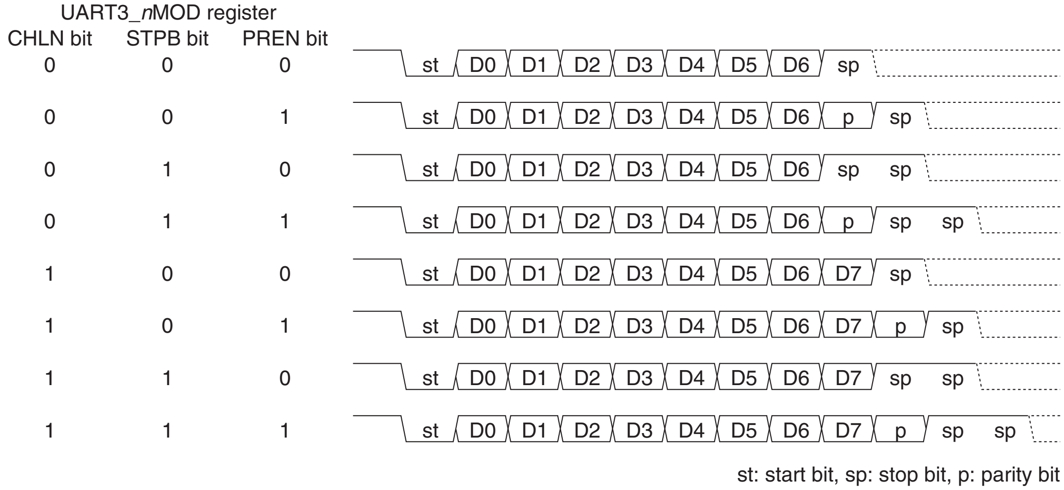


Figure 13.4.1 Data Format

13.5 Operations

13.5.1 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 UART3_*n*CLK.CLKSRC[1:0] and UART3_*n*CLK.CLKDIV[1:0] bits. (Configure operating clock)
3. Configure the following UART3_*n*MOD register bits:
   - UART3_*n*MOD.BRDIV bit (Select baud rate division ratio (1/16 or 1/4))
   - UART3_nMOD.INVRX bit (Enable/disable USIN*n* input signal inversion)
   - UART3_nMOD.INVTX bit (Enable/disable USOUT*n* output signal inversion)
   - UART3_*n*MOD.PUEN bit (Enable/disable USIN*n* pin pull-up)
   - UART3_*n*MOD.OUTMD bit (Enable/disable USOUT*n* pin open-drain output)
   - UART3_*n*MOD.IRMD bit (Enable/disable IrDA interface)
   - UART3_*n*MOD.CHLN bit (Set data length (7 or 8 bits))
   - UART3_*n*MOD.PREN bit (Enable/disable parity function)
   - UART3_*n*MOD.PRMD bit (Select parity mode (even or odd))
   - UART3_*n*MOD.STPB bit (Set stop bit length (1 or 2 bits))
   - UART3_*n*MOD.CAREN bit (Enable/disable carrier modulation function)
   - UART3_nMOD.PECAR bit (Select carrier modulation period (H data period/L data period))
4. Set the UART3_*n*BR.BRT[7:0] and UART3_*n*BR.FMD[3:0] bits. (Set transfer rate)
5. Set the UART3_*n*CAWF.CRPER[7:0] bits. (Set carrier cycle)
6. Set the following UART3_*n*CTL register bits:
   - Set the UART3_*n*CTL.SFTRST bit to 1. (Execute software reset)
   - Set the UART3_*n*CTL.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 UART3_*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the UART3_*n*INTE register to 1. * (Enable interrupts)

   * The initial value of the UART3_*n*INTF.TBEIF bit is 1, therefore, an interrupt will occur immediately after the UART3_*n*INTE.TBEIE bit is set to 1.
8. Configure the DMA controller and set the following UART3 control bits when using DMA transfer:
   - Write 1 to the DMA transfer request enable bits in the UART3_*n*TBEDMAEN and UART3_*n*RB1FDMAEN registers. (Enable DMA transfer requests)

13.5.2 Data Transmission

A data sending procedure and the UART3 Ch.*n* operations 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. Check to see if the UART3_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
2. Write transmit data to the UART3_*n*TXD 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 UART3_*n*TXD register. The transmit data in the UART3_*n*TXD register is automatically transferred to the shift register and the UART3_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
The USOUT*n* pin outputs a start bit and the UART3_*n*INTF.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 UART3_*n*TXD register after making sure the UART3_*n*INTF.TBEIF bit is set to 1.

If no transmit data remains in the UART3_*n*TXD register after the stop bit has been output from the USOUT*n* pin, the UART3_*n*INTF.TBSY bit is cleared to 0 and the UART3_*n*INTF.TENDIF bit is set to 1 (transmission completed).

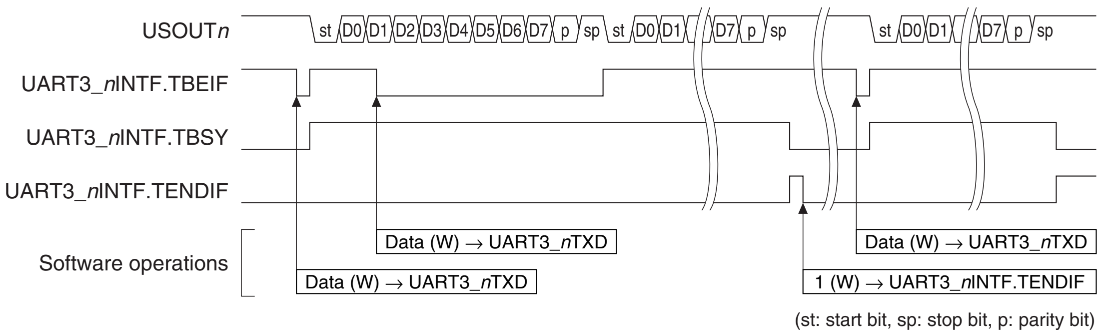


Figure 13.5.2.1 Example of Data Sending Operations

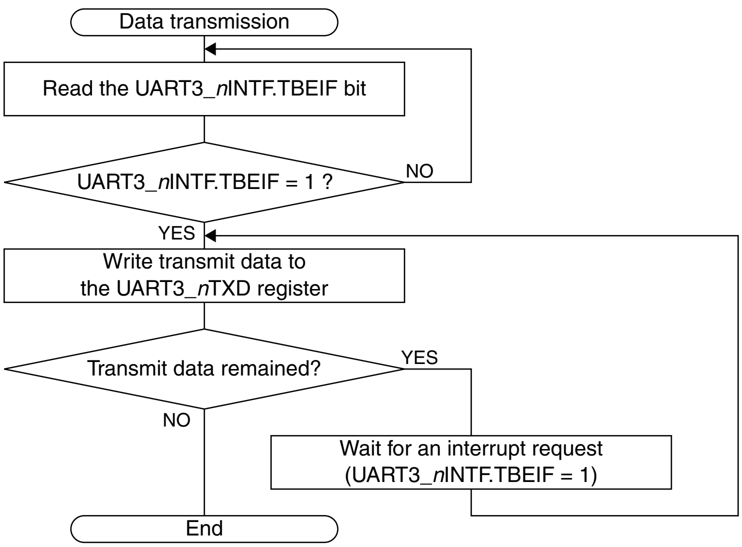


Figure 13.5.2.2 Data Transmission Flowchart

Data transmission using DMA

By setting the UART3_*n*TBEDMAEN.TBEDMAEN*x* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and transmit data is transferred from the specified memory to the UART3_*n*TXD register via DMA Ch.*x* when the UART3_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty). This automates the data sending procedure described above.

The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance so that transmit data will be transferred to the UART3_*n*TXD register. For more information on DMA, refer to the "DMA Controller" chapter.

Table 13.5.2.1 DMA Data Structure Configuration Example (for Data Transmission)

Item		Setting example
End pointer	Transfer source	Memory address in which the last transmit data is stored
	Transfer destination	UART3_*n*TXD register address
Control data	dst_inc	0x3 (no increment)
	dst_size	0x0 (byte)
	src_inc	0x0 (+1)
	src_size	0x0 (byte)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

13.5.3 Data Reception

A data receiving procedure and the UART3 Ch.*n* operations are shown below. Figures 13.5.3.1 and 13.5.3.2 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 UART3_*n*INTF.RB1FIF bit is set to 1 (receive buffer one byte full).
3. Read the received data from the UART3_*n*RXD 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 UART3_*n*INTF.RB2FIF bit is set to 1 (receive buffer two bytes full).
3. Read the received data from the UART3_*n*RXD 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 UART3_*n*INTF.RBSY bit is set to 1 when the start bit is detected.

The UART3_*n*INTF.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 UART3_*n*INTF.RB1FIF bit to 1 (receive buffer one byte full). If the second data is received without reading the first data, the UART3_*n*INTF.RB2FIF bit is set to 1 (receive buffer two bytes full).

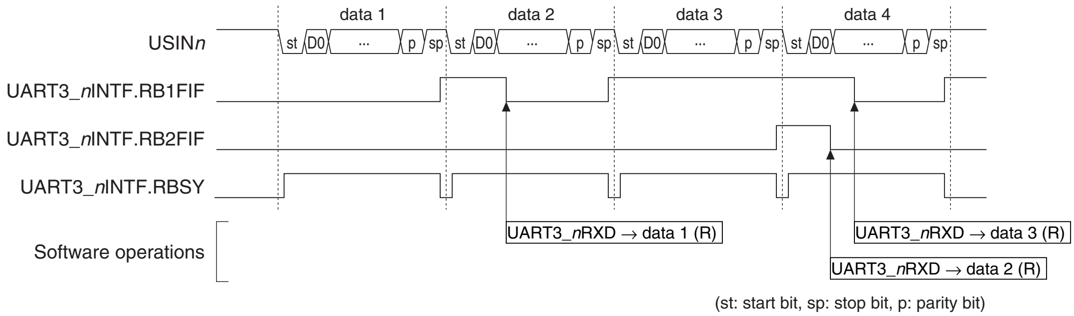


Figure 13.5.3.1 Example of Data Receiving Operations

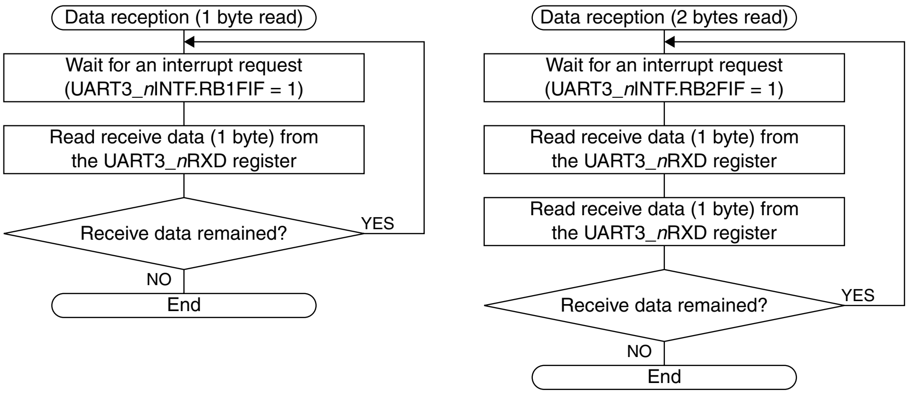


Figure 13.5.3.2 Data Reception Flowcharts

Data reception using DMA

By setting the UART3_*n*RB1FDMAEN.RB1FDMAEN*x* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and the received data is transferred from the UART3_*n*RXD register to the specified memory via DMA Ch.*x* when the UART3_*n*INTF.RB1FIF bit is set to 1 (receive buffer one byte full).

This automates the procedure (read by one byte) described above.

The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance. For more information on DMA, refer to the "DMA Controller" chapter.

Table 13.5.3.1 DMA Data Structure Configuration Example (for Data Reception)

Item		Setting example
End pointer	Transfer source	UART3_*n*RXD register address
	Transfer destination	Memory address to which the last received data is stored
Control data	dst_inc	0x0 (+1)
	dst_size	0x0 (byte)
	src_inc	0x3 (no increment)
	src_size	0x0 (byte)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

13.5.4 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 UART3_*n*MOD.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.

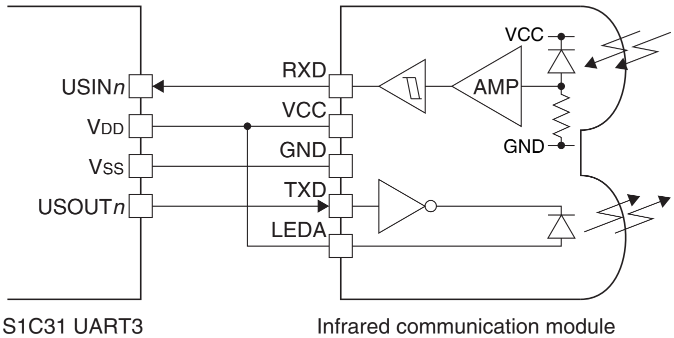


Figure 13.5.4.1 Example of Connections with an Infrared Communication Module

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.

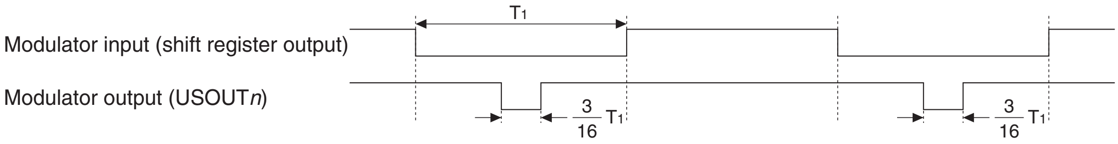


Figure 13.5.4.2 IrDA Transmission Signal Waveform

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.

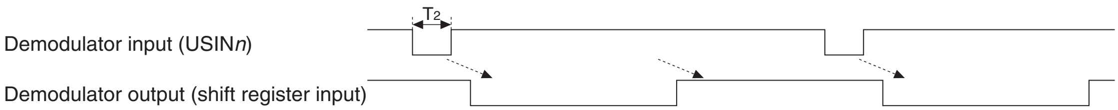


Figure 13.5.4.3 IrDA Receive Signal Waveform

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_*n* × 3 cycles or longer.

13.5.5 Carrier Modulation

The UART3 has a carrier modulation function.
Writing 1 to the UART3_*n*MOD.CAREN bit enables the carrier modulation function allowing carrier modulation waveforms to be output according to the UART3_*n*MOD.PECAR bit setting. Data transmit control is identical to that for normal interface even in this case.

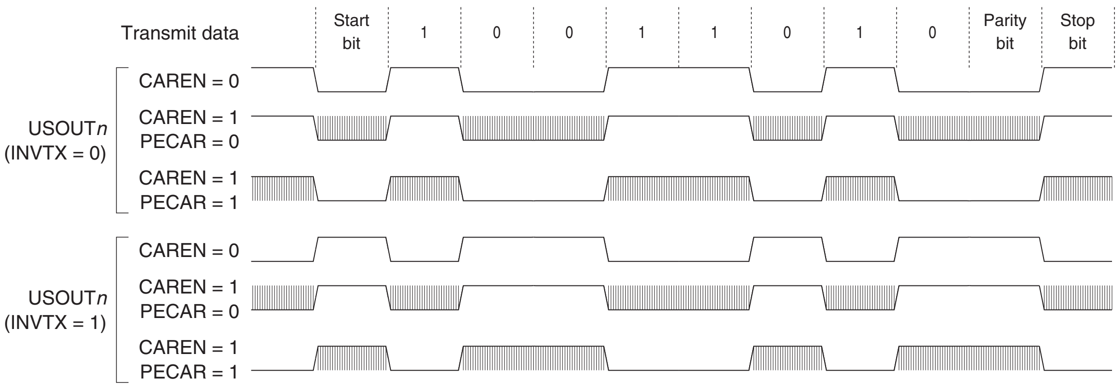


Figure 13.5.5.1 Carrier Modulation Waveform
(UART3_*n*MOD.CHLN = 1, UART3_*n*MOD.STPB = 0, UART3_*n*MOD.PREN = 1)

The carrier modulation output frequency is determined by the UART3_*n*CAWF.CRPER[7:0] bit settings. Use the following equations to calculate the setting values for obtaining the desired frequency.

$$\text{Carrier modulation output frequency} = \frac{\text{CLK\_UART3}}{(\text{CRPER} + 1) \times 2} \text{ [Hz]} \qquad \text{(Eq. 13.2)}$$

Where

CLK_UART3: UART3 operating clock frequency [Hz]

CRPER: UART3_*n*CAWF.CRPER[7:0] setting value (0 to 255)

13.6 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.

13.6.1 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 UART3_*n*INTF.FEIF bit (framing error interrupt flag) is set to 1 when the data becomes ready to read from the UART3_*n*RXD 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.

13.6.2 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 UART3_*n*INTF.PEIF bit (parity error interrupt flag) is set to 1 when the data becomes ready to read from the UART3_*n*RXD register (see the Note on framing error).

13.6.3 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 UART3_*n*INTF.OEIF bit (overrun error interrupt flag) is set to 1.

13.7 Interrupts

The UART3 has a function to generate the interrupts shown in Table 13.7.1.

Table 13.7.1 UART3 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
End of transmission	UART3_*n*INTF.TENDIF	When the UART3_*n*INTF.TBEIF bit = 1 after the stop bit has been sent	Writing 1 or software reset
Framing error	UART3_*n*INTF.FEIF	Refer to the "Receive Errors."	Writing 1, reading received data that encountered an error, or software reset
Parity error	UART3_*n*INTF.PEIF	Refer to the "Receive Errors."	Writing 1, reading received data that encountered an error, or software reset
Overrun error	UART3_*n*INTF.OEIF	Refer to the "Receive Errors."	Writing 1 or software reset
Receive buffer two bytes full	UART3_*n*INTF.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	UART3_*n*INTF.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	UART3_*n*INTF.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 CPU 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" chapter.

13.8 DMA Transfer Requests

The UART3 has a function to generate DMA transfer requests from the causes shown in Table 13.8.1.

Table 13.8.1 DMA Transfer Request Causes of UART3

Cause to request DMA transfer	DMA transfer request flag	Set condition	Clear condition
Receive buffer one byte full	Receive buffer one byte full flag (UART3_*n*INTF.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	Transmit buffer empty flag (UART3_*n*INTF.TBEIF)	When transmit data written to the transmit data buffer is transferred to the shift register	Writing transmit data

The UART3 provides DMA transfer request enable bits corresponding to each DMA transfer request flag shown above for the number of DMA channels. A DMA transfer request is sent to the pertinent channel of the DMA controller only when the DMA transfer request flag, of which DMA transfer has been enabled by the DMA transfer request enable bit, is set. The DMA transfer request flag also serves as an interrupt flag, therefore, both the DMA transfer request and the interrupt cannot be enabled at the same time. After a DMA transfer has completed, disable the DMA transfer to prevent unintended DMA transfer requests from being issued. For more information on the DMA control, refer to the "DMA Controller" chapter.

13.9 Control Registers

UART3 Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*n*CLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–6	–	0x0	–	R	
	5–4	CLKDIV[1:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

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

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

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

Bits 7–6 Reserved

Bits 5–4 CLKDIV[1:0]

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

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the UART3.

Table 13.9.1 Clock Source and Division Ratio Settings

UART3_*n*CLK. CLKDIV[1:0] bits	UART3_*n*CLK.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 UART3_*n*CLK register settings can be altered only when the UART3_*n*CTL.MODEN bit = 0.

UART3 Ch.*n* Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*n*MOD	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 UART3_*n*MOD register settings can be altered only when the UART3_*n*CTL.MODEN bit = 0.
- Do not set both the UART3_*n*MOD.IRMD and UART3_*n*MOD.CAREN bits simultaneously.

UART3 Ch.*n* Baud–Rate Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*n*BR	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 UART3_*n*BR register settings can be altered only when the UART3_*n*CTL.MODEN bit = 0.
- Do not set the UART3_*n*BR.FMD[3:0] bits to a value other than 0 to 3 when the UART3_*n*MOD.BRDIV bit = 1.

UART3 Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*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 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 UART3_*n*CTL.MODEN bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the UART3_*n*CTL.MODEN bit to 1 again after that, be sure to write 1 to the UART3_*n*CTL.SFTRST bit as well.

UART3 Ch.*n* Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*n*TXD	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 UART3_*n*INTF. 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
UART3_*n*RXD	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
UART3_*n*INTF	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 UART3_*n*RXD 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 UART3_*n*RXD register.
	1	RB1FIF	0	H0/S0	R	
	0	TBEIF	1	H0/S0	R	Cleared by writing to the UART3_*n*TXD register.

Bits 15–10 Reserved

Bit 9 RBSY

This bit indicates the receiving status. (See Figure 13.5.3.1.)

1 (R): During receiving

0 (R): Idle

Bit 8 TBSY

This bit indicates the sending status. (See Figure 13.5.2.1.)

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:
UART3_*n*INTF.TENDIF bit: End-of-transmission interrupt
UART3_*n*INTF.FEIF bit: Framing error interrupt
UART3_*n*INTF.PEIF bit: Parity error interrupt
UART3_*n*INTF.OEIF bit: Overrun error interrupt
UART3_*n*INTF.RB2FIF bit: Receive buffer two bytes full interrupt
UART3_*n*INTF.RB1FIF bit: Receive buffer one byte full interrupt
UART3_*n*INTF.TBEIF bit: Transmit buffer empty interrupt

UART3 Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*n*INTE	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6	TENDIE	0	H0	R/W	
	5	FEIE	0	H0	R/W	
	4	PEIE	0	H0	R/W	
	3	OEIE	0	H0	R/W	
	2	RB2FIE	0	H0	R/W	
	1	RB1FIE	0	H0	R/W	
	0	TBEIE	0	H0	R/W	

Bits 15–7 Reserved

Bit 6 TENDIE
Bit 5 FEIE
Bit 4 PEIE
Bit 3 OEIE
Bit 2 RB2FIE
Bit 1 RB1FIE
Bit 0 TBEIE

These bits enable UART3 interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

The following shows the correspondence between the bit and interrupt:
UART3_*n*INTE.TENDIE bit: End-of-transmission interrupt
UART3_*n*INTE.FEIE bit: Framing error interrupt
UART3_*n*INTE.PEIE bit: Parity error interrupt
UART3_*n*INTE.OEIE bit: Overrun error interrupt
UART3_*n*INTE.RB2FIE bit: Receive buffer two bytes full interrupt
UART3_*n*INTE.RB1FIE bit: Receive buffer one byte full interrupt
UART3_*n*INTE.TBEIE bit: Transmit buffer empty interrupt

UART3 Ch.*n* Transmit Buffer Empty DMA Request Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*n*T BEDMAEN	15–0	TBEDMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 TBEDMAEN[15:0]

These bits enable the UART3 to issue a DMA transfer request to the corresponding DMA controller channel (Ch.0–Ch.15) when a transmit buffer empty state has occurred.

1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

UART3 Ch.*n* Receive Buffer One Byte Full DMA Request Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*n* RB1FDMAEN	15–0	RB1FDMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 RB1FDMAEN[15:0]

These bits enable the UART3 to issue a DMA transfer request to the corresponding DMA controller channel (Ch.0–Ch.15) when a receive buffer one byte full state has occurred.

1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

UART3 Ch.*n* Carrier Waveform Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
UART3_*n*CAWF	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."

14 Synchronous Serial Interface (SPIA)

14.1 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.
- Can issue a DMA transfer request when a receive buffer full or a transmit buffer empty occurs.
- 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 14.1.1 shows the SPIA configuration.

Table 14.1.1 SPIA Channel Configuration of S1C31W65

Item	S1C31W65
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.6

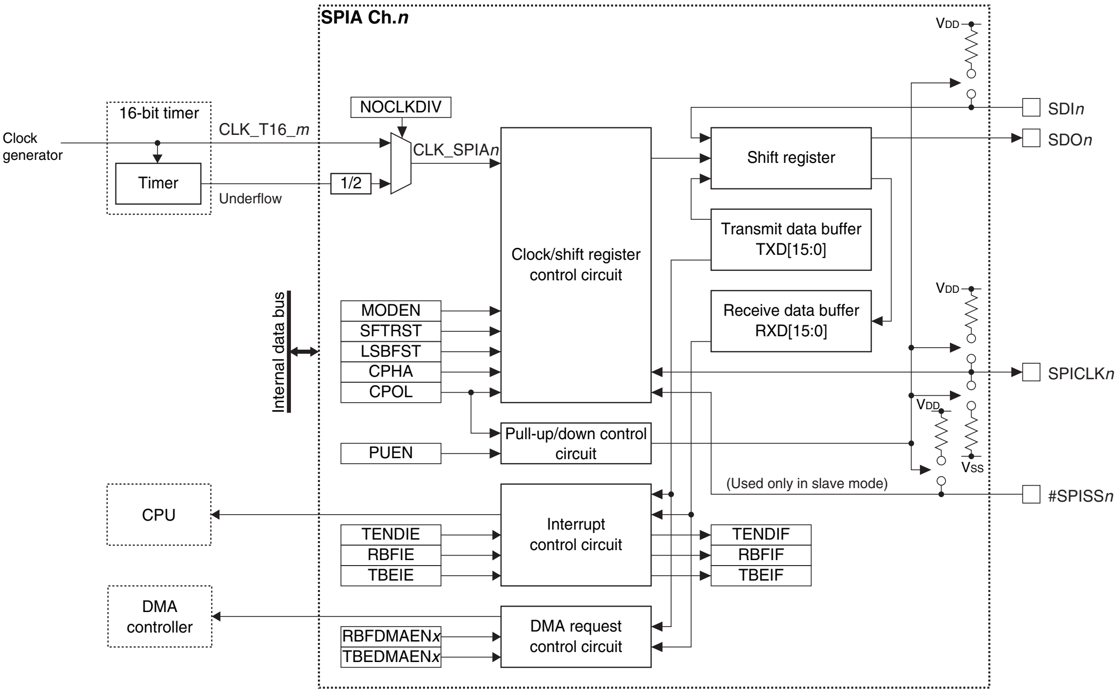


Figure 14.1.1 SPIA Configuration

14.2 Input/Output Pins and External Connections

14.2.1 List of Input/Output Pins

Table 14.2.1.1 lists the SPIA pins.

Table 14.2.1.1 List of SPIA Pins

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

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

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

14.2.2 External Connections

SPIA operates in master mode or slave mode. Figures 14.2.2.1 and 14.2.2.2 show connection diagrams between SPIA in each mode and external SPI devices.

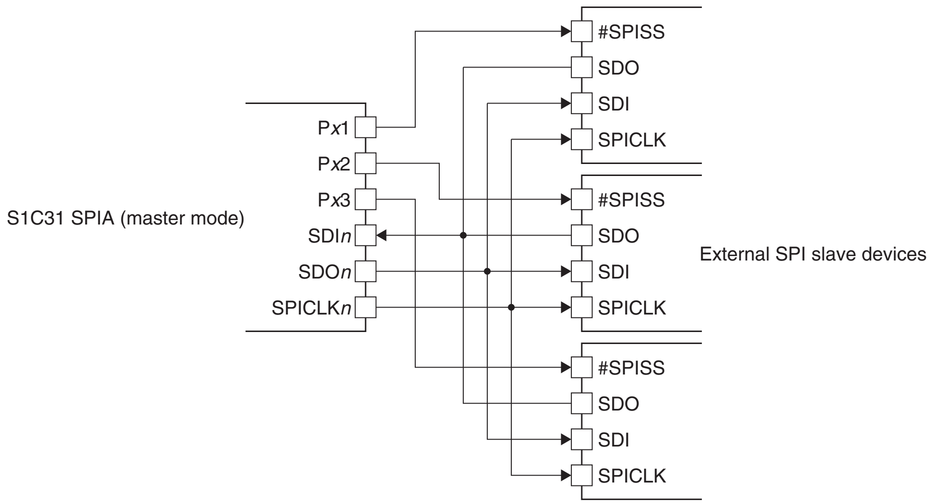


Figure 14.2.2.1 Connections between SPIA in Master Mode and External SPI Slave Devices

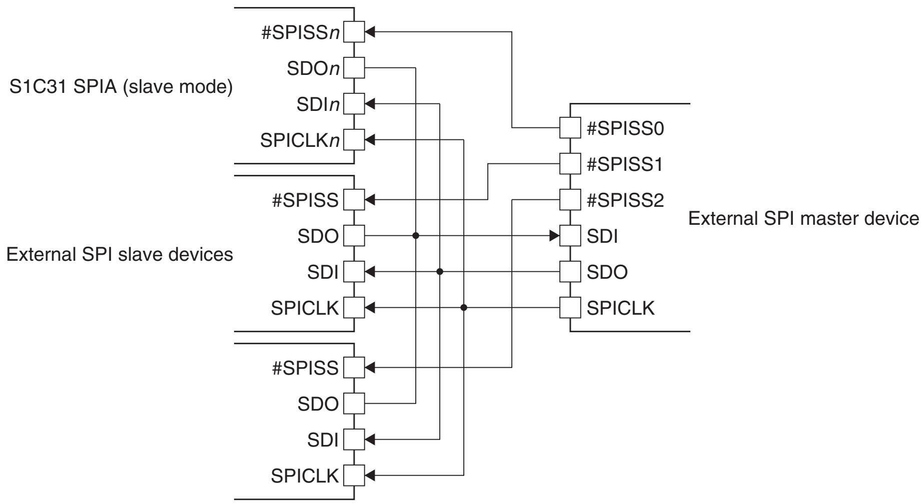


Figure 14.2.2.2 Connections between SPIA in Slave Mode and External SPI Master Device

14.2.3 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 14.2.3.1.

Table 14.2.3.1 Pin Function Differences between Modes

Pin	Function in master mode	Function in slave mode
SDI*n*	Always placed into input state.	
SDO*n*	Always placed into output state.	This pin is placed into output state while a low level is applied to the #SPISS*n* pin or placed into Hi-Z state while a high level is applied to the #SPISS*n* 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.
#SPISS*n*	Not used. This input function is not required to be assigned to the port. To output the slave select signal in master mode, use a general-purpose I/O port function.	Applying a low level to the #SPISS*n* 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 SDI*n* 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.

14.2.4 Input Pin Pull-Up/Pull-Down Function

The SPIA input pins (SDI*n* in master mode or SDI*n*, SPICLK*n*, and #SPISS*n* pins in slave mode) have a pull-up or pull-down function as shown in Table 14.2.4.1. This function is enabled by setting the SPIA_*n*MOD.PUEN bit to 1.

Table 14.2.4.1 Pull-Up or Pull-Down of Input Pins

Pin	Master mode	Slave mode
SDI*n*	Pull-up	Pull-up
SPICLK*n*	–	SPIA_*n*MOD.CPOL bit = 1: Pull-up SPIA_*n*MOD.CPOL bit = 0: Pull-down
#SPISS*n*	–	Pull-up

14.3 Clock Settings

14.3.1 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 SPIA_*n*MOD.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_*m*CTL.MODEN and T16_*m*CTL.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 SPIA_*n*MOD.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.

$$f_{SPICLK} = \frac{f_{CLK\_SPIA}}{2 \times (RLD + 1)} \qquad RLD = \frac{f_{CLK\_SPIA}}{f_{SPICLK} \times 2} - 1 \qquad \text{(Eq. 14.1)}$$

Where

f_{SPICLK}: SPICLK*n* frequency [Hz] (= baud rate [bps])

f_{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 SPIA_*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.

14.3.2 Clock Supply During Debugging

In master mode, the operating clock supply during debugging should be controlled using the T16_*m*CLK.DBRUN bit.

The CLK_T16_*m* supply to SPIA Ch.*n* is suspended when the CPU enters debug state if the T16_*m*CLK.DBRUN bit = 0. After the CPU returns to normal operation, 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 the debug state 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 a debug state.

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 state or normal operation state.

14.3.3 SPI Clock (SPICLK*n*) Phase and Polarity

The SPICLK*n* phase and polarity can be configured separately using the SPIA_*n*MOD.CPHA bit and the SPIA_*n*MOD.CPOL bit, respectively. Figure 14.3.3.1 shows the clock waveform and data input/output timing in each setting.

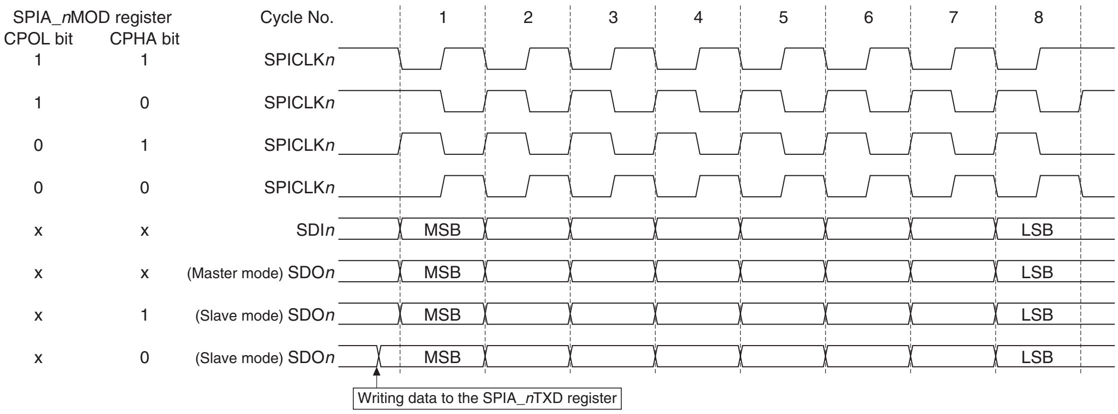


Figure 14.3.3.1 SPI Clock Phase and Polarity (SPIA_*n*MOD.LSBFST bit = 0, SPIA_*n*MOD.CHLN[3:0] bits = 0x7)

14.4 Data Format

The SPIA data length can be selected from 2 bits to 16 bits by setting the SPIA_*n*MOD.CHLN[3:0] bits. The input/output permutation is configurable to MSB first or LSB first using the SPIA_*n*MOD.LSBFST bit. Figure 14.4.1 shows a data format example when the SPIA_*n*MOD.CHLN[3:0] bits = 0x7, the SPIA_*n*MOD.CPOL bit = 0 and the SPIA_*n*MOD.CPHA bit = 0.

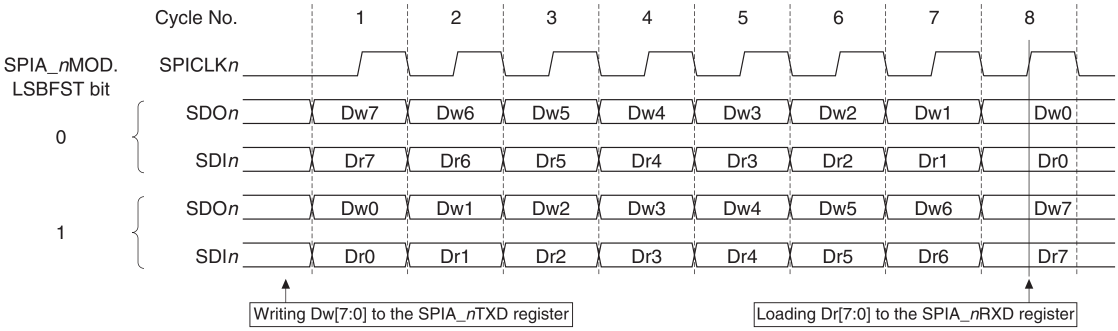


Figure 14.4.1 Data Format Selection Using the SPIA_*n*MOD.LSBFST Bit
(SPIA_*n*MOD.CHLN[3:0] bits = 0x7, SPIA_*n*MOD.CPOL bit = 0, SPIA_*n*MOD.CPHA bit = 0)

14.5 Operations

14.5.1 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 SPIA_*n*MOD register bits:
   - SPIA_*n*MOD.PUEN bit (Enable input pin pull-up/down)
   - SPIA_*n*MOD.NOCLKDIV bit (Select master mode operating clock)
   - SPIA_*n*MOD.LSBFST bit (Select MSB first/LSB first)
   - SPIA_*n*MOD.CPHA bit (Select clock phase)
   - SPIA_*n*MOD.CPOL bit (Select clock polarity)
   - SPIA_*n*MOD.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 SPIA_*n*CTL register bits:
   - Set the SPIA_*n*CTL.SFTRST bit to 1. (Execute software reset)
   - Set the SPIA_*n*CTL.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 SPIA_*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the SPIA_*n*INTE register to 1. * (Enable interrupts)

   * The initial value of the SPIA_*n*INTF.TBEIF bit is 1, therefore, an interrupt will occur immediately after the SPIA_*n*INTE.TBEIE bit is set to 1.
6. Configure the DMA controller and set the following SPIA control bits when using DMA transfer:
   - Write 1 to the DMA transfer request enable bits in the SPIA_*n*TBEDMAEN and SPIA_*n*RBFDMAEN registers. (Enable DMA transfer requests)

14.5.2 Data Transmission in Master Mode

A data sending procedure and operations in master mode are shown below. Figures 14.5.2.1 and 14.5.2.2 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 SPIA_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
3. Write transmit data to the SPIA_*n*TXD 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 SPIA_*n*TXD register.
The transmit data in the SPIA_*n*TXD register is automatically transferred to the shift register and the SPIA_*n*INTF.TBEIF bit is set to 1. If the SPIA_*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 SPIA_*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 SPIA_*n*TXD register after making sure the SPIA_*n*INTF.TBEIF bit is set to 1.

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

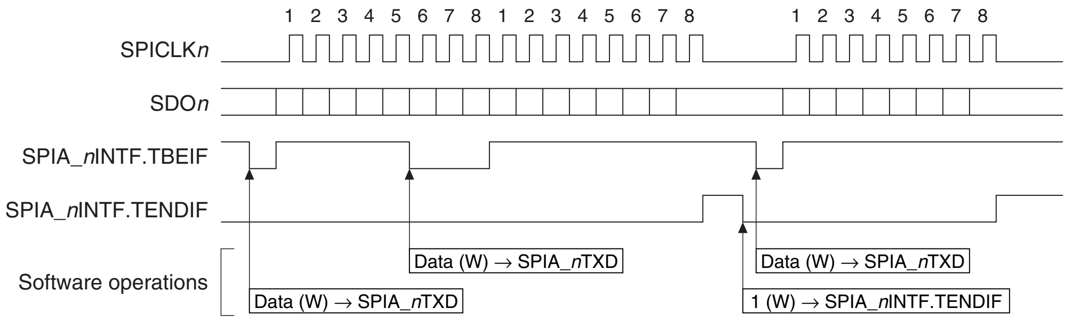


Figure 14.5.2.1 Example of Data Sending Operations in Master Mode (SPIA_*n*MOD.CHLN[3:0] bits = 0x7)

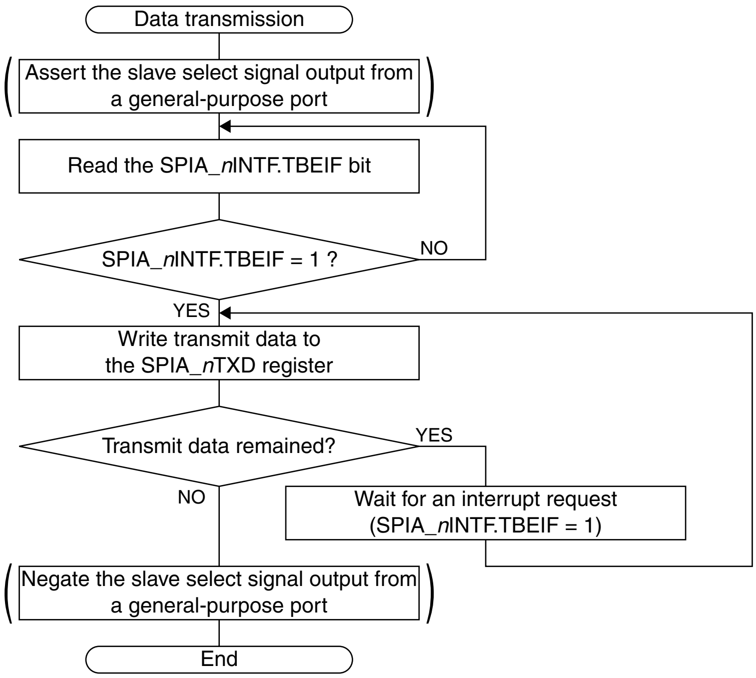


Figure 14.5.2.2 Data Transmission Flowchart in Master Mode

Data transmission using DMA

By setting the SPIA_*n*TBEDMAEN.TBEDMAEN*x* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and transmit data is transferred from the specified memory to the SPIA_*n*TXD register via DMA Ch.*x* when the SPIA_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).

This automates the procedure from Step 2 to Step 5 described above.

The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance so that transmit data will be transferred to the SPIA_*n*TXD register. For more information on DMA, refer to the "DMA Controller" chapter.

Table 14.5.2.1 DMA Data Structure Configuration Example (for 16-bit Data Transmission)

Item		Setting example
End pointer	Transfer source	Memory address in which the last transmit data is stored
	Transfer destination	SPIA_*n*TXD register address
Control data	dst_inc	0x3 (no increment)
	dst_size	0x1 (haflword)
	src_inc	0x1 (+2)
	src_size	0x1 (halfword)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

14.5.3 Data Reception in Master Mode

A data receiving procedure and operations in master mode are shown below. Figures 14.5.3.1 and 14.5.3.2 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 SPIA_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).
3. Write dummy data (or transmit data) to the SPIA_*n*TXD register.
4. Wait for a transmit buffer empty interrupt (SPIA_*n*INTF.TBEIF bit = 1).
5. Write dummy data (or transmit data) to the SPIA_*n*TXD register.
6. Wait for a receive buffer full interrupt (SPIA_*n*INTF.RBFIF bit = 1).
7. Read the received data from the SPIA_*n*RXD register.
8. Repeat Steps 5 to 7 until the end of data reception.
9. Negate the slave select signal by controlling the general-purpose output port (if necessary).

Note: To perform continuous data reception without stopping 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 SPIA_*n*TXD register.
The SPICLK*n* pin outputs clocks of the number of the bits specified by the SPIA_*n*MOD.CHLN[3:0] bits. The transmit data bits are output in sequence from the SDO*n* pin in sync with these clocks and the receive data bits input from the SDI*n* pin are shifted into the shift register.
When the last clock is output from the 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 SPIA_*n*INTF.RBFIF bit is set to 1. At the same time SPIA issues a receive buffer full interrupt request if the SPIA_*n*INTE.RBFIE bit = 1. After that, the received data in the receive data buffer can be read through the SPIA_*n*RXD register.

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

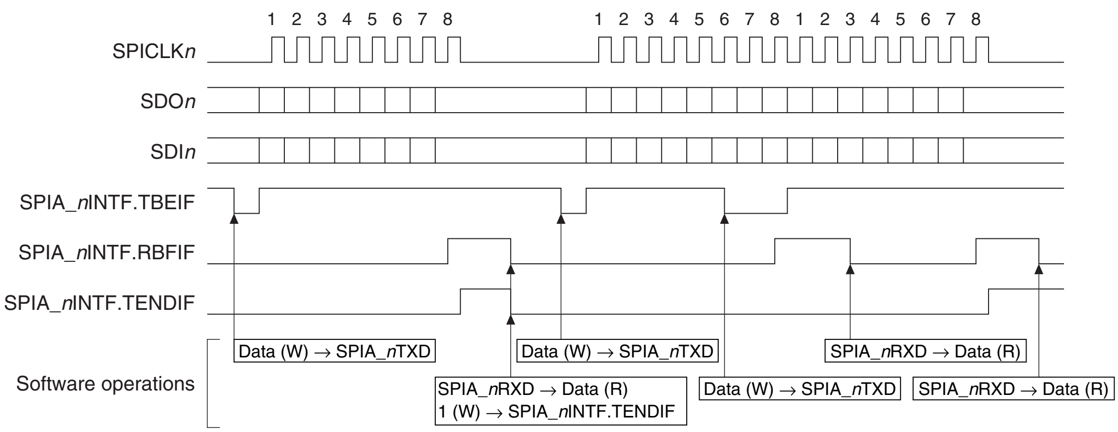
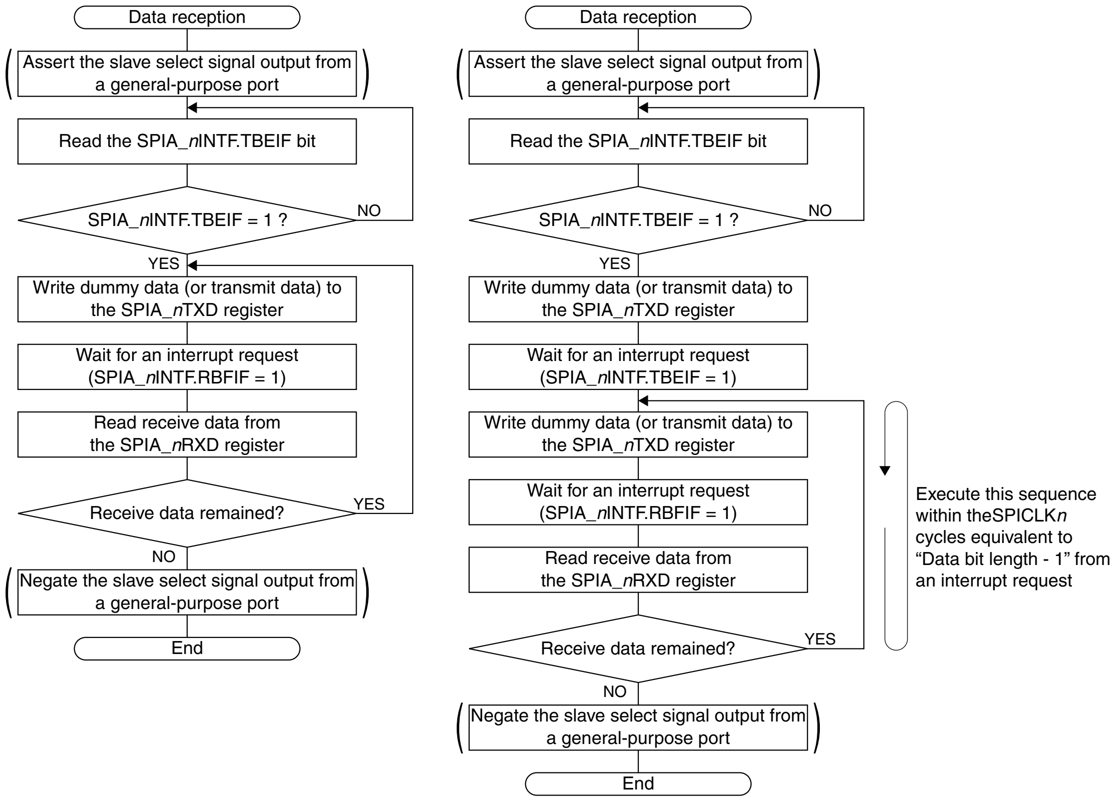


Figure 14.5.3.1 Example of Data Receiving Operations in Master Mode (SPIA_*n*MOD.CHLN[3:0] bits = 0x7)



(A) Intermittent data reception (B) Continuous data reception

Figure 14.5.3.2 Data Reception Flowcharts in Master Mode

Data reception using DMA

For data reception, two DMA controller channels should be used to write dummy data to the SPIA_*n*TXD register as a reception start trigger and to read the received data from the SPIA_*n*RXD register.

By setting the SPIA_*n*TBEDMAEN.TBEDMAEN*x1* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and dummy data is transferred from the specified memory to the SPIA_*n*TXD register via DMA Ch.*x1* when the SPIA_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).

By setting the SPIA_*n*RBFDMAEN.RBFDMAEN*x2* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and the received data is transferred from the SPIA_*n*RXD register to the specified memory via DMA Ch.*x2* when the SPIA_*n*INTF.RBFIF bit is set to 1 (receive buffer full).

This automates the procedure from Step 2 to Step 8 described above.

The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance. For more information on DMA, refer to the "DMA Controller" chapter.

Table 14.5.3.1 DMA Data Structure Configuration Example (for Writing 16-bit Dummy Transmit Data)

Item		Setting example
End pointer	Transfer source	Memory address in which dummy data is stored
	Transfer destination	SPIA_*n*TXD register address
Control data	dst_inc	0x3 (no increment)
	dst_size	0x1 (haflword)
	src_inc	0x3 (no increment)
	src_size	0x1 (halfword)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

Table 14.5.3.2 DMA Data Structure Configuration Example (for 16-bit Data Reception)

Item		Setting example
End pointer	Transfer source	SPIA_*n*RXD register address
	Transfer destination	Memory address to which the last received data is stored
Control data	dst_inc	0x1 (+2)
	dst_size	0x1 (haflword)
	src_inc	0x3 (no increment)
	src_size	0x1 (halfword)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

14.5.4 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 (SPIA_*n*INTF.TENDIF bit = 1).
2. Set the SPIA_*n*CTL.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*.

14.5.5 Data Transfer in Slave Mode

A data sending/receiving procedure and operations in slave mode are shown below. Figures 14.5.5.1 and 14.5.5.2 show a timing chart and flowcharts, respectively.

Data sending procedure

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

Note: Transmit data must be written to the SPIA_*n*TXD register after the SPIA_*n*INTF.TBEIF bit is set to 1 by the time the sending SPIA_*n*TXD register data written is completed. If no transmit data is written during this period, the data bits input from the SDI*n* pin are shifted and output from the SDO*n* pin without being modified.

Data receiving procedure

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

Data transfer operations

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

- Slave mode operates with the SPI clock supplied from the external SPI master to the SPICLK*n* pin.
  The data transfer rate is determined by the SPICLK*n* frequency. It is not necessary to control the 16-bit timer.
- SPIA can operate as a slave device only when the slave select signal input from the external SPI master to the #SPISS*n* pin is set to the active (low) level.
  If #SPISS*n* = high, the software transfer control, the SPICLK*n* pin input, and the SDI*n* pin input are all ineffective. If the #SPISS*n* signal goes high during data transfer, the transfer bit counter is cleared and data in the shift register is discarded.
- Slave mode starts data transfer when SPICLK*n* is input from the external SPI master after the #SPISS*n* signal is asserted. Writing transmit data is not a trigger to start data transfer. Therefore, it is not necessary to write dummy data to the transmit data buffer when performing data reception only.
- Data transmission/reception can be performed even in SLEEP mode, it makes it possible to wake the CPU up using an SPIA interrupt.

Other operations are the same as master mode.

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

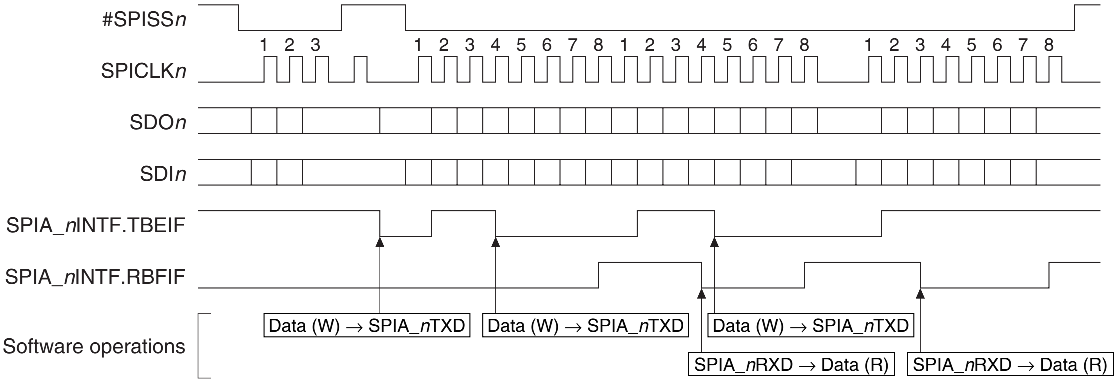


Figure 14.5.5.1 Example of Data Transfer Operations in Slave Mode (SPIA_*n*MOD.CHLN[3:0] bits = 0x7)

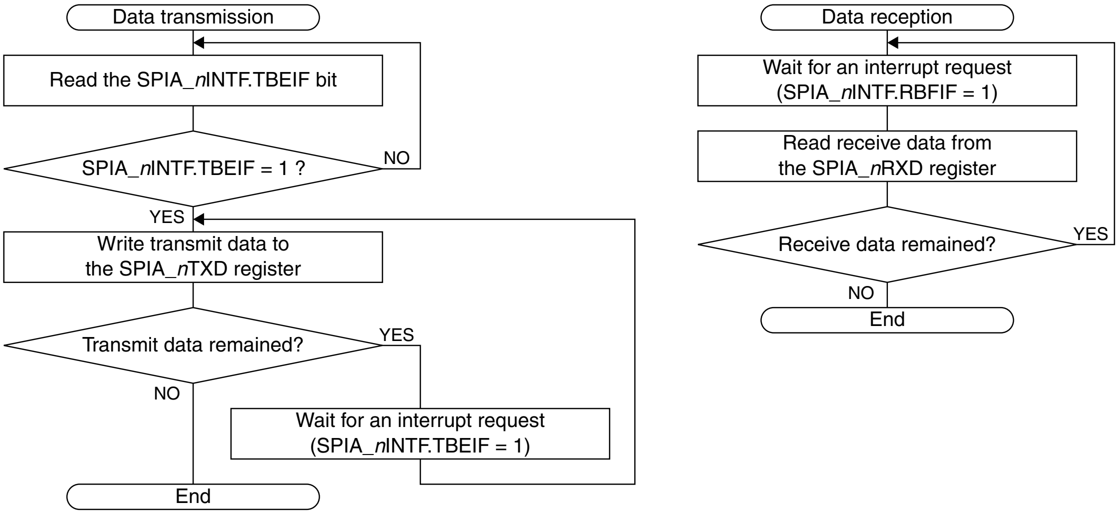


Figure 14.5.5.2 Data Transfer Flowcharts in Slave Mode

14.5.6 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 (SPIA_*n*INTF.TENDIF bit = 1). Or determine end of transfer via the received data.
2. Set the SPIA_*n*CTL.MODEN bit to 0 to disable the SPIA Ch.*n* operations.

14.6 Interrupts

SPIA has a function to generate the interrupts shown in Table 14.6.1.

Table 14.6.1 SPIA Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
End of transmission	SPIA_*n*INTF.TENDIF	When the SPIA_*n*INTF.TBEIF bit = 1 after data of the specified bit length (defined by the SPIA_*n*MOD.CHLN[3:0] bits) has been sent	Writing 1
Receive buffer full	SPIA_*n*INTF.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 SPIA_*n*RXD register
Transmit buffer empty	SPIA_*n*INTF.TBEIF	When transmit data written to the transmit data buffer is transferred to the shift register	Writing to the SPIA_*n*TXD register
Overrun error	SPIA_*n*INTF.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 CPU 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" chapter.
The SPIA_*n*INTF register also contains the BSY bit that indicates the SPIA operating status.
Figure 14.6.1 shows the SPIA_*n*INTF.BSY and SPIA_*n*INTF.TENDIF bit set timings.

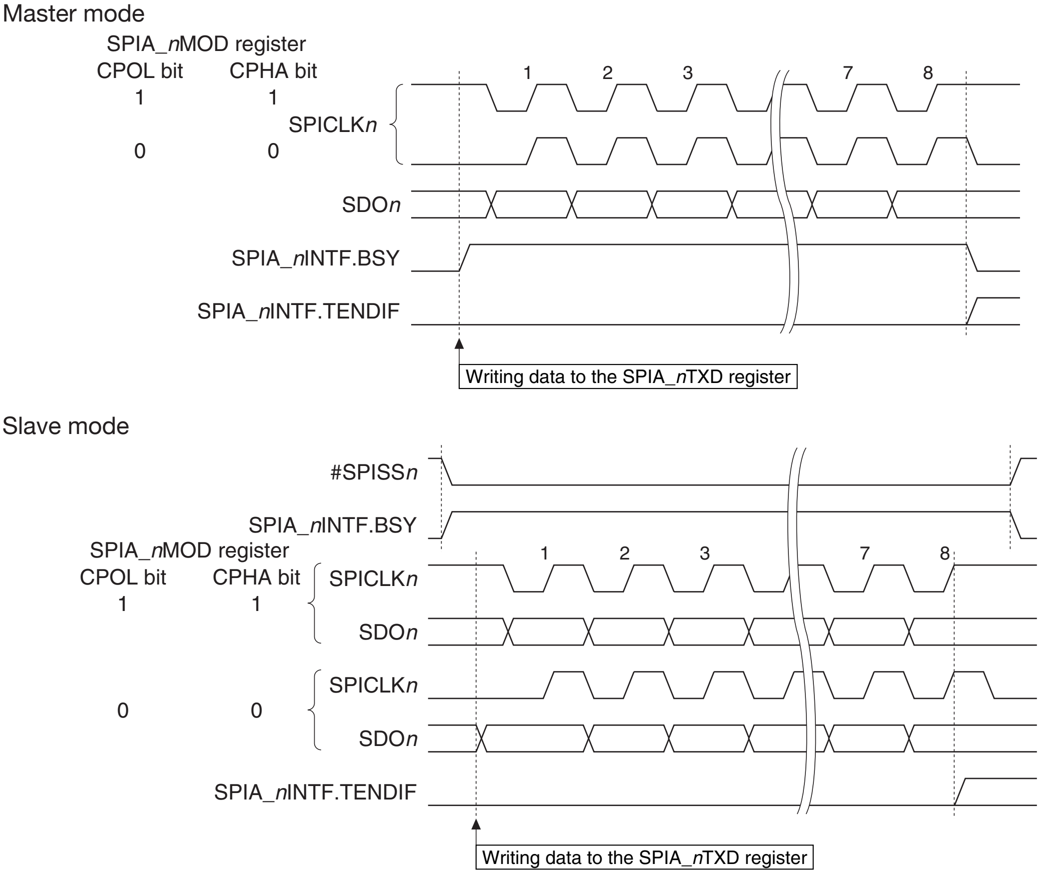


Figure 14.6.1 SPIA_*n*INTF.BSY and SPIA_*n*INTF.TENDIF Bit Set Timings (when SPIA_*n*MOD.CHLN[3:0] bits = 0x7)

14.7 DMA Transfer Requests

The SPIA has a function to generate DMA transfer requests from the causes shown in Table 14.7.1.

Table 14.7.1 DMA Transfer Request Causes of SPIA

Cause to request DMA transfer	DMA transfer request flag	Set condition	Clear condition
Receive buffer full	Receive buffer full flag (SPIA_*n*INTF.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 SPIA_*n*RXD register
Transmit buffer empty	Transmit buffer empty flag (SPIA_*n*INTF.TBEIF)	When transmit data written to the transmit data buffer is transferred to the shift register	Writing to the SPIA_*n*TXD register

The SPIA provides DMA transfer request enable bits corresponding to each DMA transfer request flag shown above for the number of DMA channels. A DMA transfer request is sent to the pertinent channel of the DMA controller only when the DMA transfer request flag, of which DMA transfer has been enabled by the DMA transfer request enable bit, is set. The DMA transfer request flag also serves as an interrupt flag, therefore, both the DMA transfer request and the interrupt cannot be enabled at the same time. After a DMA transfer has completed, disable the DMA transfer to prevent unintended DMA transfer requests from being issued. For more information on the DMA control, refer to the "DMA Controller" chapter.

14.8 Control Registers

SPIA Ch.*n* Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIA_*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.

Table 14.8.1 Data Bit Length Settings

SPIA_*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 SPIA_*n*MOD register settings can be altered only when the SPIA_*n*CTL.MODEN bit = 0.

SPIA Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIA_*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 SPIA_*n*CTL.MODEN bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the SPIA_*n*CTL.MODEN bit to 1 again after that, be sure to write 1 to the SPIA_*n*CTL.SFTRST bit as well.

SPIA Ch.*n* Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIA_*n*TXD	15–0	TXD[15:0]	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 SPIA_*n*INTF.TBEIF bit = 1 regardless of whether data is being output from the SDO*n* pin or not.
Note that the upper data bits that exceed the data bit length configured by the SPIA_*n*MOD.CHLN[3:0] bits will not be output from the SDO*n* pin.

Note: Be sure to avoid writing to the SPIA_*n*TXD register when the SPIA_*n*INTF.TBEIF bit = 0. Otherwise, transfer data cannot be guaranteed.

SPIA Ch.*n* Receive Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIA_*n*RXD	15–0	RXD[15:0]	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 SPIA_*n*INTF.RBFIF bit = 1 regardless of whether data is being input from the SDI*n* pin or not. Note that the upper bits that exceed the data bit length configured by the SPIA_*n*MOD.CHLN[3:0] bits become 0.

SPIA Ch.*n* Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIA_*n*INTF	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 SPIA_*n*RXD register.
	0	TBEIF	1	H0/S0	R	Cleared by writing to the SPIA_*n*TXD register.

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:
SPIA_*n*INTF.OEIF bit: Overrun error interrupt
SPIA_*n*INTF.TENDIF bit: End-of-transmission interrupt
SPIA_*n*INTF.RBFIF bit: Receive buffer full interrupt
SPIA_*n*INTF.TBEIF bit: Transmit buffer empty interrupt

SPIA Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIA_*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:
SPIA_*n*INTE.OEIE bit: Overrun error interrupt
SPIA_*n*INTE.TENDIE bit: End-of-transmission interrupt
SPIA_*n*INTE.RBFIE bit: Receive buffer full interrupt
SPIA_*n*INTE.TBEIE bit: Transmit buffer empty interrupt

SPIA Ch.*n* Transmit Buffer Empty DMA Request Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIA_*n*TBEDMAEN	15–0	TBEDMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 TBEDMAEN[15:0]

These bits enable the SPIA to issue a DMA transfer request to the corresponding DMA channel (Ch.0–Ch.15) when a transmit buffer empty state has occurred.
1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

SPIA Ch.*n* Receive Buffer Full DMA Request Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SPIA_*n*RBFDMAEN	15–0	RBFDMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 RBFDMAEN[15:0]

These bits enable the SPIA to issue a DMA transfer request to the corresponding DMA channel (Ch.0–Ch.15) when a receive buffer full state has occurred.
1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

15 I2C (I2C)

15.1 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.
- Can issue a DMA transfer request when a receive buffer full or a transmit buffer empty occurs.
- The input filter for the SDA and SCL inputs does not comply with the standard for removing noise spikes less than 50 ns.

Figure 15.1.1 shows the I2C configuration.

Table 15.1.1 I2C Channel Configuration of S1C31W65

Item	S1C31W65
Number of channels	2 channels (Ch.0 and Ch.1)

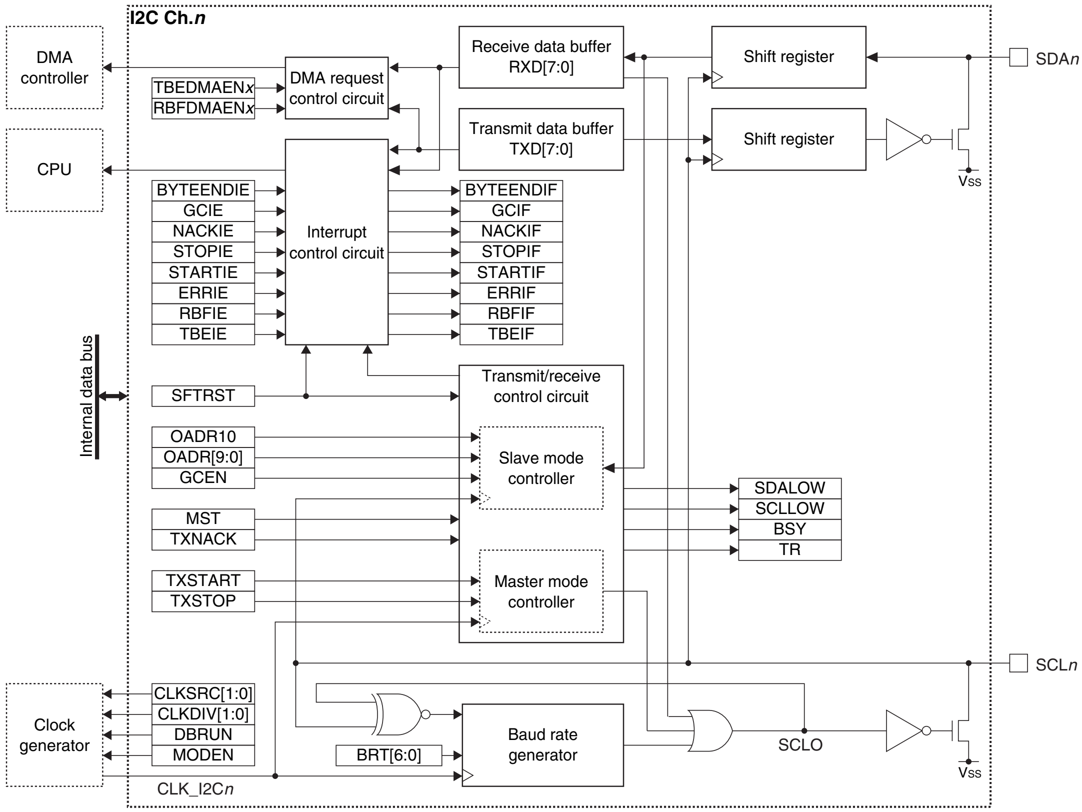


Figure 15.1.1 I2C Configuration

15.2 Input/Output Pins and External Connections

15.2.1 List of Input/Output Pins

Table 15.2.1.1 lists the I2C pins.

Table 15.2.1.1 List of I2C Pins

Pin name	I/O*	Initial status*	Function
SDA*n*	I/O	I	I2C bus serial data input/output pin
SCL*n*	I/O	I	I2C 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.

15.2.2 External Connections

Figure 15.2.2.1 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.

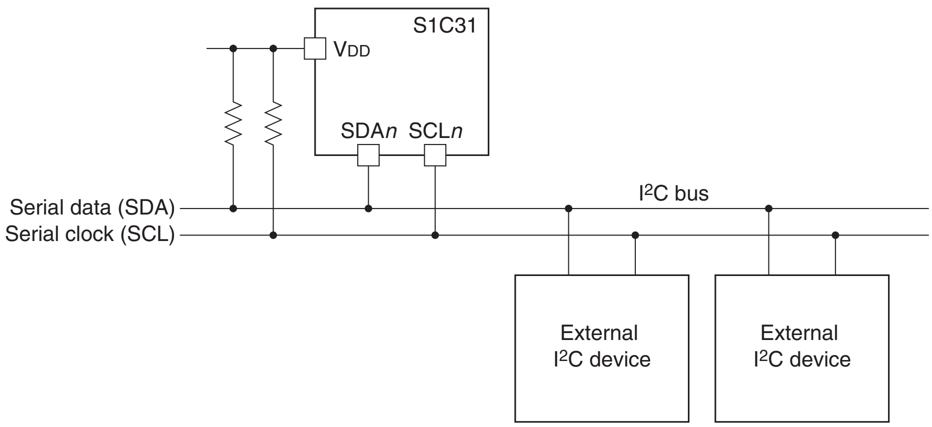


Figure 15.2.2.1 Connections between I2C and External I2C Devices

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.

15.3 Clock Settings

15.3.1 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.*xxxx*SLPC 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 I^2C 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 I^2C master even if all the internal clocks halt during SLEEP mode, so the I2C can receive data and can generate receive buffer full interrupts.

15.3.2 Clock Supply During Debugging

In master mode, the CLK_I2C*n* supply during debugging 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 state if the I2C_*n*CLK.DBRUN bit = 0. After the CPU returns to normal operation, 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 state 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 state.
In slave mode, the I2C Ch.*n* operates with the external I^2C master clock input from the SCL*n* pin regardless of whether the CPU is placed into debug state or normal operation state.

15.3.3 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.

$$bps = \frac{f_{CLK\_I2Cn}}{(BRT + 3) \times 2} \qquad BRT = \frac{f_{CLK\_I2Cn}}{bps \times 2} - 3 \qquad \text{(Eq. 15.1)}$$

Where

bps: Data transfer rate [bit/s]
f_{CLK_I2Cn}: 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 15.3.3.1).

Note: The I^2C 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 15.3.3.1 shows the clock generated by the baud rate generator and the clock waveform on the I²C bus.

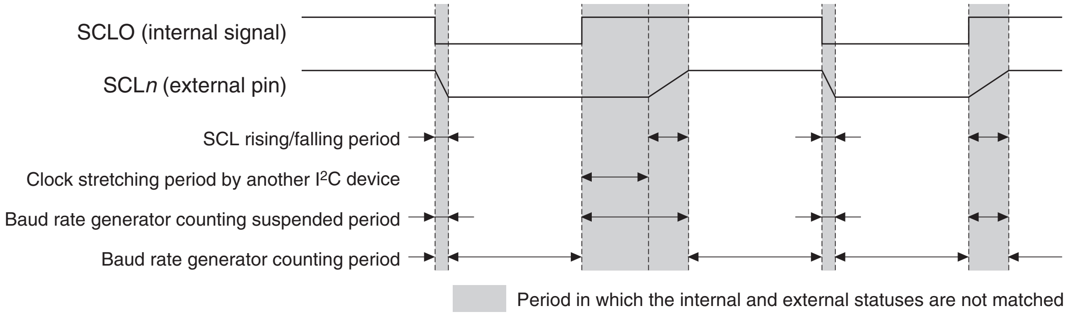


Figure 15.3.3.1 Baud Rate Generator Output Clock and SCL*n* Output Waveform

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.

15.4 Operations

15.4.1 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_*n*CLK and I2C_*n*BR 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_*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the I2C_*n*INTE register to 1. (Enable interrupts)
4. Set the following I2C_*n*CTL register bits:
   - Set the I2C_*n*CTL.MST bit to 1. (Set master mode)
   - Set the I2C_*n*CTL.SFTRST bit to 1. (Execute software reset)
   - Set the I2C_*n*CTL.MODEN bit to 1. (Enable I2C Ch.*n* operations)

When using the I2C in slave mode

1. Set the following I2C_*n*MOD register bits:
   - I2C_*n*MOD.OADR10 bit (Set 10/7-bit address mode)
   - I2C_*n*MOD.GCEN bit (Enable response to general call address)
2. Set its own address to the I2C_*n*OADR.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_*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the I2C_*n*INTE register to 1. (Enable interrupts)
5. Set the following I2C_*n*CTL register bits:
   - Set the I2C_*n*CTL.MST bit to 0. (Set slave mode)
   - Set the I2C_*n*CTL.SFTRST bit to 1. (Execute software reset)
   - Set the I2C_*n*CTL.MODEN bit to 1. (Enable I2C Ch.*n* operations)

15.4.2 Data Transmission in Master Mode

A data sending procedure in master mode and the I2C Ch.*n* operations are shown below. Figures 15.4.2.1 and 15.4.2.2 show an operation example and a flowchart, respectively.

Data sending procedure

1. Issue a START condition by setting the I2C_*n*CTL.TXSTART bit to 1.
2. 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.
3. Write the 7-bit slave address to the I2C_*n*TXD.TXD[7:1] bits and 0 that represents WRITE as the data transfer direction to the I2C_*n*TXD.TXD0 bit.
4. (When DMA is used) Configure the DMA controller and set a DMA transfer request enable bit in the I2C_*n*TBEDMAEN register to 1 (DMA transfer request enabled). (This automates the data sending procedure Steps 5, 6, and 8.)
5. (When DMA is not used) Wait for a transmit buffer empty interrupt (I2C_*n*INTF.TBEIF bit = 1) generated when an ACK is received.
6. (When DMA is not used) Write transmit data to the I2C_*n*TXD register.
7. If a NACK reception interrupt (I2C_*n*INTF.NACKIF bit = 1) has occurred, go to Step 9 or 1 after clearing the I2C_*n*INTF.NACKIF bit.
8. (When DMA is not used) Repeat Steps 5 and 6 until the end of transmit data.
9. Issue a STOP condition by setting the I2C_*n*CTL.TXSTOP bit to 1.
10. 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 sending operations

Generating a START condition

The I2C Ch.*n* starts generating a START condition when the I2C_*n*CTL.TXSTART bit is set to 1. When the generating operation has completed, the I2C Ch.*n* clears the I2C_*n*CTL.TXSTART bit to 0 and sets both the I2C_*n*INTF.STARTIF and I2C_*n*INTF.TBEIF bits to 1.

Sending slave address and data

If the I2C_*n*INTF.TBEIF bit = 1, a slave address or data can be written to the I2C_*n*TXD register. The I2C Ch.*n* pulls down SCL to low and enters standby state until data is written to the I2C_*n*TXD 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 I2C_*n*INTF.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 I2C_*n*INTF.NACKIF bit is set to 1 without setting the I2C_*n*INTF.TBEIF bit.

Generating a STOP/repeated START condition

After the I2C_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty) or the I2C_*n*INTF.NACKIF bit is set to 1 (NACK received), setting the I2C_*n*CTL.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 I2C_*n*CTL.TXSTOP bit is cleared to 0 and the I2C_*n*INTF.STOPIF bit is set to 1.

When setting the I2C_*n*CTL.TXSTART bit to 1 while the I2C_*n*INTF.TBEIF bit = 1 (transmit buffer empty) or the I2C_*n*INTF.NACKIF bit = 1 (NACK received), the I2C Ch.*n* generates a repeated START condition. When the repeated START condition has been generated, the I2C_*n*INTF.STARTIF and I2C_*n*INTF.TBEIF bits are both set to 1 same as when a START condition has been generated.

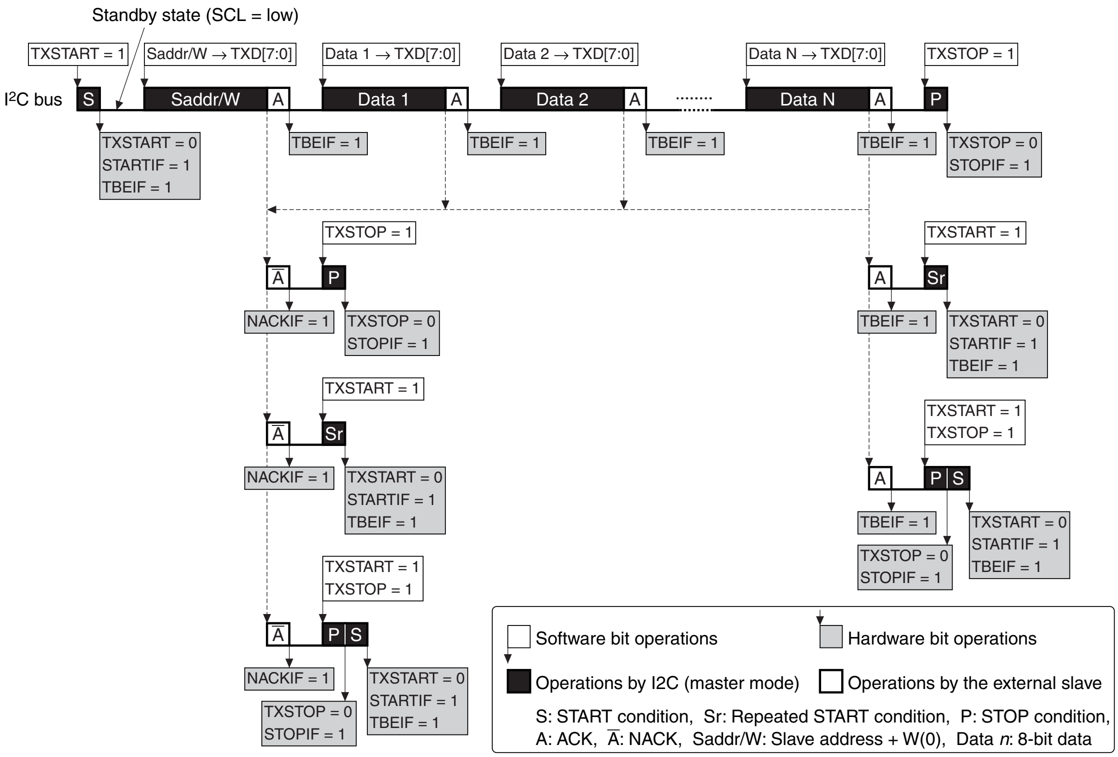


Figure 15.4.2.1 Example of Data Sending Operations in Master Mode

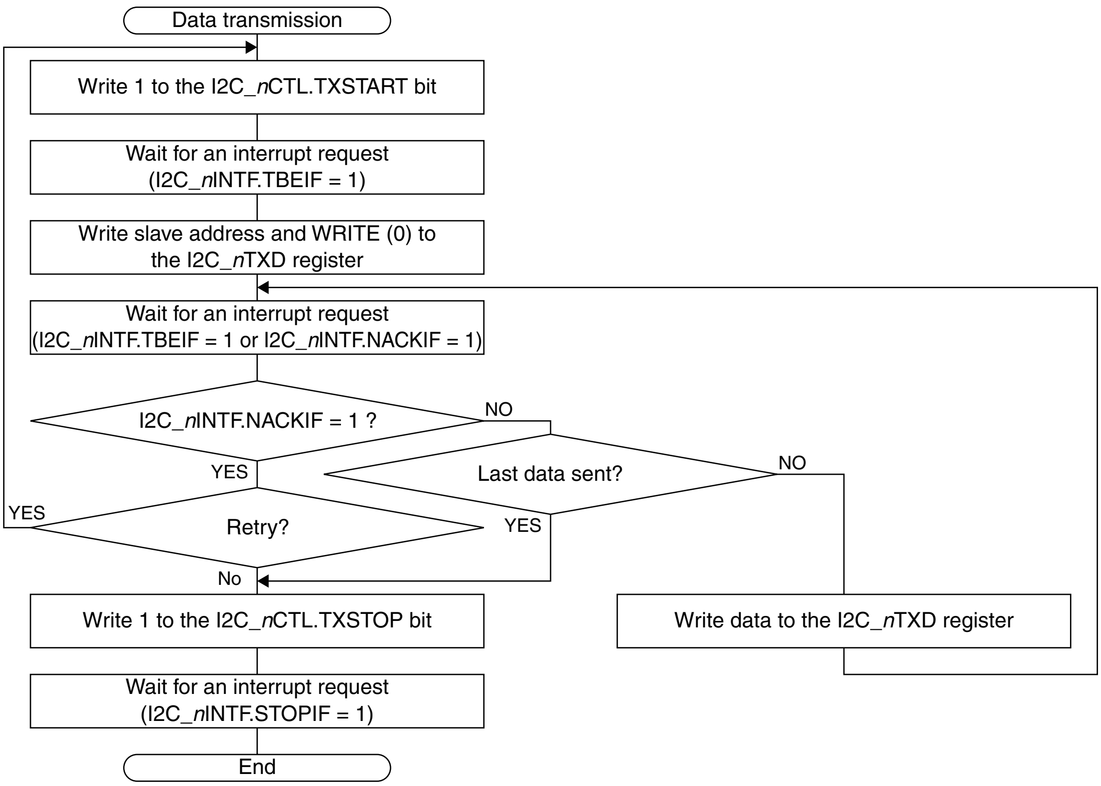


Figure 15.4.2.2 Master Mode Data Transmission Flowchart

Data transmission using DMA

By setting the I2C_*n*TBEDMAEN.TBEDMAEN*x* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and transmit data is transferred from the specified memory to the I2C_*n*TXD register via DMA Ch.*x* when the I2C_*n*INTF.TBEIF bit is set to 1 (transmit buffer empty).

This automates the data sending procedure Steps 5, 6, and 8 described above.

The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance so that transmit data will be transferred to the I2C_*n*TXD register. For more information on DMA, refer to the “DMA Controller” chapter.

Table 15.4.2.1 DMA Data Structure Configuration Example (for Data Transmission)

Item		Setting example
End pointer	Transfer source	Memory address in which the last transmit data is stored
	Transfer destination	I2C_*n*TXD register address
Control data	dst_inc	0x3 (no increment)
	dst_size	0x0 (byte)
	src_inc	0x0 (+1)
	src_size	0x0 (byte)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

15.4.3 Data Reception in Master Mode

A data receiving procedure in master mode and the I2C Ch.*n* operations are shown below. Figures 15.4.3.1 and 15.4.3.2 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. (When DMA is used) Configure the DMA controller and set a DMA transfer request enable bit in the I2C_*n*RBFDMAEN register to 1 (DMA transfer request enabled). (This automates the data receiving procedure Steps 6, 8, and 10.)
6. (When DMA is not used) Wait for a receive buffer full interrupt (I2C_*n*INTF.RBFIF bit = 1) generated when a one-byte reception has completed.
7. 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 8.
   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 11.
8. (When DMA is not used) Read the received data from the I2C_*n*RXD register.
9. If a NACK reception interrupt (I2C_*n*INTF.NACKIF bit = 1) has occurred, clear the I2C_*n*INTF.NACKIF bit and issue a STOP condition by setting the I2C_*n*CTL.TXSTOP bit to 1. Then go to Step 11 or Step 2 if making a retry.
10. (When DMA is not used) Repeat Steps 6 to 8 until the end of data reception.
11. 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.

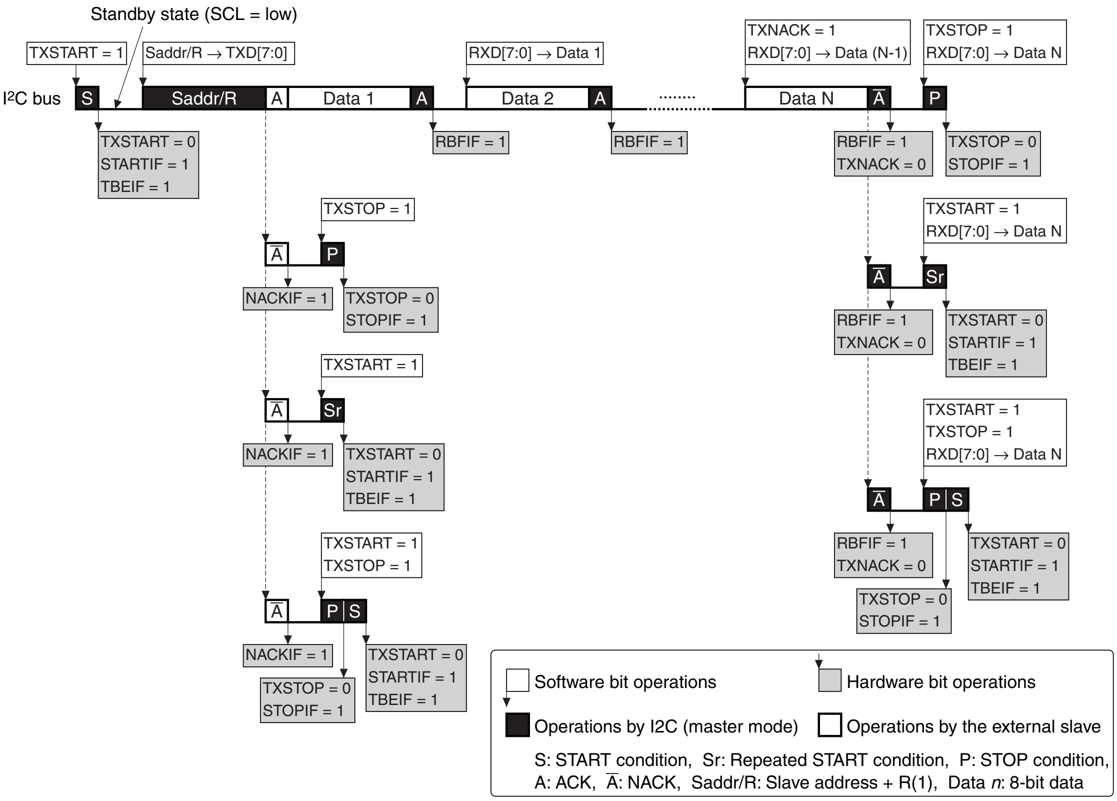


Figure 15.4.3.1 Example of Data Receiving Operations in Master Mode

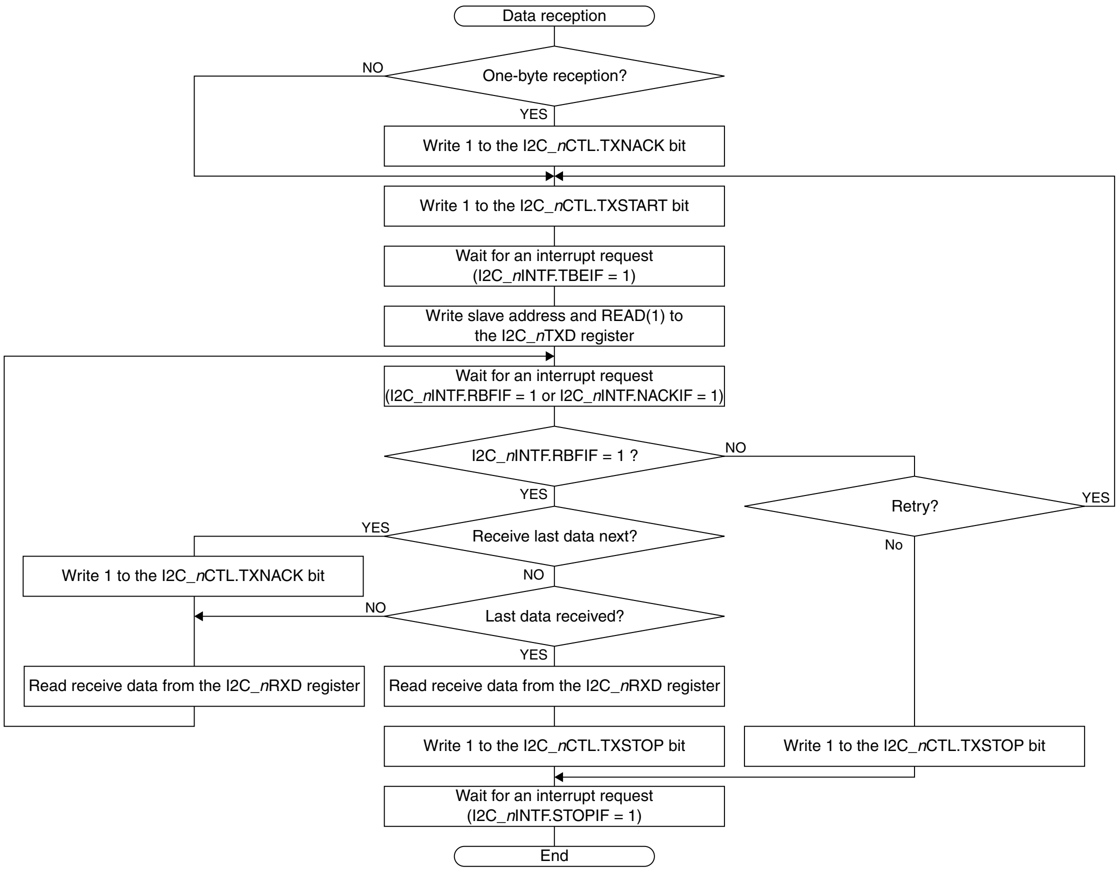


Figure 15.4.3.2 Master Mode Data Reception Flowchart

Data reception using DMA

By setting the I2C_*n*RBFDMAEN.RBFDMAEN*x* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and the received data is transferred from the I2C_*n*RXD register to the specified memory via DMA Ch.*x* when the I2C_*n*INTF.RBFIF bit is set to 1 (receive buffer full).
This automates the data receiving procedure Steps 6, 8, and 10 described above.
The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance. For more information on DMA, refer to the "DMA Controller" chapter.

Table 15.4.3.1 DMA Data Structure Configuration Example (for Data Reception)

Item		Setting example
End pointer	Transfer source	I2C_*n*RXD register address
	Transfer destination	Memory address to which the last received data is stored
Control data	dst_inc	0x0 (+1)
	dst_size	0x0 (byte)
	src_inc	0x3 (no increment)
	src_size	0x0 (byte)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of receive data
	cycle_ctrl	0x1 (basic transfer)

15.4.4 10-bit Addressing in Master Mode

A 10-bit address consists of the first address that contains two high-order bits and the second address that contains eight low-order bits.

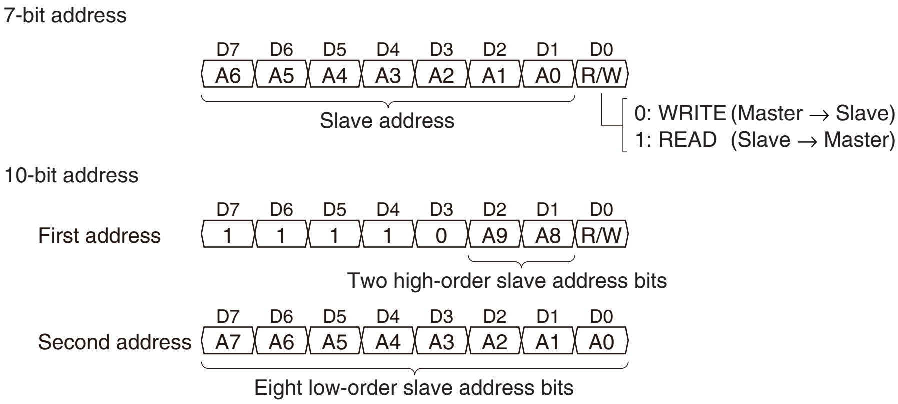


Figure 15.4.4.1 10-bit Address Configuration

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 15.4.4.2 shows an operation example.

Starting data transmission in 10-bit address mode

1. Issue a START condition by setting the I2C_*n*CTL.TXSTART bit to 1.
2. 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.
3. Write the first address to the I2C_*n*TXD.TXD[7:1] bits and 0 that represents WRITE as the data transfer direction to the I2C_*n*TXD.TXD0 bit.
4. Wait for a transmit buffer empty interrupt (I2C_*n*INTF.TBEIF bit = 1).
5. Write the second address to the I2C_*n*TXD.TXD[7:0] bits.
6. Wait for a transmit buffer empty interrupt (I2C_*n*INTF.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 I2C_*n*CTL.TXSTART bit to 1.
8. 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.
9. Write the first 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.
10. Perform data reception.

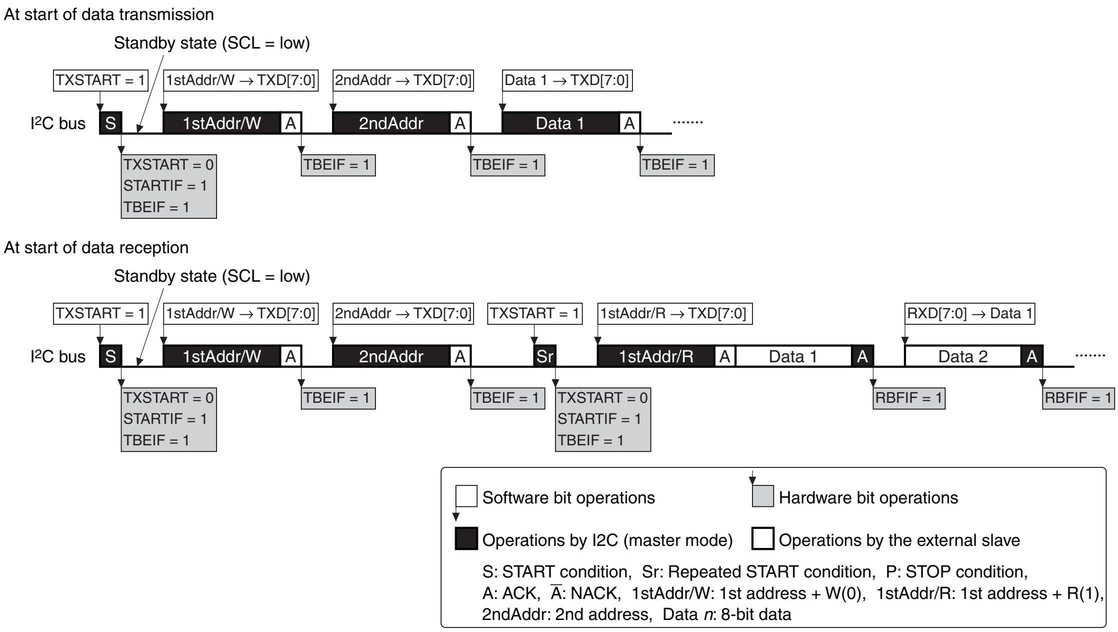


Figure 15.4.4.2 Example of Data Transfer Starting Operations in 10-bit Address Mode (Master Mode)

15.4.5 Data Transmission in Slave Mode

A data sending procedure in slave mode and the I2C Ch.*n* operations are shown below. Figures 15.4.5.1 and 15.4.5.2 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 I2C_*n*CTL.MODEN bit = 1 and the I2C_*n*CTL.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 I2C_*n*OADR.OADR[6:0] bits (when the I2C_*n*MOD.OADR10 bit = 0 (7-bit address mode)) or the I2C_*n*OADR.OADR[9:0] bits (when the I2C_*n*MOD.OADR10 bit = 1 (10-bit address mode)), the I2C_*n*INTF.STARTIF bit and the I2C_*n*INTF.BSY bit are both set to 1. The I2C Ch.*n* sets the I2C_*n*INTF.TR bit to the R/W bit value in the received address. If this value is 1, the I2C Ch.*n* sets the I2C_*n*INTF.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 I2C_*n*TXD register. This puts the I²C bus into clock stretching state and the external master into standby state. When transmit data is written to the I2C_*n*TXD register, the I2C Ch.*n* clears the I2C_*n*INTF.TBEIF bit and sends an ACK to the master. The transmit data written in the I2C_*n*TXD register is automatically transferred to the shift register and the I2C_*n*INTF.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 I2C_*n*INTF.TBEIF bit = 1, subsequent transmit data can be written during data transmission. If the I2C_*n*INTF.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 I2C_*n*TXD register.

If the next transmit data already exists in the I2C_*n*TXD 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 I2C_*n*INTF.BYTEENDIF bit is set to 1. If a NACK is received, the I2C_*n*INTF.NACKIF bit is set to 1 without sending data.

STOP/repeated START condition detection

While the I2C_*n*CTL.MST bit = 0 (slave mode) and the I2C_*n*INTF.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 I2C_*n*INTF.BSY bit is cleared to 0 and the I2C_*n*INTF.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 I2C_*n*INTF.STARTIF bit is set to 1.

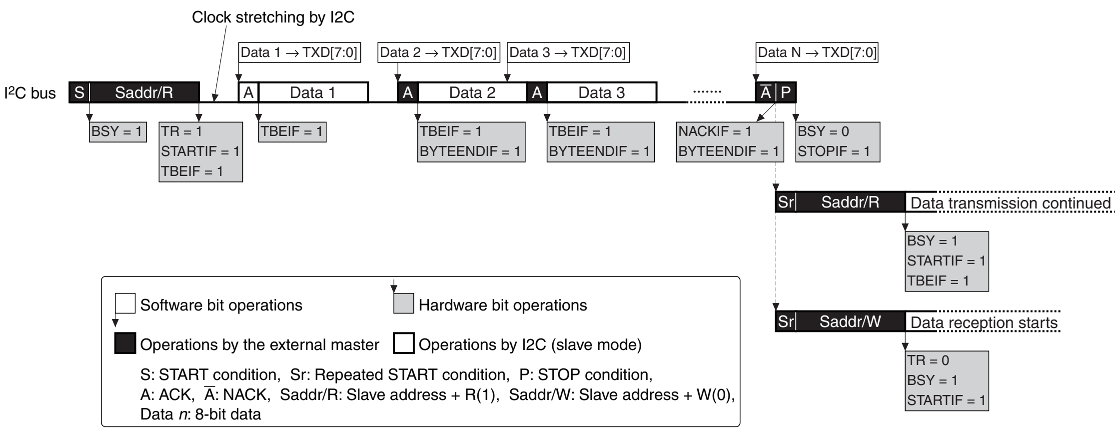


Figure 15.4.5.1 Example of Data Sending Operations in Slave Mode

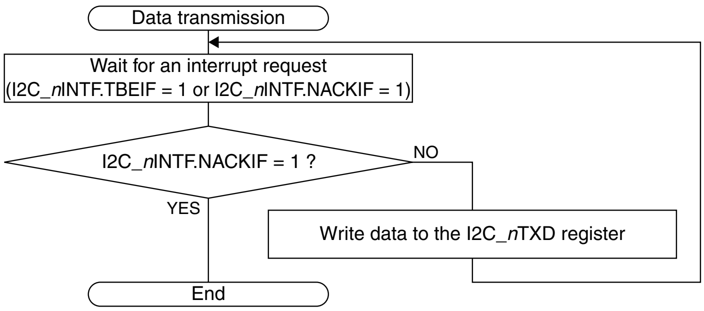


Figure 15.4.5.2 Slave Mode Data Transmission Flowchart

15.4.6 Data Reception in Slave Mode

A data receiving procedure in slave mode and the I2C Ch.*n* operations are shown below. Figures 15.4.6.1 and 15.4.6.2 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. Wait for a START condition interrupt (I2C_*n*INTF.STARTIF bit = 1).
3. Check to see if the I2C_*n*INTF.TR bit = 0 (reception mode).
   (Start a data sending procedure if I2C_*n*INTF.TR bit = 1.)
4. Clear the I2C_*n*INTF.STARTIF bit by writing 1.
5. Wait for a receive buffer full interrupt (I2C_*n*INTF.RBFIF bit = 1) generated when a one-byte reception has completed or an end of transfer interrupt (I2C_*n*INTF.BYTEENDIF bit = 1).
   Clear the I2C_*n*INTF.BYTEENDIF bit by writing 1 after the interrupt has occurred.
6. If the next receive data is the last one, write 1 to the I2C_*n*CTL.TXNACK bit to send a NACK after it is received.
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) or a START condition interrupt (I2C_*n*INTF.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 I2C_*n*INTF.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 I2C_*n*INTF.TR bit is cleared to 0 and the I2C_*n*INTF.TBEIF bit is not set.
If the I2C_*n*MOD.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 I2C_*n*INTF.STARTIF bit. This puts the I^2C bus into clock stretching state and the external master into standby state. When 1 is written to the I2C_*n*INTF.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 I2C_*n*INTF.RBFIF and I2C_*n*INTF.BYTEENDIF bits are both set to 1. After that, the received data can be read out from the I2C_*n*RXD register.

Receiving subsequent data

When the received data is read out from the I2C_*n*RXD register after the I2C_*n*INTF.RBFIF bit has been set to 1, the I2C Ch.*n* clears the I2C_*n*INTF.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 I2C_*n*INTF.RBFIF and I2C_*n*INTF.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 I2C_*n*CTL.TXNACK bit before the data reception is completed. The I2C_*n*CTL.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.

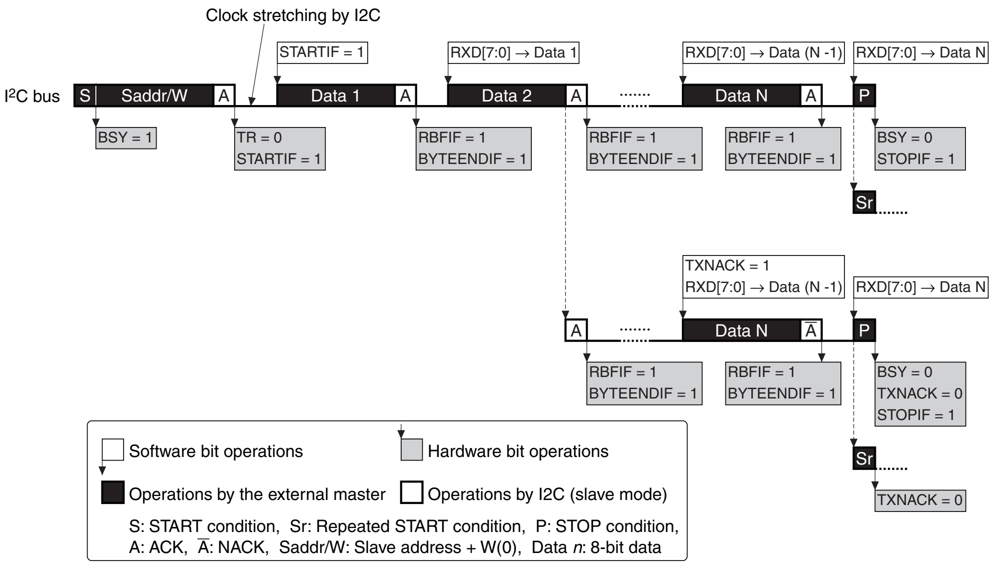


Figure 15.4.6.1 Example of Data Receiving Operations in Slave Mode

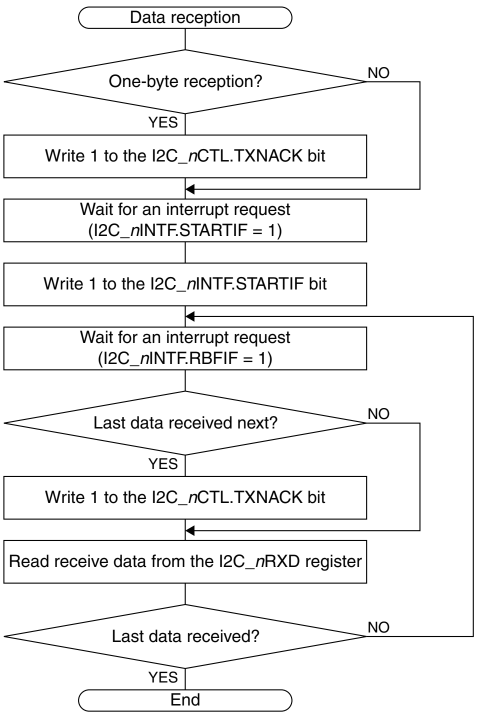


Figure 15.4.6.2 Slave Mode Data Reception Flowchart

15.4.7 Slave Operations in 10-bit Address Mode

The I2C Ch.*n* functions as a slave device in 10-bit address mode when the I2C_*n*CTL.MST bit = 0 and the I2C_*n*MOD.OADR10 bit = 1.
The following shows the address receiving operations in 10-bit address mode. Figure 15.4.7.1 shows an operation example. See Figure 15.4.4.1 for the 10-bit address configuration.

10-bit address receiving operations

After a START condition is issued, the master sends the first address that includes the two high-order slave address bits and the R/W bit (= 0). If the received two high-order slave address bits are matched with the I2C_*n*OADR.OADR[9:8] bits, the I2C Ch.*n* returns an ACK. At this time, other slaves may returns an ACK as the two high-order bits may be matched.
Then the master sends the eight low-order slave address bits as the second address. If this address is matched with the I2C_*n*OADR.OADR[7:0] bits, the I2C Ch.*n* returns an ACK and starts data receiving operations.
If the master issues a request to the slave to send data (data reception in the master), the master generates a repeated START condition and sends the first address with the R/W bit set to 1. This reception switches the I2C Ch.*n* to data sending mode.

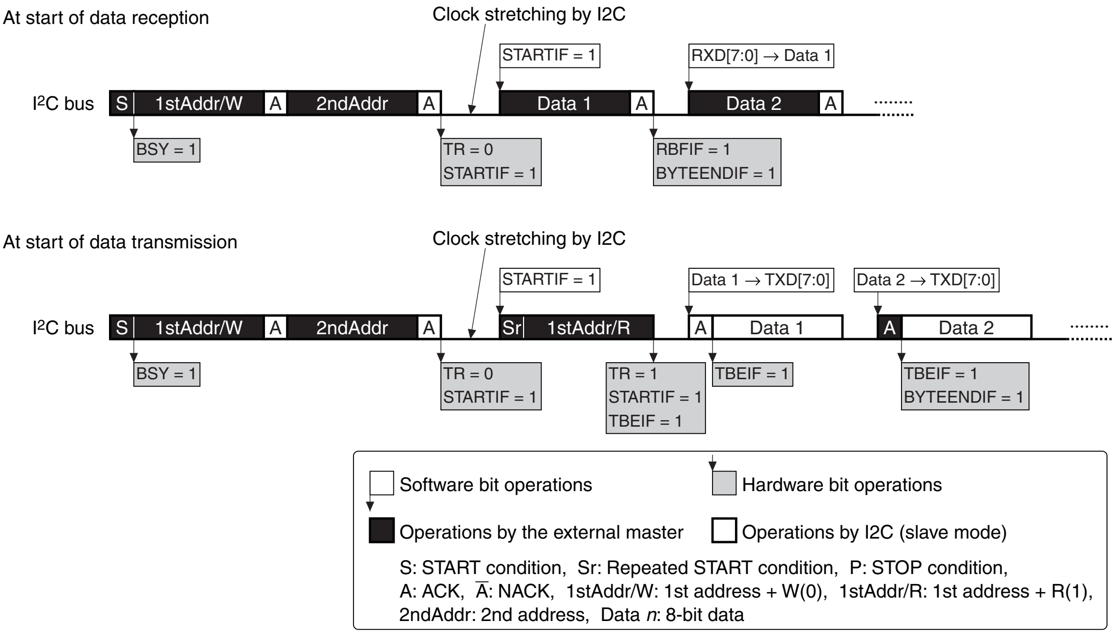


Figure 15.4.7.1 Example of Data Transfer Starting Operations in 10-bit Address Mode (Slave Mode)

15.4.8 Automatic Bus Clearing Operation

The I2C Ch.*n* set into master mode checks the SDA state immediately before generating a START condition. If SDA is set to a low level at this time, the I2C Ch.*n* automatically executes bus clearing operations that output up to ten clocks from the SCL*n* pin with SDA left free state.
When SDA goes high from low within nine clocks, the I2C Ch.*n* issues a START condition and starts normal operations. If SDA does not change from low when the I2C Ch.*n* outputs the ninth clock, it is regarded as an automatic bus clearing failure. In this case, the I2C Ch.*n* clears the I2C_*n*CTL.TXSTART bit to 0 and sets both the I2C_*n*INTF.ERRIF and I2C_*n*INTF.STARTIF bits to 1.

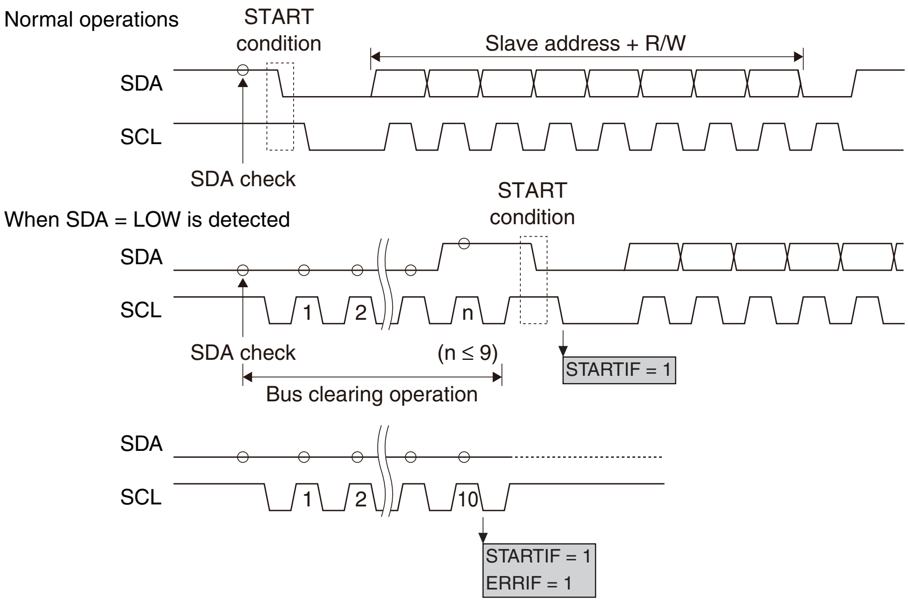


Figure 15.4.8.1 Automatic Bus Clearing Operation

15.4.9 Error Detection

The I2C includes a hardware error detection function.

Furthermore, the I2C_*n*INTF.SDALOW and I2C_*n*INTF.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 15.4.9.1 Hardware Error Detection Function

No.	Error detecting period/timing	I^2C 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	I2C_*n*INTF.ERRIF = 1
2	<Master mode only> When 1 is written to the I2C_*n*CTL.TXSTART bit while the I2C_*n*INTF.BSY bit = 0	SCL = low	I2C_*n*INTF.ERRIF = 1 I2C_*n*CTL.TXSTART = 0 I2C_*n*INTF.STARTIF = 1
3	<Master mode only> When 1 is written to the I2C_*n*CTL.TXSTOP bit while the I2C_*n*INTF.BSY bit = 0	SCL = low	I2C_*n*INTF.ERRIF = 1 I2C_*n*CTL.TXSTOP = 0 I2C_*n*INTF.STOPIF = 1
4	<Master mode only> When 1 is written to the I2C_*n*CTL.TXSTART bit while the I2C_*n*INTF.BSY bit = 0 (Refer to "Automatic Bus Clearing Operation.")	SDA Automatic bus clearing failure	I2C_*n*INTF.ERRIF = 1 I2C_*n*CTL.TXSTART = 0 I2C_*n*INTF.STARTIF = 1

15.5 Interrupts

The I2C has a function to generate the interrupts shown in Table 15.5.1.

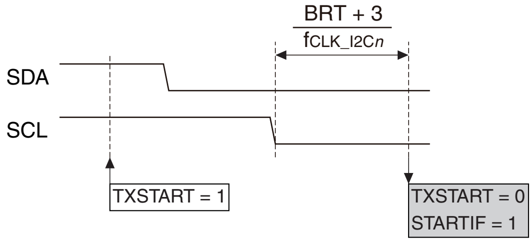
Table 15.5.1 I2C Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
End of data transfer	I2C_*n*INTF.BYTEENDIF	When eight-bit data transfer and the following ACK/NACK transfer are completed	Writing 1, software reset
General call address reception	I2C_*n*INTF.GCIF	Slave mode only: When the general call address is received	Writing 1, software reset
NACK reception	I2C_*n*INTF.NACKIF	When a NACK is received	Writing 1, software reset
STOP condition	I2C_*n*INTF. STOPIF	Master mode: When a STOP condition is generated and the bus free time (t_{BUF}) between STOP and START conditions has elapsed Slave mode: When a STOP condition is detected while the I2C Ch.*n* is selected as the slave currently accessed	Writing 1, software reset
START condition	I2C_*n*INTF. 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	I2C_*n*INTF. ERRIF	Refer to “Error Detection.”	Writing 1, software reset
Receive buffer full	I2C_*n*INTF. 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	I2C_*n*INTF. 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

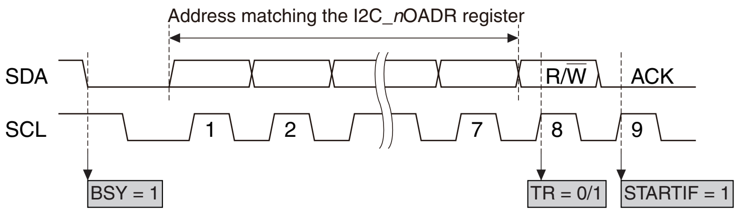
The I2C provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the CPU only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt” chapter.

(1) START condition interrupt

Master mode

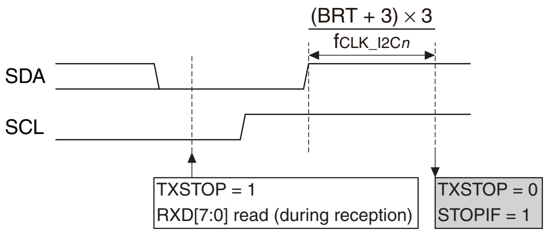


Slave mode

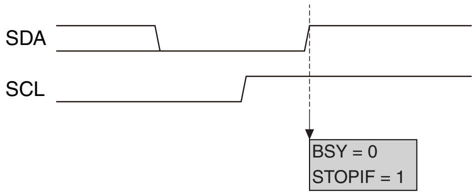


(2) STOP condition interrupt

Master mode



Slave mode



(f_{CLK_I2Cn}: I2C operating clock frequency [Hz], BRT: I2C_*n*BR.BRT[6:0] bits setting value (1 to 127))

Figure 15.5.1 START/STOP Condition Interrupt Timings

15.6 DMA Transfer Requests

The I2C has a function to generate DMA transfer requests from the causes shown in Table 15.6.1.

Table 15.6.1 DMA Transfer Request Causes of I2C

Cause to request DMA transfer	DMA transfer request flag	Set condition	Clear condition
Receive buffer full	Receive buffer full flag (I2C_*n*INTF.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	Transmit buffer empty flag (I2C_*n*INTF.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 DMA transfer request enable bits corresponding to each DMA transfer request flag shown above for the number of DMA channels. A DMA transfer request is sent to the pertinent channel of the DMA controller only when the DMA transfer request flag, of which DMA transfer has been enabled by the DMA transfer request enable bit, is set. The DMA transfer request flag also serves as an interrupt flag, therefore, both the DMA transfer request and the interrupt cannot be enabled at the same time. After a DMA transfer has completed, disable the DMA transfer to prevent unintended DMA transfer requests from being issued. For more information on the DMA control, refer to the “DMA Controller” chapter.

15.7 Control Registers

I2C Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*CLK	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 during debugging or not.

1 (R/W): Clock supplied during debugging

0 (R/W): No clock supplied during debugging

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 15.7.1 Clock Source and Division Ratio Settings

I2C_*n*CLK. CLKDIV[1:0] bits	I2C_*n*CLK.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 I2C_*n*CLK register settings can be altered only when the I2C_*n*CTL.MODEN bit = 0.

I2C Ch.*n* Mode Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*MOD	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 I2C_*n*MOD register settings can be altered only when the I2C_*n*CTL.MODEN bit = 0.

I2C Ch.*n* Baud-Rate Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*BR	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 I2C_*n*BR register settings can be altered only when the I2C_*n*CTL.MODEN bit = 0.

• Be sure to avoid setting the I2C_*n*BR register to 0.

I2C Ch.*n* Own Address Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*OADR	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 I2C_*n*OADR.OADR[9:0] bits are effective in 10-bit address mode (I2C_*n*MOD.OADR10 bit = 1), or the I2C_*n*OADR.OADR[6:0] bits are effective in 7-bit address mode (I2C_*n*MOD.OADR10 bit = 0).

Note: The I2C_*n*OADR register settings can be altered only when the I2C_*n*CTL.MODEN bit = 0.

I2C Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*CTL	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 I2C_*n*CTL.MODEN bit is altered from 1 to 0 while sending/receiving data, the data being sent/received cannot be guaranteed. When setting the I2C_*n*CTL.MODEN bit to 1 again after that, be sure to write 1 to the I2C_*n*CTL.SFTRST bit as well.

I2C Ch.*n* Transmit Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*TXD	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 I2C_*n*INTF.TBEIF bit is set to 1 before writing data.

Note: Be sure to avoid writing to the I2C_*n*TXD register when the I2C_*n*INTF.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
I2C_*n*RXD	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
I2C_*n*INTF	15–13	–	0x0	–	R	–
	12	SDALOW	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 I2C_*n*RXD register.
	0	TBEIF	0	H0/S0	R	Cleared by writing to the I2C_*n*TXD register.

Bits 15–13 Reserved

Bit 12 SDALOW

This bit indicates that SDA is set to low level.
1 (R): SDA = Low level
0 (R): SDA = High level

Bit 11 SCLLOW

This bit indicates that SCL is set to low level.
1 (R): SCL = Low level
0 (R): SCL = High level

Bit 10 BSY

This bit indicates that the I²C bus is placed into busy status.
1 (R): I²C bus busy
0 (R): I²C bus free

Bit 9 TR

This bit indicates whether the I2C is set in transmission mode or not.
1 (R): Transmission mode
0 (R): Reception mode

Bit 8 Reserved

Bit 7 BYTEENDIF
Bit 6 GCIF
Bit 5 NACKIF
Bit 4 STOPIF
Bit 3 STARTIF
Bit 2 ERRIF
Bit 1 RBFIF
Bit 0 TBEIF

These bits indicate the I2C interrupt cause occurrence status.
1 (R): Cause of interrupt occurred
0 (R): No cause of interrupt occurred
1 (W): Clear flag
0 (W): Ineffective

The following shows the correspondence between the bit and interrupt:
I2C_*n*INTF.BYTEENDIF bit: End of transfer interrupt
I2C_*n*INTF.GCIF bit: General call address reception interrupt
I2C_*n*INTF.NACKIF bit: NACK reception interrupt
I2C_*n*INTF.STOPIF bit: STOP condition interrupt
I2C_*n*INTF.STARTIF bit: START condition interrupt
I2C_*n*INTF.ERRIF bit: Error detection interrupt
I2C_*n*INTF.RBFIF bit: Receive buffer full interrupt
I2C_*n*INTF.TBEIF bit: Transmit buffer empty interrupt

I2C Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*INTE	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:
I2C_*n*INTE.BYTEENDIE bit: End of transfer interrupt
I2C_*n*INTE.GCIE bit: General call address reception interrupt
I2C_*n*INTE.NACKIE bit: NACK reception interrupt
I2C_*n*INTE.STOPIE bit: STOP condition interrupt
I2C_*n*INTE.STARTIE bit: START condition interrupt
I2C_*n*INTE.ERRIE bit: Error detection interrupt
I2C_*n*INTE.RBFIE bit: Receive buffer full interrupt
I2C_*n*INTE.TBEIE bit: Transmit buffer empty interrupt

I2C Ch.*n* Transmit Buffer Empty DMA Request Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*TBEDMAEN	15–0	TBEDMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 TBEDMAEN[15:0]

These bits enable the I2C to issue a DMA transfer request to the corresponding DMA controller channel (Ch.0–Ch.15) when a transmit buffer empty state has occurred.

1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

I2C Ch.*n* Receive Buffer Full DMA Request Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
I2C_*n*RBFDMAEN	15–0	RBFDMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 RBFDMAEN[15:0]

These bits enable the I2C to issue a DMA transfer request to the corresponding DMA controller channel (Ch.0–Ch.15) when a receive buffer full state has occurred.

1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

16 16-bit PWM Timers (T16B)

16.1 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 or DMA request 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 or DMA requests. (Can be used to measure external event periods/cycles.)

Figure 16.1.1 shows the T16B configuration.

Table 16.1.1 T16B Channel Configuration of S1C31W65

Item	S1C31W65
Number of channels	3 channels (Ch.0–Ch.2)
Event counter function	Ch.0: EXCL00 or EXCL01 pin input Ch.1: EXCL10 or EXCL11 pin input Ch.2: EXCL20 or EXCL21 pin input
Number of comparator/ capture circuits per channel	4 systems (0 to 3)
Timer generating signal output	Ch.0: TOUT00 to TOUT03 pin outputs (4 systems) Ch.1: TOUT10 to TOUT13 pin outputs (4 systems) Ch.2: TOUT20 to TOUT23 pin outputs (4 systems)
Capture signal input	Ch.0: CAP00 tp CAP03 pin inputs (4 systems) Ch.1: CAP10 to CAP13 pin inputs (4 systems) Ch.2: CAP20 to CAP23 pin inputs (4 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.

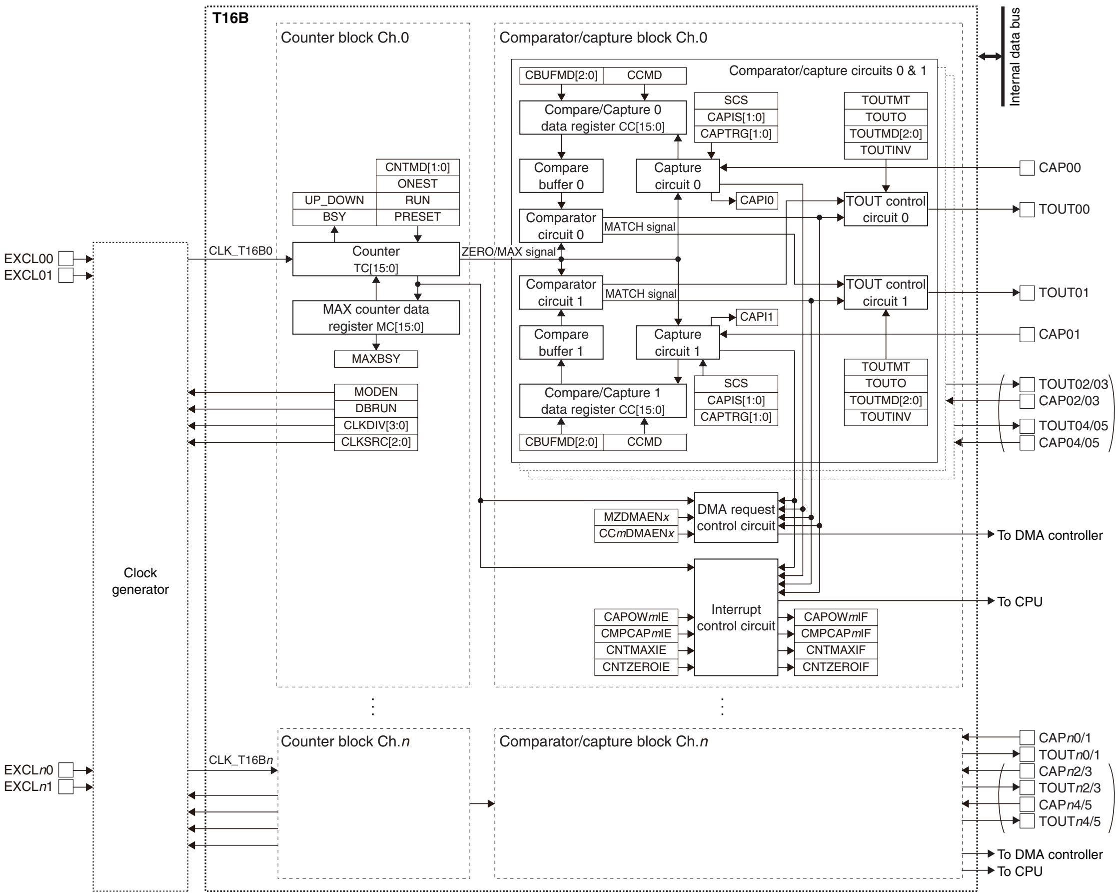


Figure 16.1.1 T16B Configuration

16.2 Input/Output Pins

Table 16.2.1 lists the T16B pins.

Table 16.2.1 List of T16B Pins

Pin name	I/O*	Initial status*	Function
EXCL*nm*	I	I (Hi-Z)	External clock input
TOUT*nm*/CAP*nm*	O or I	O (L)	TOUT signal output (in comparator mode) or capture trigger signal input (in capture mode)

* Indicates the status when the pin is configured for T16B.

If the port is shared with the T16B pin and other functions, the T16B input/output function must be assigned to the port before activating T16B. For more information, refer to the "I/O Ports" chapter.

16.3 Clock Settings

16.3.1 T16B Operating Clock

When using T16B Ch.*n*, the T16B Ch.*n* operating clock CLK_T16B*n* must be supplied to T16B Ch.*n* from the clock generator. The CLK_T16B*n* supply should be controlled as in the procedure shown below.

1. Enable the clock source in the clock generator if it is stopped (refer to “Clock Generator” in the “Power Supply, Reset, and Clocks” chapter).

   When an external clock is used, select the EXCL*nm* pin function (refer to the “I/O Ports” chapter).

2. Set the following T16B_*n*CLK register bits:
   - T16B_*n*CLK.CLKSRC[2:0] bits (Clock source selection)
   - T16B_*n*CLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)

16.3.2 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.*xxxx*SLPC bit for the CLK_T16B*n* clock source.
If the CLGOSC.*xxxx*SLPC 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.

16.3.3 Clock Supply During Debugging

The CLK_T16B*n* supply during debugging should be controlled using the T16B_*n*CLK.DBRUN bit.
The CLK_T16B*n* supply to T16B Ch.*n* is suspended when the CPU enters debug state if the T16B_*n*CLK.DBRUN bit = 0. After the CPU returns to normal operation, 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 state was entered. If the T16B_*n*CLK.DBRUN bit = 1, the CLK_T16B*n* supply is not suspended and T16B Ch.*n* will keep operating in debug state.

16.3.4 Event Counter Clock

When EXCL*nm* is selected as the clock source using the T16B_*n*CLK.CLKSRC[2:0] bits, the channel functions as a timer or event counter that counts the EXCL*nm* pin input clocks.
The counter counts rising edges of the input signal. This can be changed so that the counter will count falling edges of the original signal by selecting EXCL*nm* inverted input as the clock source.

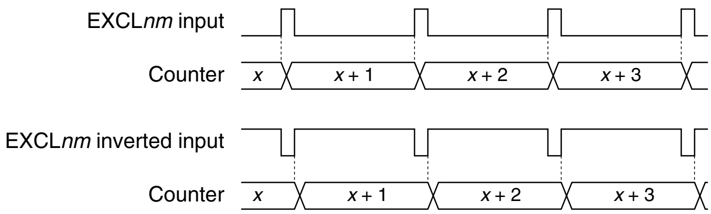


Figure 16.3.4.1 Count Timing (During Count Up Operation)

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.

16.4 Operations

16.4.1 Initialization

T16B Ch.*n* should be initialized and started counting with the procedure shown below. Perform initial settings for comparator mode when using T16B as an interval timer, PWM waveform generator, or external event counter. Perform initial settings for capture mode when using T16B to measure external event periods/cycles.

Initial settings for comparator mode

1. Configure the T16B Ch.*n* operating clock.
2. Set the T16B_*n*CTL.MODEN bit to 1. (Enable T16B operations)
3. Set the following T16B_*n*CCCTL0 and T16B_*n*CCCTL1 register bits:
   - Set the T16B_*n*CCCTL*m*.CCMD bit to 0. * (Set comparator mode)
   - T16B_*n*CCCTL*m*.CBUFMD[2:0] bits (Configure compare buffer)

   * Another circuit in the comparator/capture circuit pair (circuits 0 and 1, 2 and 3, 4 and 5) can be set to capture mode.

   Set the following bits when the TOUT*nm* output is used.
   - T16B_*n*CCCTL*m*.TOUTMT bit (Select waveform generation signal)
   - T16B_*n*CCCTL*m*.TOUTMD[2:0] bits (Select TOUT signal generation mode)
   - T16B_*n*CCCTL*m*.TOUTINV bit (Select TOUT signal polarity)
4. Set the T16B_*n*MC register. (Set MAX counter data)
5. Set the T16B_*n*CCR0 and T16B_*n*CCR1 registers. (Set the counter comparison value)
6. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the T16B_*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the T16B_*n*INTE register to 1. (Enable interrupts)
7. Configure the DMA controller and set the following T16B control bits when using DMA transfer:
   - Write 1 to the DMA transfer request enable bits in the T16B_*n*MZDMAEN and T16B_*n*CC*m*DMAEN registers. (Enable DMA transfer requests)
8. Set the following T16B_*n*CTL register bits:
   - T16B_*n*CTL.CNTMD[1:0] bits (Select count up/down operation)
   - T16B_*n*CTL.ONEST bit (Select one-shot/repeat operation)
   - Set the T16B_*n*CTL.PRESET bit to 1. (Reset counter)
   - Set the T16B_*n*CTL.RUN bit to 1. (Start counting)

Initial settings for capture mode

1. Configure the T16B Ch.*n* operating clock.
2. Set the T16B_*n*CTL.MODEN bit to 1. (Enable T16B operations)
3. Set the following T16B_*n*CCCTL0 and T16B_*n*CCCTL1 register bits:
   - Set the T16B_*n*CCCTL*m*.CCMD bit to 1. * (Set capture mode)
   - T16B_*n*CCCTL*m*.SCS bit (Set synchronous/asynchronous mode)
   - T16B_*n*CCCTL*m*.CAPIS[1:0] bits (Set trigger signal)
   - T16B_*n*CCCTL*m*.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 T16B_*n*MC register. (Set MAX counter data)
5. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the T16B_*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the T16B_*n*INTE register to 1. (Enable interrupts)

6. Configure the DMA controller and set the following T16B control bits when using DMA transfer:
   - Write 1 to the DMA transfer request enable bits in the T16B_*n*MZDMAEN and T16B_*n*CC*m*DMAEN registers. (Enable DMA transfer requests)
7. Set the following T16B_*n*CTL register bits:
   - T16B_*n*CTL.CNTMD[1:0] bits (Select count up/down operation)
   - T16B_*n*CTL.ONEST bit (Select one-shot/repeat operation)
   - Set the T16B_*n*CTL.PRESET bit to 1. (Reset counter)
   - Set the T16B_*n*CTL.RUN bit to 1. (Start counting)

16.4.2 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.

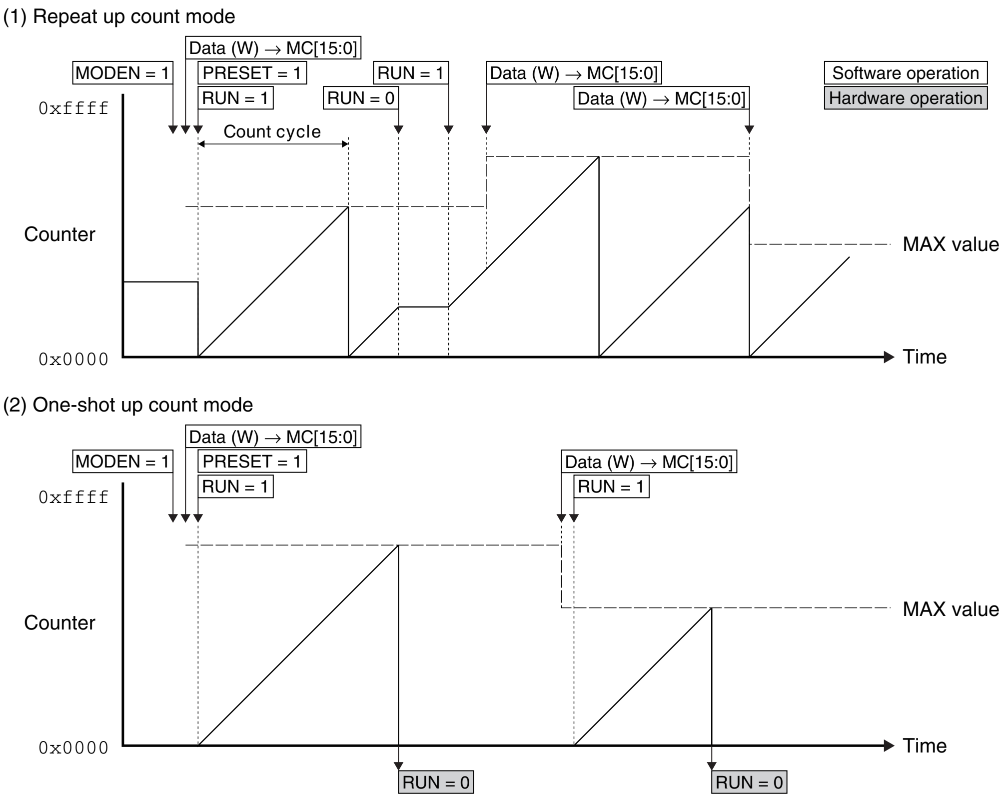


Figure 16.4.2.1 Operations in Repeat Up Count and One-shot Up Count Modes

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 T16B_*n*CTL.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

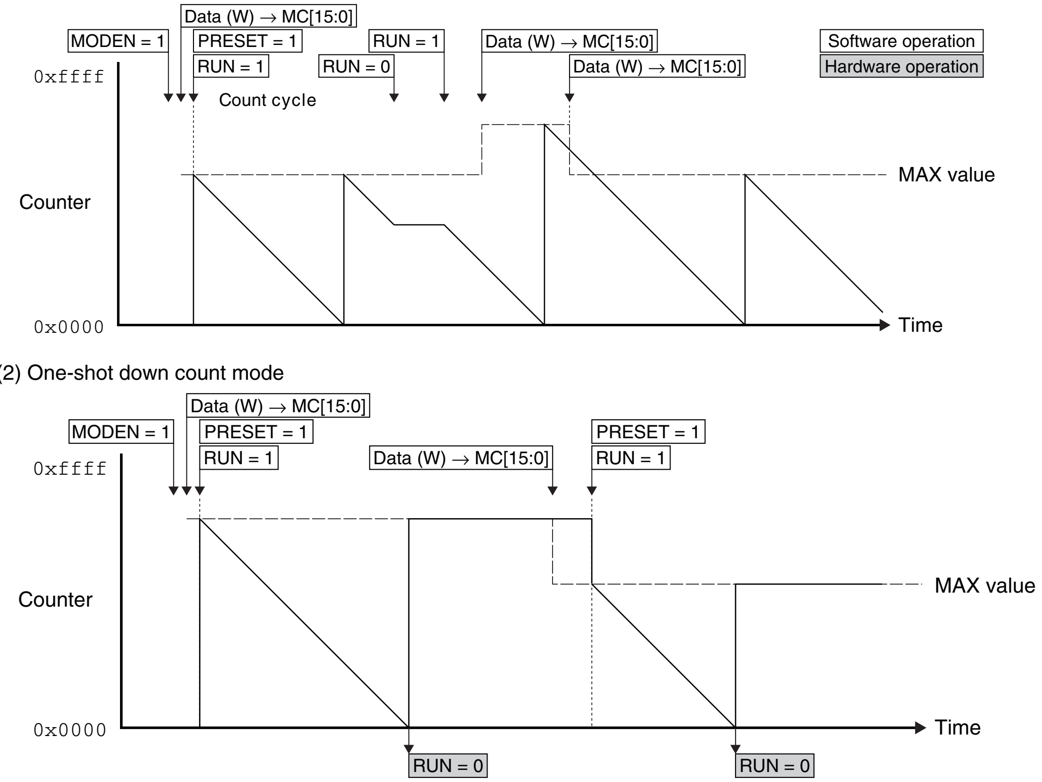


Figure 16.4.2.2 Operations in Repeat Down Count and One-shot Down Count Modes

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 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 count up operation, the counter keeps counting up to the new MAX value. If the MAX value is altered to a value smaller than the current counter value, the counter is cleared to 0x0000 and continues counting up to the new MAX value. If the MAX value is altered during count down operation, the counter keeps counting down to 0x0000 and then starts counting up to the new MAX value.

In one-shot up/down count mode, the counter stops automatically when it reaches 0x0000 during count down operation.

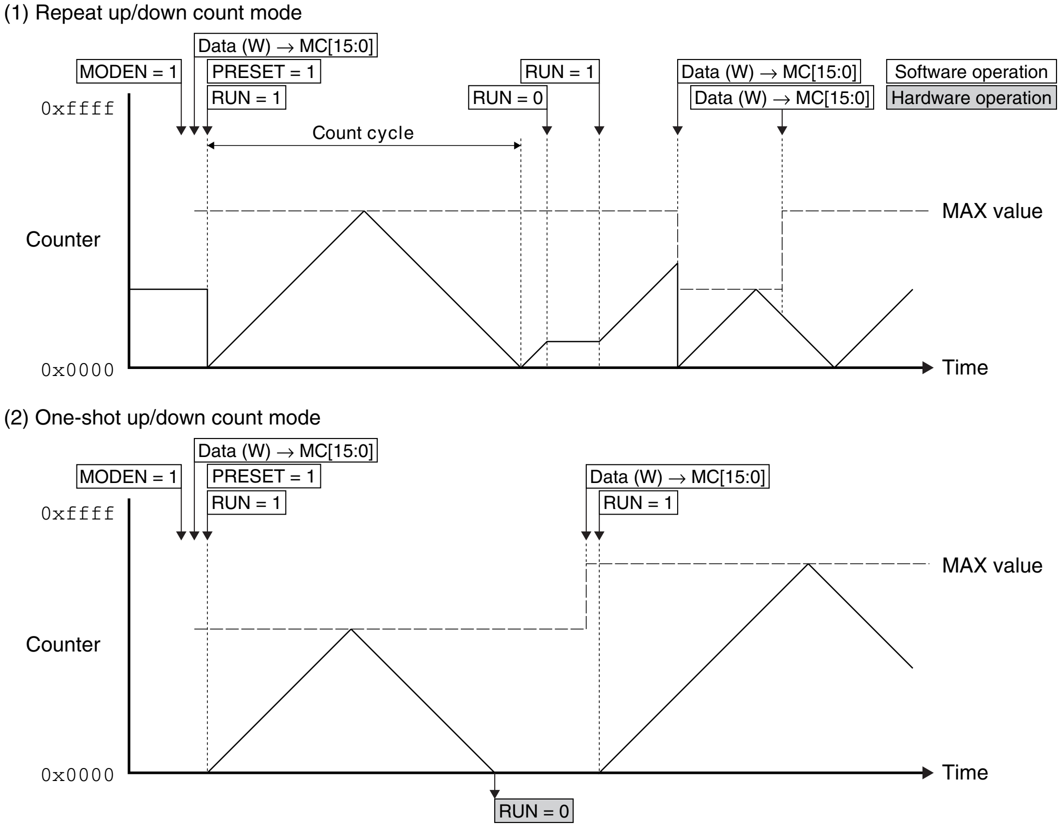


Figure 16.4.2.3 Operations in Repeat Up/Down Count and One-shot Up/Down Count Modes

16.4.3 Comparator/Capture Block Operations

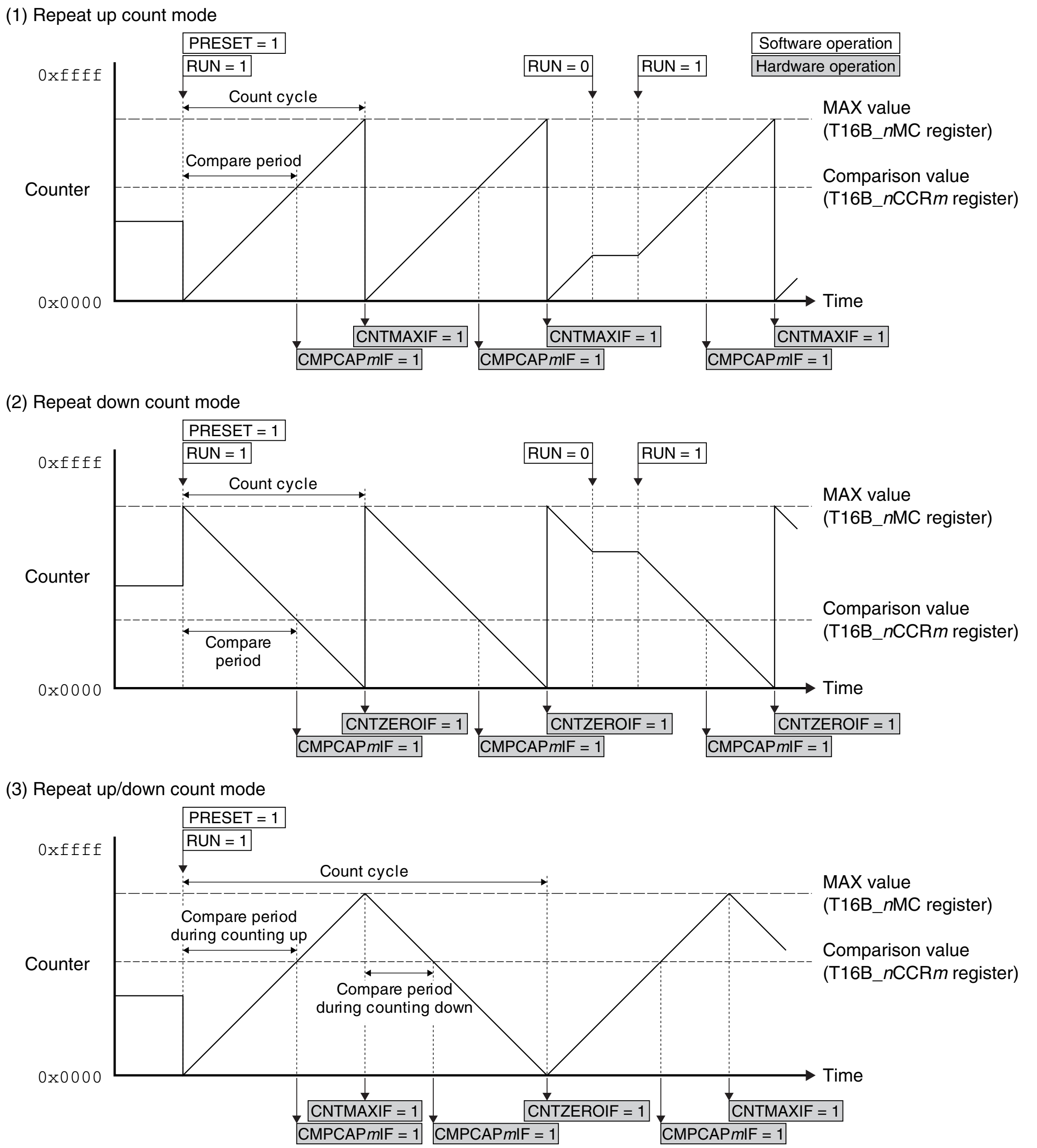
The comparator/capture block functions as a comparator to compare the counter value with the register value set or a capture circuit to capture counter values using the external/software trigger signals.

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 T16B_*n*CCCTL*m*.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 T16B_*n*CCR*m* register functions as the compare data register used for setting a comparison value in this mode. The TOUT*nm*/CAP*nm* pin is configured to the TOUT*nm* pin.
When the counter reaches the value set in the T16B_*n*CCR*m* register during counting, the comparator asserts the MATCH signal and sets the T16B_*n*INTF.COMPCAP*m*IF bit (compare interrupt flag) to 1.
When the counter reaches the MAX value in comparator mode, the T16B_*n*INTF.CNTMAXIF bit (counter MAX interrupt flag) is set to 1. When the counter reaches 0x0000, the T16B_*n*INTF.CNTZEROIF bit (counter zero interrupt flag) is set to 1.



(Note that the T16B_*n*INTF.CMPCAP*m*IF/CNTMAXIF/CNTZEROIF bit clearing operations via software are omitted from the figure.)

Figure 16.4.3.1 Operation Examples in Comparator Mode

The time from counter = 0x0000 or MAX value to occurrence of a compare interrupt (compare period) and the time to occurrence of a counter MAX or counter zero interrupt (count cycle) can be calculated as follows:

During counting up

$$\text{Compare period} = \frac{(CC + 1)}{f_{CLK\_T16B}} \text{ [s]} \qquad \text{Count cycle} = \frac{(MAX + 1)}{f_{CLK\_T16B}} \text{ [s]} \qquad \text{(Eq. 16.1)}$$

During counting down

$$\text{Compare period} = \frac{(MAX - CC + 1)}{f_{CLK\_T16B}} \text{ [s]} \qquad \text{Count cycle} = \frac{(MAX + 1)}{f_{CLK\_T16B}} \text{ [s]} \qquad \text{(Eq. 16.2)}$$

Where

CC: T16B_*n*CCR*m* register setting value (0 to 65,535)

MAX: T16B_*n*MC register setting value (0 to 65,535)

f_{CLK_T16B}: Count clock frequency [Hz]

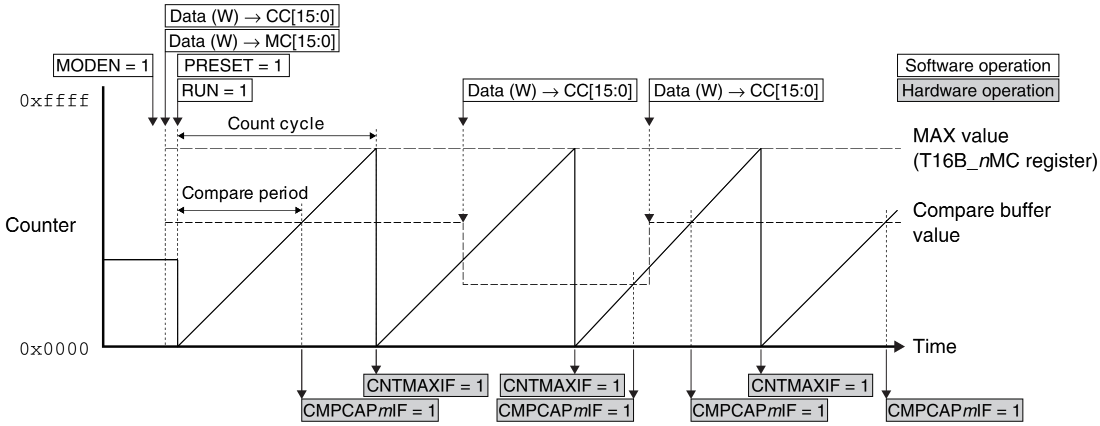
The comparator MATCH signal and counter MAX/ZERO signals are also used to generate a timer output waveform (TOUT). Refer to “TOUT Output Control” for more information.

Compare buffer

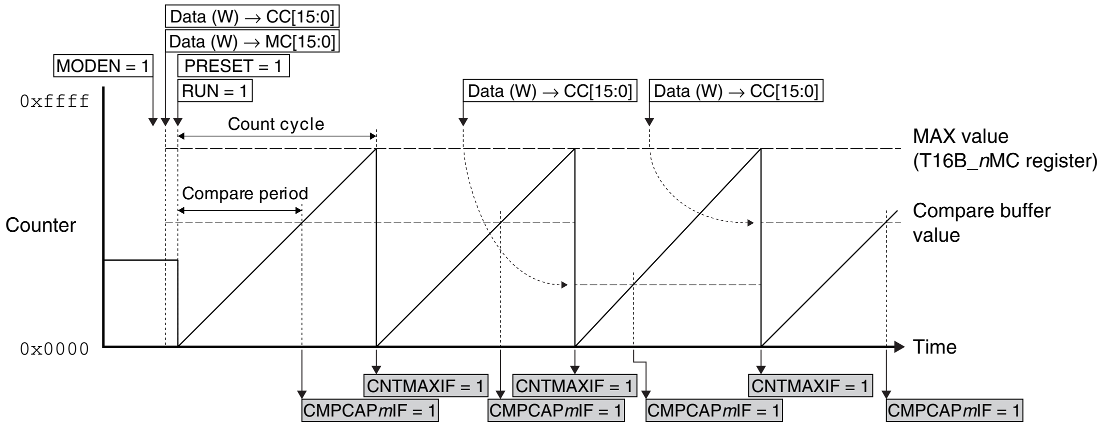
The comparator loads the comparison value, which has been written to the T16B_*n*CCR*m* register, to the compare buffer before comparing it with the counter value. For example, when generating a PWM waveform, the waveform with the desired duty ratio may not be generated if the comparison value is altered 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 T16B_*n*CCCTL*m*.CBUFMD[2:0] bits for synchronization with the count operation.

(1) Repeat up count mode

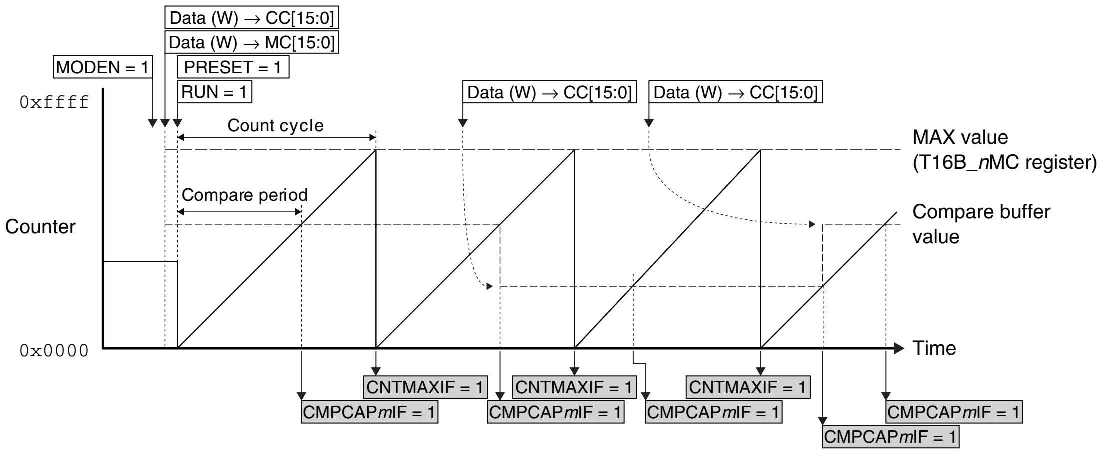
(1.1) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x0



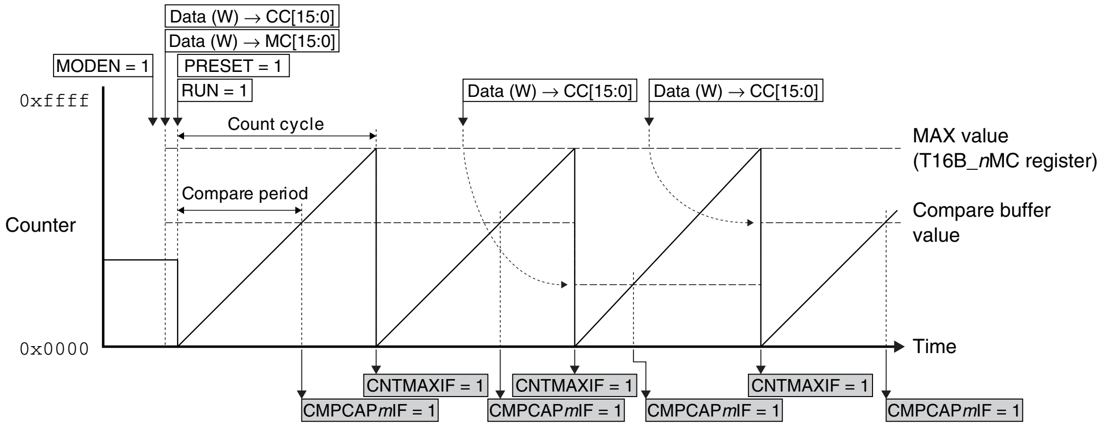
(1.2) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x1



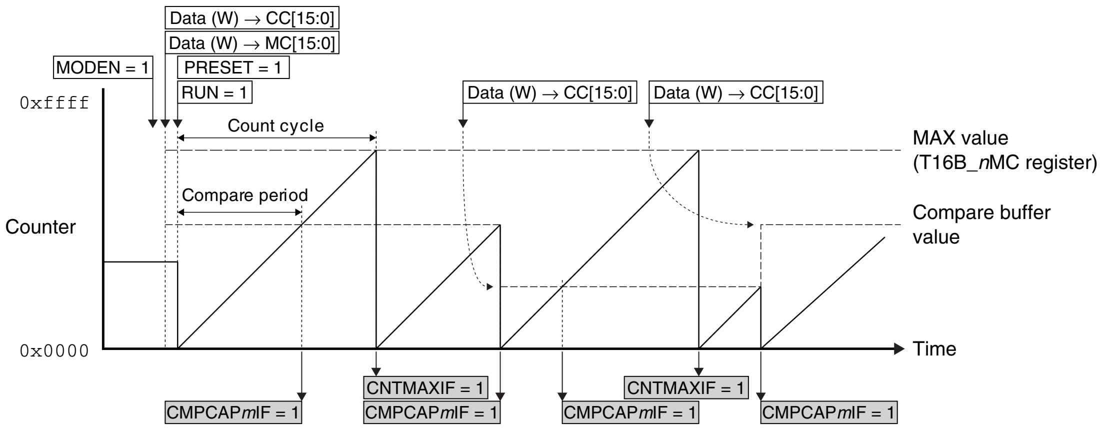
(1.3) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x2



(1.4) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x3

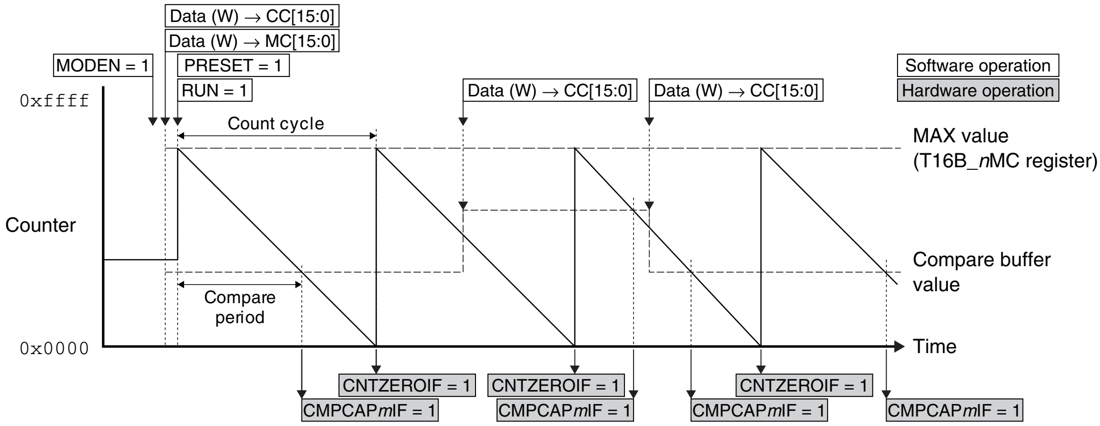


(1.5) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x4

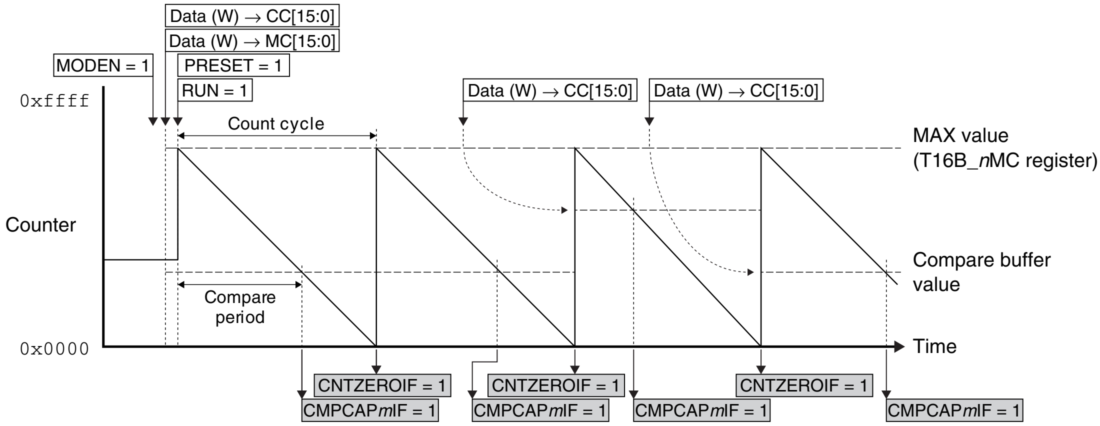


(2) Repeat down count mode

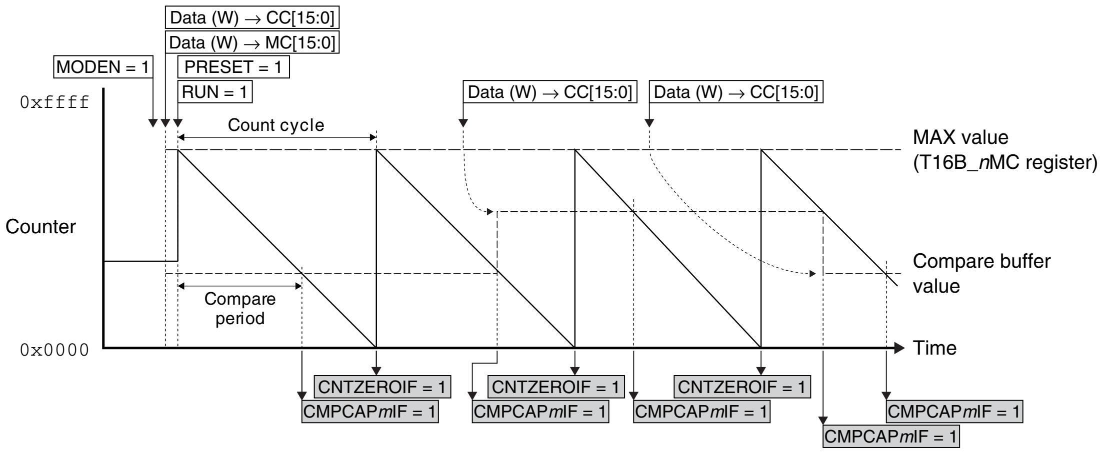
(2.1) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x0



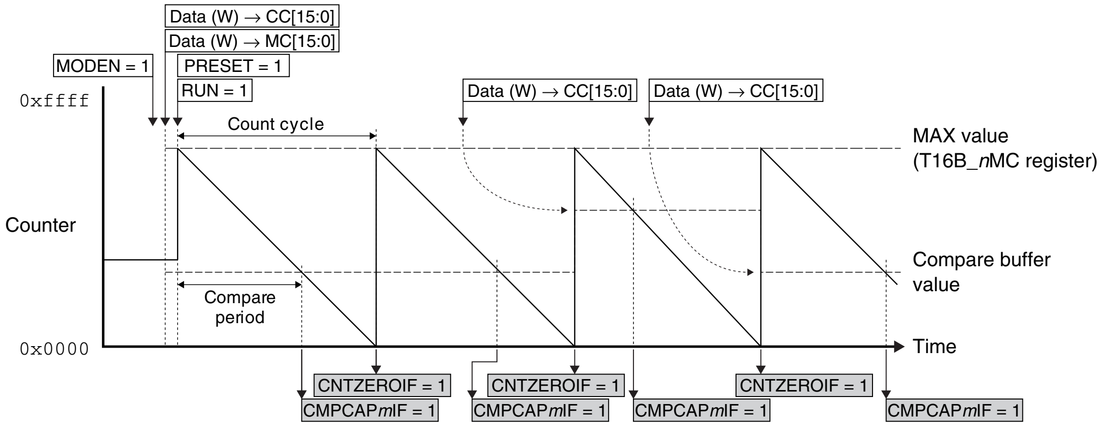
(2.2) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x1



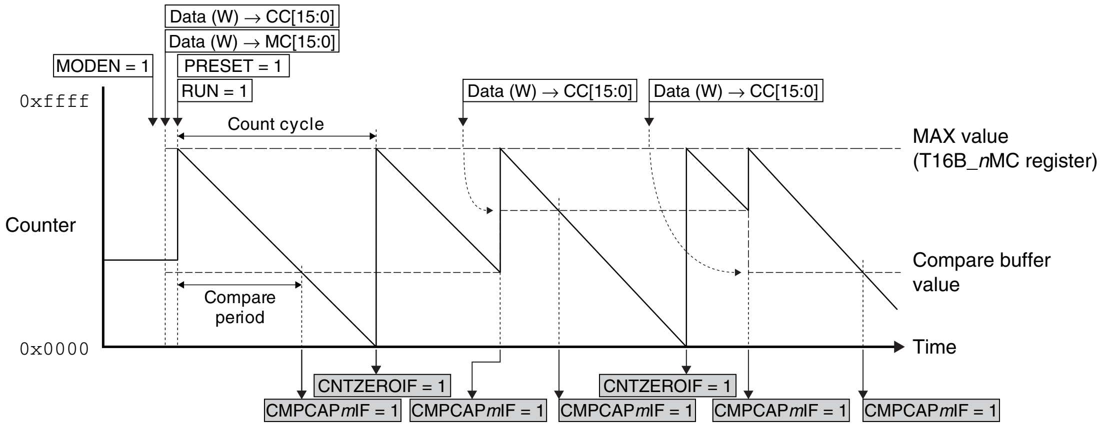
(2.3) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x2



(2.4) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x3

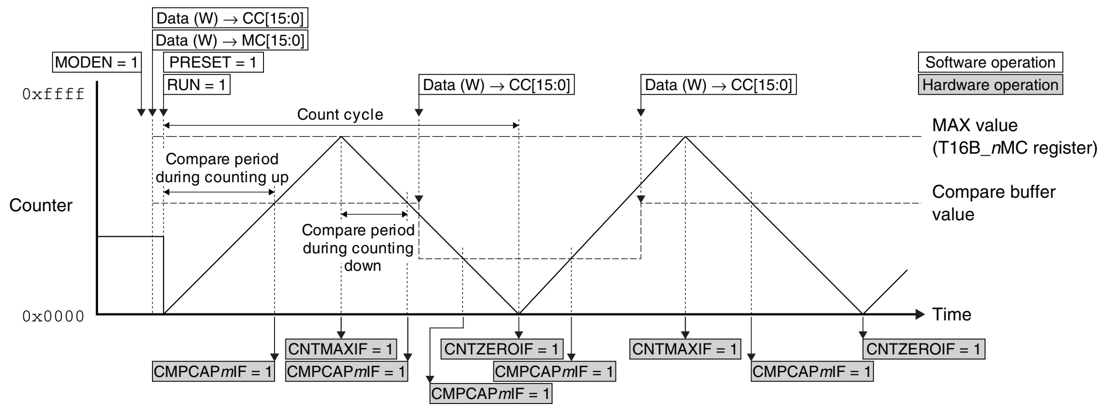


(2.5) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x4

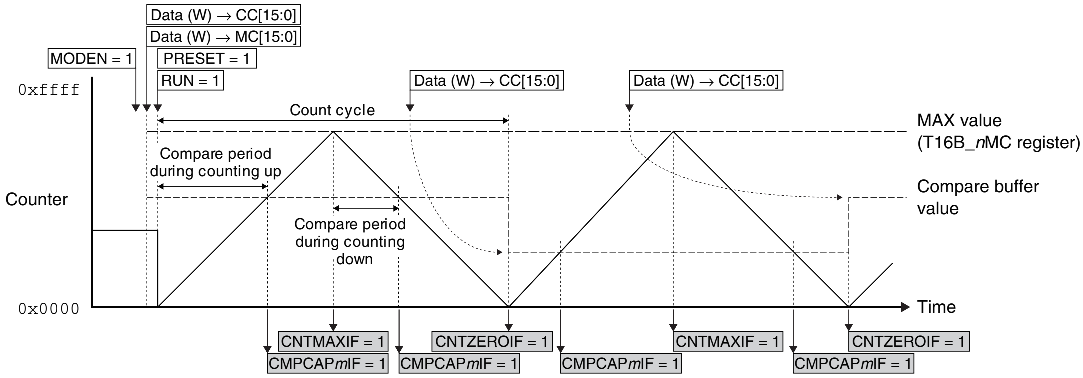


(3) Repeat up/down count mode

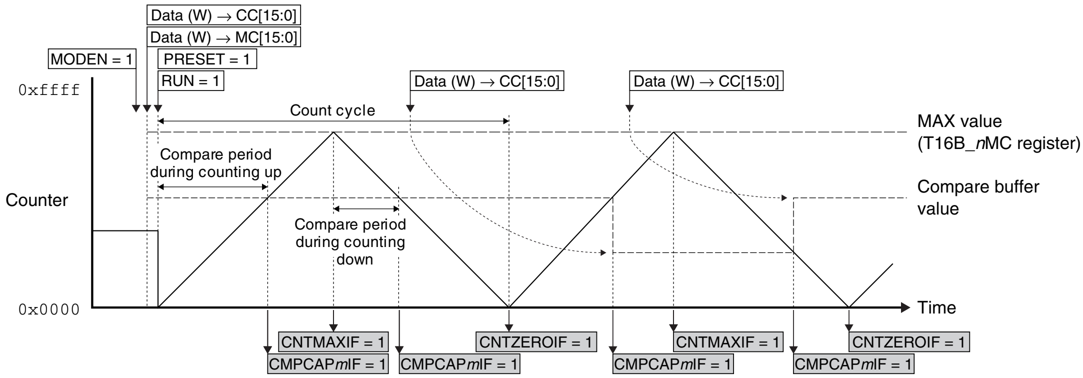
(3.1) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x0



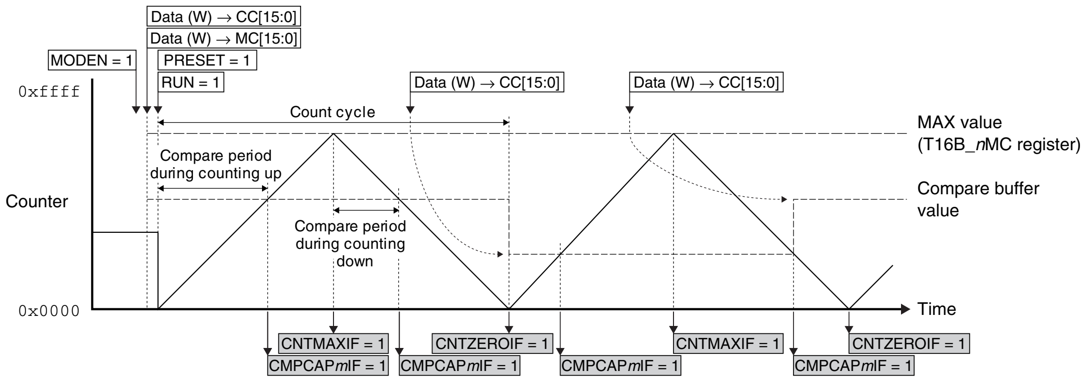
(3.2) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x1



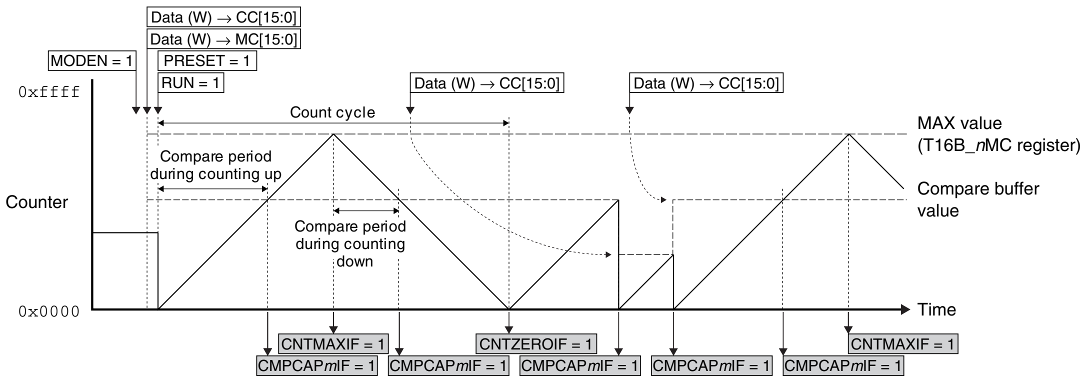
(3.3) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x2



(3.4) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x3



(3.5) T16B_*n*CCCTL*m*.CBUFMD[2:0] bits = 0x4



(Note that the T16B_*n*INTF.CMPCAP*m*IF/CNTMAXIF/CNTZEROIF bit clearing operations via software are omitted from the figure.)

Figure 16.4.3.2 Compare Buffer Operations

Compare period and count cycle settings using DMA

By setting the T16B_*n*CC*m*DMAEN.CC*m*DMAEN*x* bit to 1 (DMA transfer request enabled) in comparator mode, a DMA transfer request is sent to the DMA controller and compare data is transferred from the specified memory to the T16B_*n*CCR*m* register via DMA Ch.*x* when the T16B_*n*INTF.CMPCAP*m*IF bit is set to 1 (when the counter reaches the compare buffer value).

Similarly, by setting the T16B_*n*CC*m*DMAEN.MZDMAEN*x* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and a counter MAX value is transferred from the specified memory to the T16B_*n*MC register via DMA Ch.*x* when the T16B_*n*INTF.CNTMAXIF bit is set to 1 (when the counter reaches the MAX value) in up or up/down count mode, or when the T16B_*n*INTF.CNTZEROIF bit is set to 1 (when the counter reaches zero) in down count mode.

This automates the compare period and count cycle settings of the timer counter.

The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance so that the setting data will be transferred to the T16B_*n*CCR*m* or T16B_*n*MC register. For more information on DMA, refer to the "DMA Controller" chapter.

Table 16.4.3.1 DMA Data Structure Configuration Example (T16B Compare Period and Count Cycle Settings)

Item		Setting example
End pointer	Transfer source	Memory address in which the last setting data is stored
	Transfer destination	T16B_*n*CCR*m* or T16B_*n*MC register address
Control data	dst_inc	0x3 (no increment)
	dst_size	0x1 (haflword)
	src_inc	0x1 (+2)
	src_size	0x1 (halfword)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

Operations in capture mode

The capture mode captures the counter value when an external event, such as a key entry, occurs (at the specified edge of the external input/software trigger signal). In this mode, the T16B_*n*CCR*m* register functions as the capture register from which the captured data is read. Furthermore, the TOUT*nm*/CAP*nm* pin is configured to the CAP*nm* pin.

The trigger signal and the trigger edge to capture the counter value are selected using the T16B_*n*CCCTL*m*.CAPIS[1:0] bits and the T16B_*n*CCCTL*m*.CAPTRG[1:0] bits, respectively.

When a specified trigger edge is input during counting, the current counter value is loaded to the T16B_*n*CCR*m* register. At the same time the T16B_*n*INTF.CMPCAP*m*IF bit is set. The interrupt occurred by this bit can be used to read the captured data from the T16B_*n*CCR*m* register. For example, external event cycles and pulse widths can be measured from the difference between two captured counter values read.

If the captured data stored in the T16B_*n*CCR*m* register is overwritten by the next trigger when the T16B_*n*INTF.CMPCAP*m*IF bit is still set, an overwrite error occurs (the T16B_*n*INTF.CAPOW*m*IF bit is set).

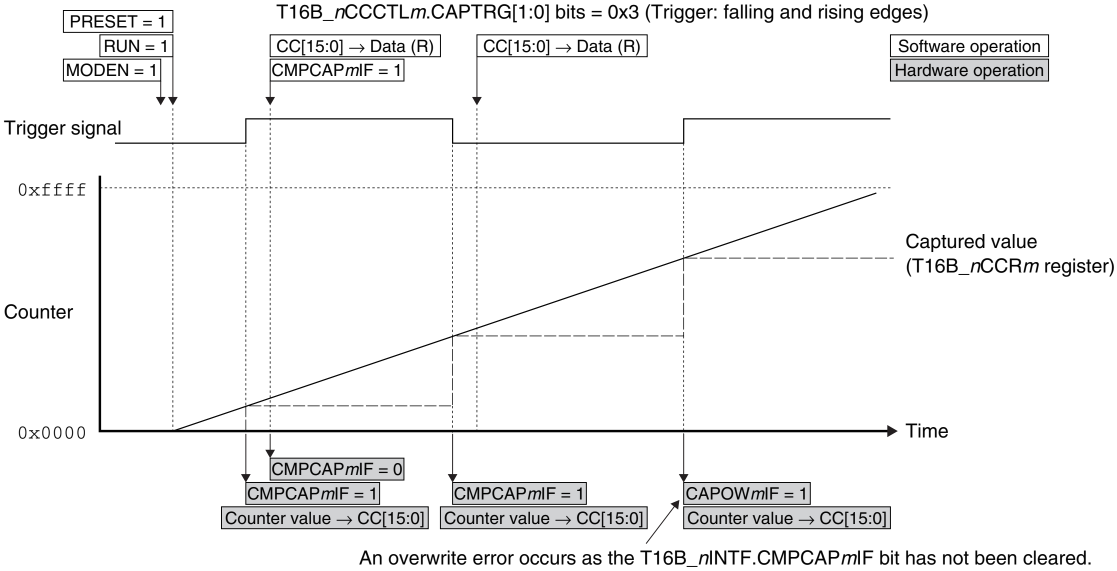


Figure 16.4.3.3 Operations in Capture Mode (Example in One-shot Up Count Mode)

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 T16B_*n*CCCTL*m*.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 T16B_*n*CCCTL*m*.SCS bit to 0 to set the capture circuit to asynchronous capture mode.

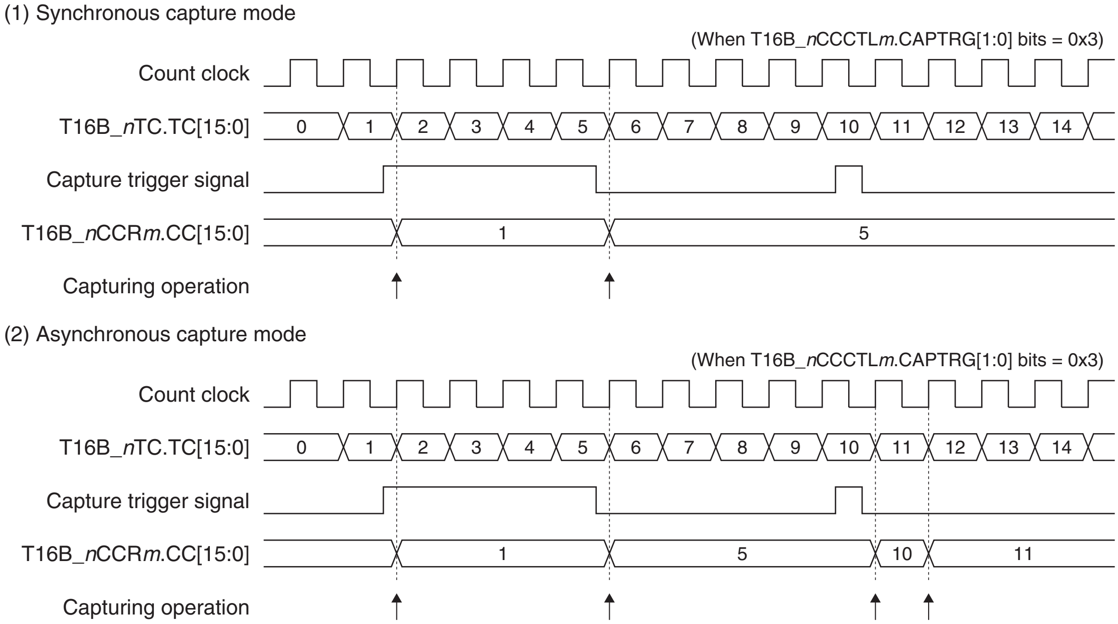


Figure 16.4.3.4 Synchronous Capture Mode/Asynchronous Capture Mode

Capture data transfer using DMA

By setting the T16B_*n*CC*m*DMAEN.CC*m*DMAEN*x* bit to 1 (DMA transfer request enabled) in capture mode, a DMA transfer request is sent to the DMA controller and the T16B_*n*CCR*m* register value is transferred to the specified memory via DMA Ch.*x* when the T16B_*n*INTF.CMPCAP*m*IF bit is set to 1 (when data has been captured).

This automates reading and saving of capture data.

The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance. For more information on DMA, refer to the "DMA Controller" chapter.

Table 16.4.3.2 DMA Data Structure Configuration Example (Capture Data Transfer)

Item		Setting example
End pointer	Transfer source	T16B_*n*CCR*m* register address
	Transfer destination	Memory address to which the last capture data is stored
Control data	dst_inc	0x1 (+2)
	dst_size	0x1 (haflword)
	src_inc	0x3 (no increment)
	src_size	0x1 (halfword)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

16.4.4 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 16.4.4.1 shows the TOUT output circuits (circuits 0 and 1).

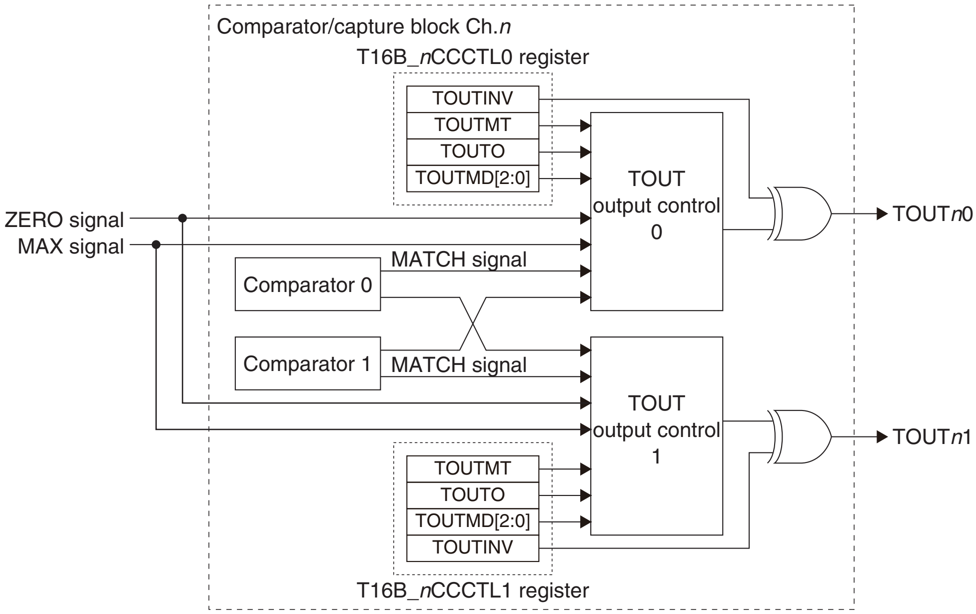


Figure 16.4.4.1 TOUT Output Circuits (Circuits 0 and 1)

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 T16B_*n*CCCTL*m*.TOUTMD[2:0] bits are used to set how the TOUT signal waveform is changed by the MATCH and MAX/ZERO signals.

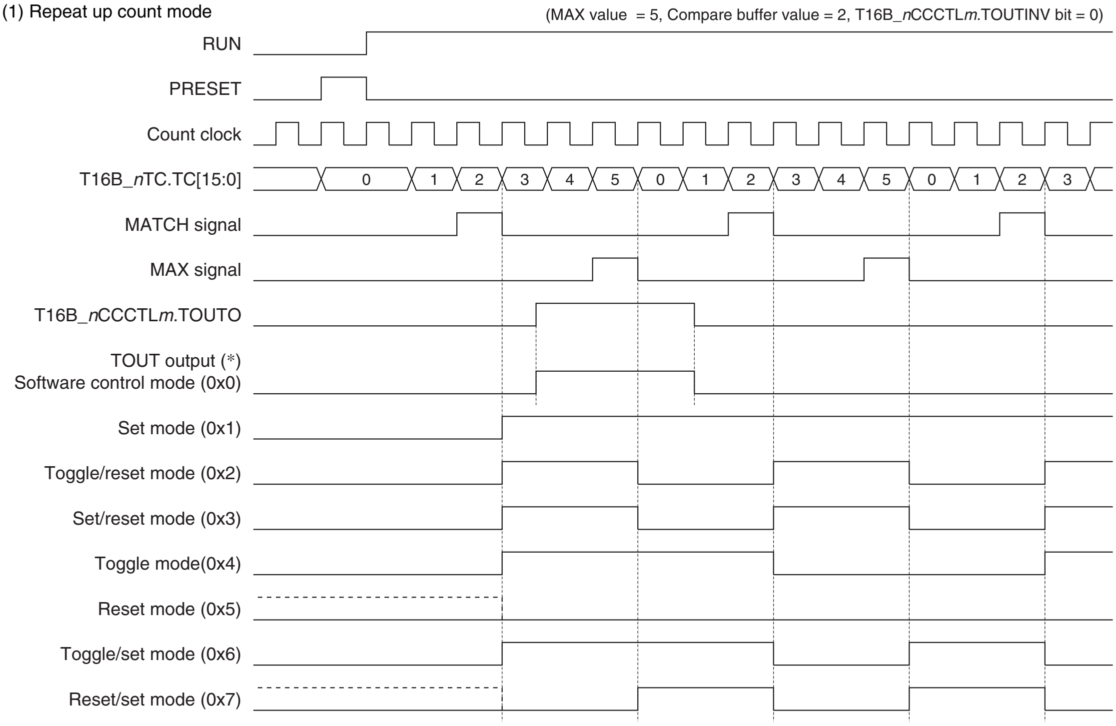
Furthermore, when the T16B_*n*CCCTL*m*.TOUTMT bit is set to 1, the TOUT circuit uses the MATCH signal output from another system in the circuit pair (0 and 1, 2 and 3, 4 and 5). This makes it possible to change the signal twice within a counter cycle.

TOUT signal polarity

The TOUT signal polarity (active level) can be set using the T16B_*n*CCCTL*m*.TOUTINV bit. It is set to active high by setting the T16B_*n*CCCTL*m*.TOUTINV bit to 0 and active low by setting to 1.

Figures 16.4.4.2 and 16.4.4.3 show the TOUT output waveforms.

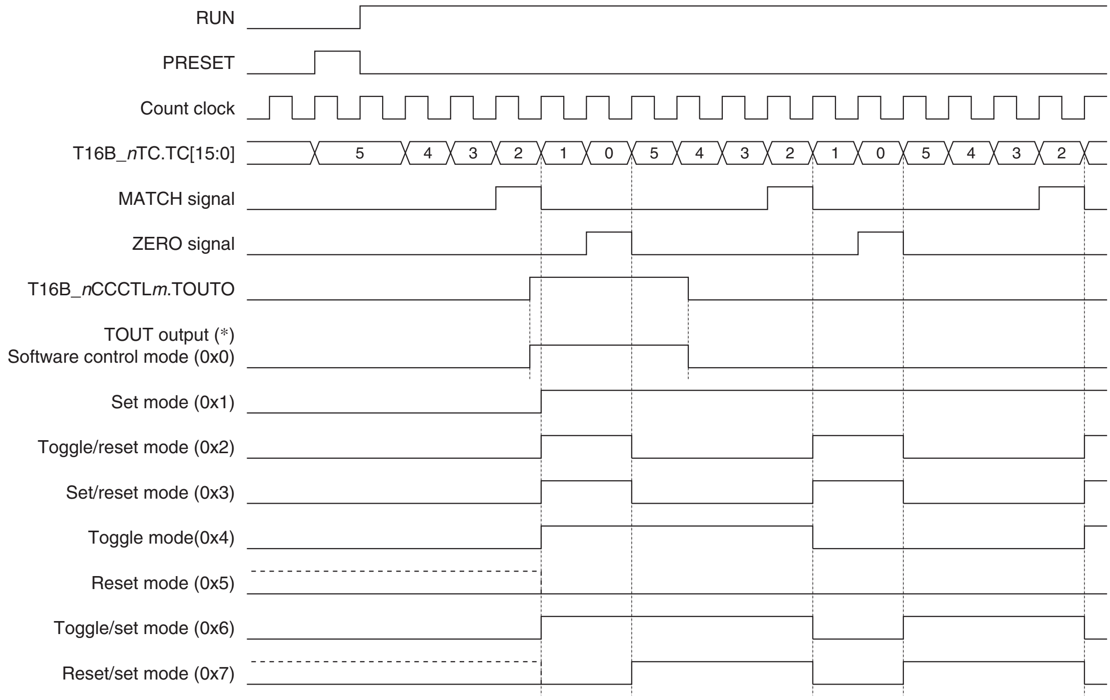
(1) Repeat up count mode



* ( ) indicates the T16B_*n*CCCTL*m*.TOUTMD[2:0] bit-setting value.

(2) Repeat down count mode

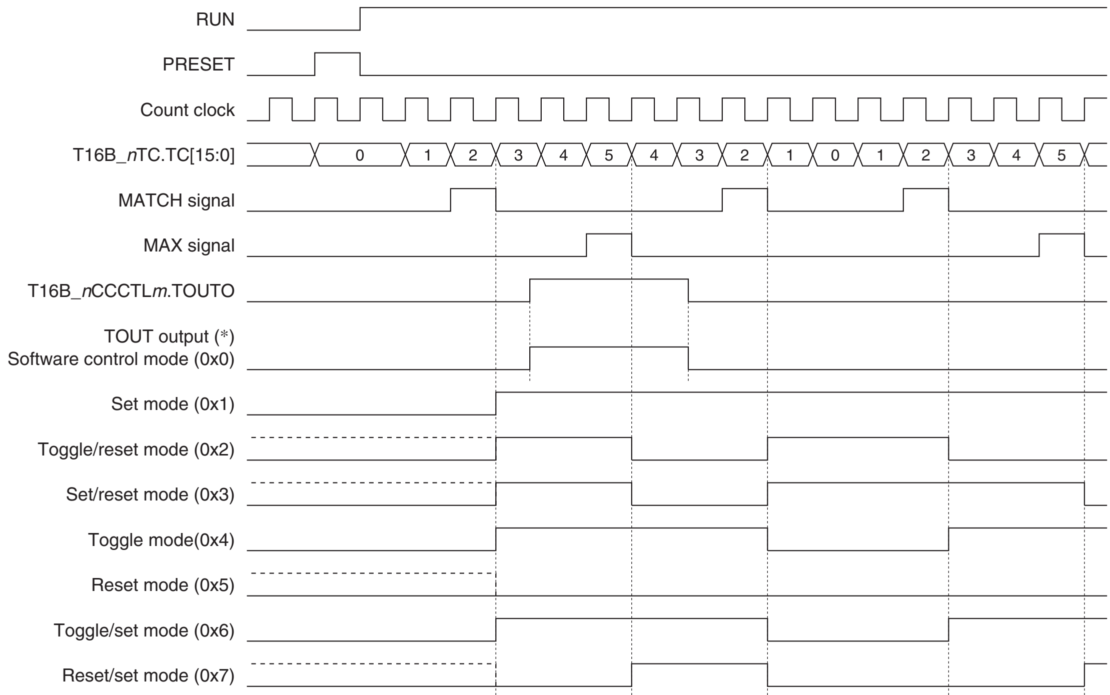
(MAX value = 5, Compare buffer value = 2, T16B_*n*CCCTL*m*.TOUTINV bit = 0)



* ( ) indicates the T16B_*n*CCCTL*m*.TOUTMD[2:0] bit-setting value.

(3) Repeat up/down count mode

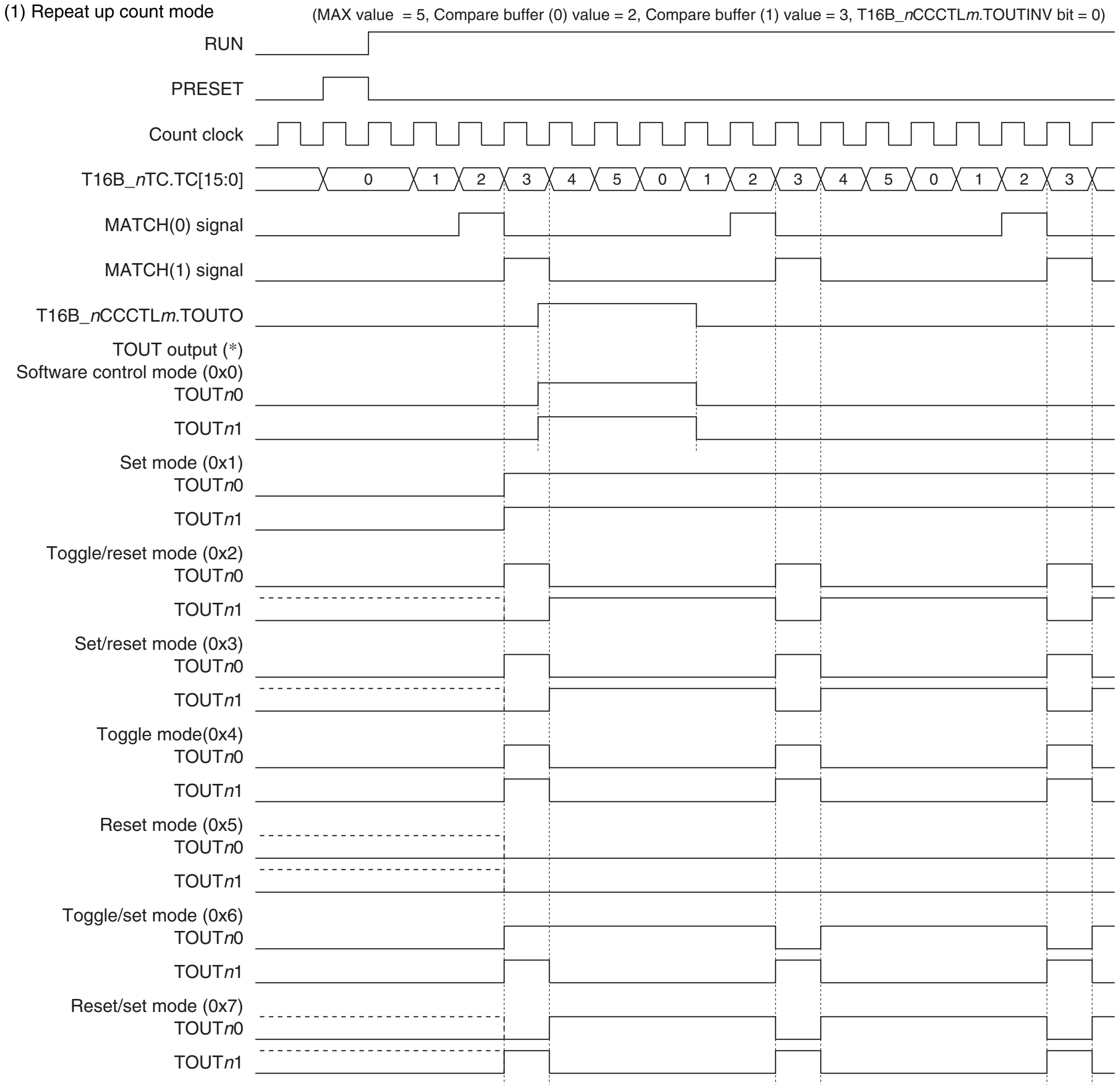
(MAX value = 5, Compare buffer value = 2, T16B_*n*CCCTL*m*.TOUTINV bit = 0)



* ( ) indicates the T16B_*n*CCCTL*m*.TOUTMD[2:0] bit-setting value.

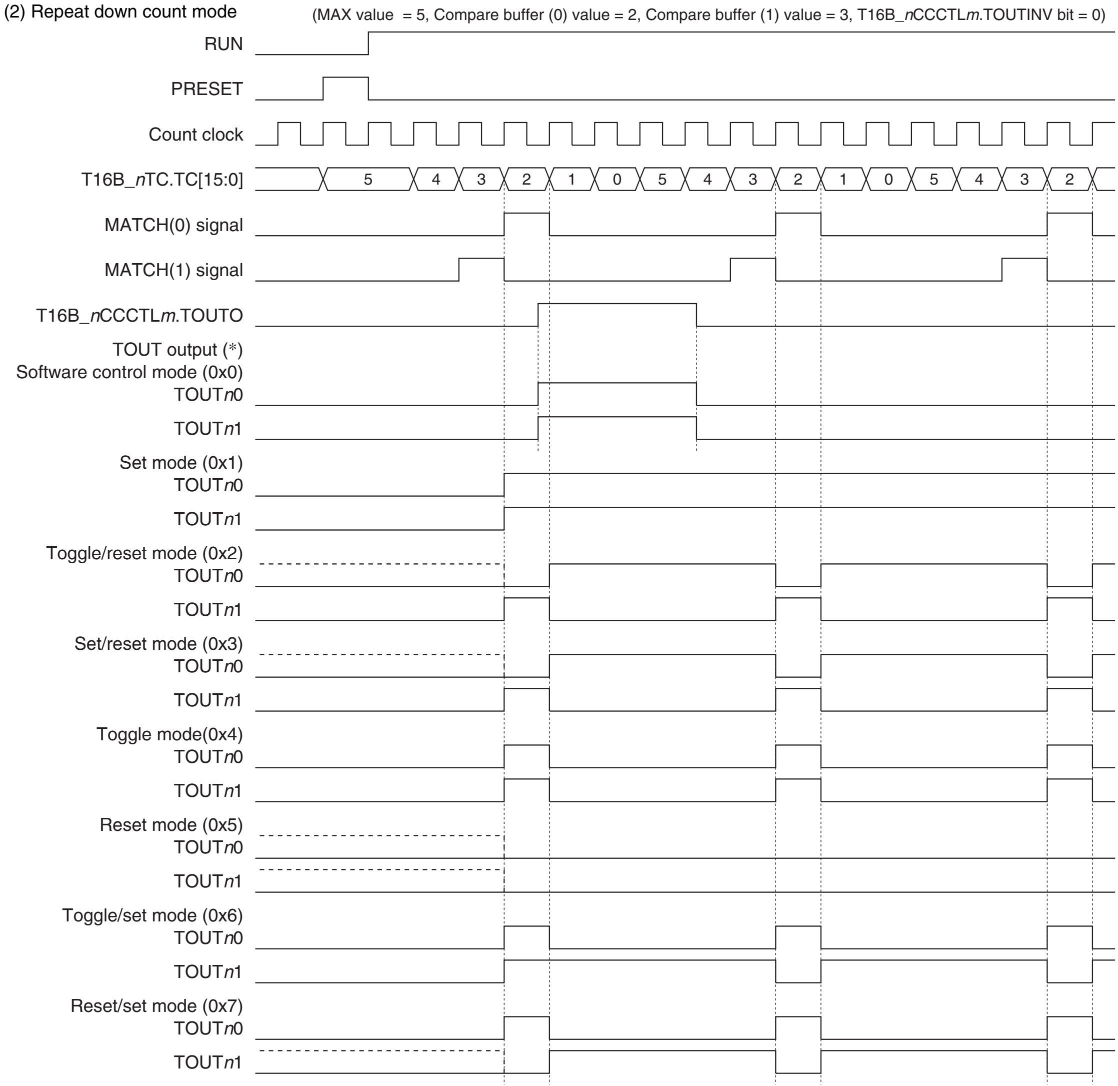
Figure 16.4.4.2 TOUT Output Waveform (T16B_*n*CCCTL*m*.TOUTMT bit = 0)

(1) Repeat up count mode



* ( ) indicates the T16B_nCCCTLm.TOUTMD[2:0] bit-setting value.

(2) Repeat down count mode



* ( ) indicates the T16B_*n*CCCTL*m*.TOUTMD[2:0] bit-setting value.

(3) Repeat up/down count mode

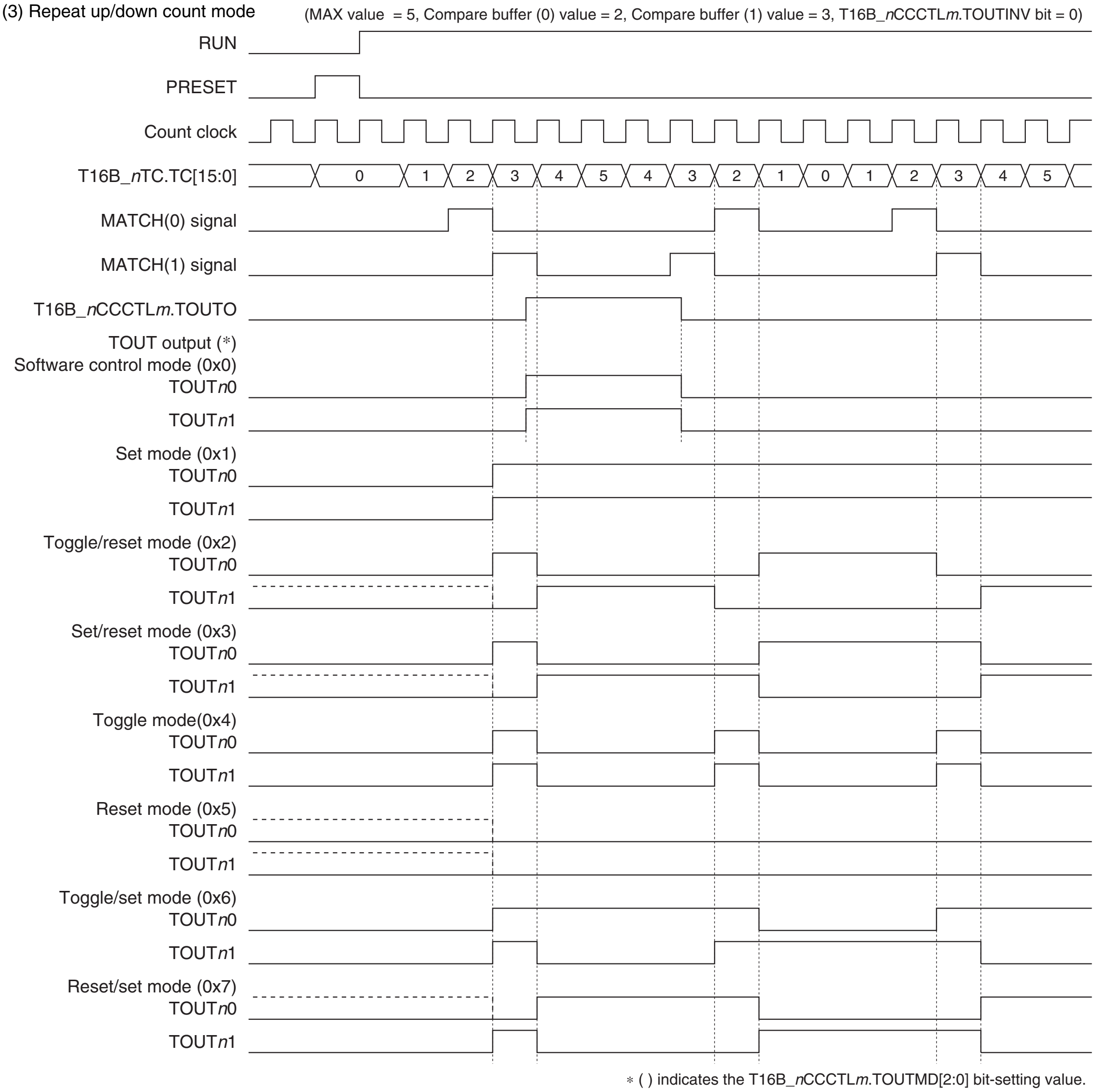


Figure 16.4.4.3 TOUT Output Waveform (T16B_*n*CCCTL0.TOUTMT bit = 1, T16B_*n*CCCTL1.TOUTMT bit = 0)

16.5 Interrupt

Each T16B channel has a function to generate the interrupt shown in Table 16.5.1.

Table 16.5.1 T16B Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Capture overwrite	T16B_*n*INTF.CAPOW*m*IF	When the T16B_*n*INTF.CMPCAP*m*IF bit =1 and the T16B_*n* CCR*m* register is overwritten with new captured data in capture mode	Writing 1
Compare/ capture	T16B_*n*INTF.CMPCAP*m*IF	When the counter value becomes equal to the compare buffer value in comparator mode When the counter value is loaded to the T16B_*n*CCR*m* register by a capture trigger input in capture mode	Writing 1
Counter MAX	T16B_*n*INTF.CNTMAXIF	When the counter reaches the MAX value	Writing 1
Counter zero	T16B_*n*INTF.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 CPU 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" chapter.

16.6 DMA Transfer Requests

The T16B has a function to generate DMA transfer requests from the causes shown in Table 16.6.1.

Table 16.6.1 DMA Transfer Request Causes of T16B

Cause to request DMA transfer	DMA transfer request flag	Set condition	Clear condition
Compare/ capture	Compare/capture flag (T16B_*n*INTF.CMPCAP*m*IF)	When the counter value becomes equal to the compare buffer value in comparator mode When the counter value is loaded to the T16B_*n*CCR*m* register by a capture trigger input in capture mode	When the DMA transfer request is accepted
Counter MAX/ zero	Counter MAX flag (T16B_*n*INTF.CNTMAXIF) Counter zero flag (T16B_*n*INTF.CNTZEROIF)	When the counter reaches the MAX value in up or up/down count mode When the counter reaches 0x0000 in down count mode	When the DMA transfer request is accepted

The T16B provides DMA transfer request enable bits corresponding to each DMA transfer request flag shown above for the number of DMA channels. A DMA transfer request is sent to the pertinent channel of the DMA controller only when the DMA transfer request flag, of which DMA transfer has been enabled by the DMA transfer request enable bit, is set. The DMA transfer request flag also serves as an interrupt flag, therefore, both the DMA transfer request and the interrupt cannot be enabled at the same time. After a DMA transfer has completed, disable the DMA transfer to prevent unintended DMA transfer requests from being issued. For more information on the DMA control, refer to the "DMA Controller" chapter.

16.7 Control Registers

T16B Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*CLK	15–9	–	0x00	–	R	–
	8	DBRUN	0	H0	R/W	
	7–4	CLKDIV[3:0]	0x0	H0	R/W	
	3	–	0	–	R	
	2–0	CLKSRC[2:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the T16B Ch.*n* operating clock is supplied during debugging or not.

1 (R/W): Clock supplied during debugging

0 (R/W): No clock supplied during debugging

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 16.7.1 Clock Source and Division Ratio Settings

T16B_*n*CLK.CLKDIV[3:0] bits	T16B_*n*CLK.CLKSRC[2:0] bits							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
	IOSC	OSC1	OSC3	EXOSC	EXCL*n*0	EXCL*n*1	EXCL*n*0 inverted input	EXCL*n*1 inverted input
0xf	1/32,768	1/1	1/32,768	1/1	1/1	1/1	1/1	1/1
0xe	1/16,384		1/16,384					
0xd	1/8,192		1/8,192					
0xc	1/4,096		1/4,096					
0xb	1/2,048		1/2,048					
0xa	1/1,024		1/1,024					
0x9	1/512		1/512					
0x8	1/256	1/256	1/256					
0x7	1/128	1/128	1/128					
0x6	1/64	1/64	1/64					
0x5	1/32	1/32	1/32					
0x4	1/16	1/16	1/16					
0x3	1/8	1/8	1/8					
0x2	1/4	1/4	1/4					
0x1	1/2	1/2	1/2					
0x0	1/1	1/1	1/1					

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

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 16.7.2).

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 16.7.2).

Table 16.7.2 Count Mode

T16B_*n*CTL.CNTMD[1:0] bits	Count mode	
	T16B_*n*CTL.ONEST bit = 1	T16B_*n*CTL.ONEST bit = 0
0x3	Reserved	
0x2	One-shot up/down count mode	Repeat up/down count mode
0x1	One-shot down count mode	Repeat down count mode
0x0	One-shot up count mode	Repeat up count mode

Bit 2 RUN

This bit starts/stops counting.

1 (W): Start counting
0 (W): Stop counting
1 (R): Counting
0 (R): Idle

By writing 1 to this bit, the counter block starts count operations. However, the T16B_*n*CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance. While the timer is running, writing 0 to the T16B_*n*CTL.RUN bit stops count operations. When the counter stops by the counter MAX/ZERO signal in one-shot mode, this bit is automatically cleared to 0.

Bit 1 PRESET

This bit resets the counter.

1 (W): Reset
0 (W): Ineffective
1 (R): Resetting in progress
0 (R): Resetting finished or normal operation

In up mode or up/down mode, the counter is cleared to 0x0000 by writing 1 to this bit. In down mode, the MAX value, which has been set to the T16B_*n*MC register, is preset to the counter. However, the T16B_*n*CTL.MODEN bit must be set to 1 in conjunction with this bit or it must be set in advance.

Bit 0 MODEN

This bit enables the T16B Ch.*n* operations.

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

Note: The counter reset operation using the T16B_*n*CTL.PRESET bit and the counting start operation using the T16B_*n*CTL.RUN bit take effect only when the T16B_*n*CTL.MODEN bit = 1.

T16B Ch.*n* Max Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*MC	15–0	MC[15:0]	0xffff	H0	R/W	–

Bits 15–0 MC[15:0]

These bits are used to set the MAX value to preset to the counter. For more information, refer to "Counter Block Operations - MAX counter data register."

Notes:

- When one-shot mode is selected, do not alter the T16B_*n*MC.MC[15:0] bits (MAX value) during counting.
- Make sure the T16B_*n*CTL.MODEN bit is set to 1 before writing data to the T16B_*n*MC.MC[15:0] bits. If the T16B_*n*CTL.MODEN bit = 0 when writing to the T16B_*n*MC.MC[15:0] bits, set the T16B_*n*CTL.MODEN bit to 1 until the T16B_*n*CS.BSY bit is set to 0 from 1.
- Do not set the T16B_*n*MC.MC[15:0] bits to 0x0000.

T16B Ch.*n* Timer Counter Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*TC	15–0	TC[15:0]	0x0000	H0	R	–

Bits 15–0 TC[15:0]

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

T16B Ch.*n* Counter Status Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*CS	15–8	–	0x00	–	R	–
	7	CAPI5	0	H0	R	
	6	CAPI4	0	H0	R	
	5	CAPI3	0	H0	R	
	4	CAPI2	0	H0	R	
	3	CAPI1	0	H0	R	
	2	CAPI0	0	H0	R	
	1	UP_DOWN	1	H0	R	
	0	BSY	0	H0	R	

Bits 15–8 Reserved

Bit 7 CAPI5
Bit 6 CAPI4
Bit 5 CAPI3
Bit 4 CAPI2
Bit 3 CAPI1
Bit 2 CAPI0

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.CAPI5 bit: CAP*n*5 pin
T16B_*n*CS.CAPI4 bit: CAP*n*4 pin
T16B_*n*CS.CAPI3 bit: CAP*n*3 pin
T16B_*n*CS.CAPI2 bit: CAP*n*2 pin
T16B_*n*CS.CAPI1 bit: CAP*n*1 pin
T16B_*n*CS.CAPI0 bit: CAP*n*0 pin

Note: The configuration of the T16B_*n*CS.CAPI*m* bits depends on the model. The bits corresponding to the CAP*nm* pins that do not exist are read-only bits and are always fixed at 0.

Bit 1 UP_DOWN

This bit indicates the currently set count direction.

1 (R): Count up
0 (R): Count down

Bit 0 BSY

This bit indicates the counter operating status.

1 (R): Running
0 (R): Idle

T16B Ch.*n* Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*INTF	15–14	–	0x0	–	R	–
	13	CAPOW5IF	0	H0	R/W	Cleared by writing 1.
	12	CMPCAP5IF	0	H0	R/W	
	11	CAPOW4IF	0	H0	R/W	
	10	CMPCAP4IF	0	H0	R/W	
	9	CAPOW3IF	0	H0	R/W	
	8	CMPCAP3IF	0	H0	R/W	
	7	CAPOW2IF	0	H0	R/W	
	6	CMPCAP2IF	0	H0	R/W	
	5	CAPOW1IF	0	H0	R/W	
	4	CMPCAP1IF	0	H0	R/W	
	3	CAPOW0IF	0	H0	R/W	
	2	CMPCAP0IF	0	H0	R/W	
	1	CNTMAXIF	0	H0	R/W	
	0	CNTZEROIF	0	H0	R/W	

Bits 15–14 Reserved

Bit 13 CAPOW5IF
Bit 12 CMPCAP5IF
Bit 11 CAPOW4IF
Bit 10 CMPCAP4IF
Bit 9 CAPOW3IF
Bit 8 CMPCAP3IF
Bit 7 CAPOW2IF
Bit 6 CMPCAP2IF
Bit 5 CAPOW1IF
Bit 4 CMPCAP1IF
Bit 3 CAPOW0IF
Bit 2 CMPCAP0IF
Bit 1 CNTMAXIF
Bit 0 CNTZEROIF

These bits indicate the T16B Ch.*n* interrupt cause occurrence status.

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

The following shows the correspondence between the bit and interrupt:

T16B_*n*INTF.CAPOW5IF bit: Capture 5 overwrite interrupt
T16B_*n*INTF.CMPCAP5IF bit: Compare/capture 5 interrupt
T16B_*n*INTF.CAPOW4IF bit: Capture 4 overwrite interrupt
T16B_*n*INTF.CMPCAP4IF bit: Compare/capture 4 interrupt
T16B_*n*INTF.CAPOW3IF bit: Capture 3 overwrite interrupt
T16B_*n*INTF.CMPCAP3IF bit: Compare/capture 3 interrupt
T16B_*n*INTF.CAPOW2IF bit: Capture 2 overwrite interrupt
T16B_*n*INTF.CMPCAP2IF bit: Compare/capture 2 interrupt
T16B_*n*INTF.CAPOW1IF bit: Capture 1 overwrite interrupt
T16B_*n*INTF.CMPCAP1IF bit: Compare/capture 1 interrupt
T16B_*n*INTF.CAPOW0IF bit: Capture 0 overwrite interrupt
T16B_*n*INTF.CMPCAP0IF bit: Compare/capture 0 interrupt
T16B_*n*INTF.CNTMAXIF bit: Counter MAX interrupt
T16B_*n*INTF.CNTZEROIF bit: Counter zero interrupt

Note: The configuration of the T16B_*n*INTF.CAPOW*m*IF and T16B_*n*INTF.CMPCAP*m*IF bits depends on the model. The bits corresponding to the comparator/capture circuits that do not exist are read-only bits and are always fixed at 0.

T16B Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*INTE	15–14	–	0x0	–	R	–
	13	CAPOW5IE	0	H0	R/W	
	12	CMPCAP5IE	0	H0	R/W	
	11	CAPOW4IE	0	H0	R/W	
	10	CMPCAP4IE	0	H0	R/W	
	9	CAPOW3IE	0	H0	R/W	
	8	CMPCAP3IE	0	H0	R/W	
	7	CAPOW2IE	0	H0	R/W	
	6	CMPCAP2IE	0	H0	R/W	
	5	CAPOW1IE	0	H0	R/W	
	4	CMPCAP1IE	0	H0	R/W	
	3	CAPOW0IE	0	H0	R/W	
	2	CMPCAP0IE	0	H0	R/W	
	1	CNTMAXIE	0	H0	R/W	
	0	CNTZEROIE	0	H0	R/W	

Bits 15–14 Reserved

Bit 13 CAPOW5IE
Bit 12 CMPCAP5IE
Bit 11 CAPOW4IE
Bit 10 CMPCAP4IE
Bit 9 CAPOW3IE
Bit 8 CMPCAP3IE
Bit 7 CAPOW2IE
Bit 6 CMPCAP2IE
Bit 5 CAPOW1IE
Bit 4 CMPCAP1IE
Bit 3 CAPOW0IE
Bit 2 CMPCAP0IE
Bit 1 CNTMAXIE
Bit 0 CNTZEROIE

These bits enable T16B Ch.*n* interrupts.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

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_*n*CCCTL*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_*n*CCR*m* register to the compare buffer. The T16B_*n*CCCTL*m*.CBUFMD[2:0] bits are control bits for comparator mode and are ineffective in capture mode.

Table 16.7.3 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 16.7.4). The T16B_*n*CCCTL*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_*n*CCR*m* register in capture mode (see Table 16.7.4). The T16B_*n*CCCTL*m*.CAPTRG[1:0] bits are control bits for capture mode and are ineffective in comparator mode.

Table 16.7.4 Trigger Signal/Edge for Capturing Counter Value

T16B_*n*CCCTL*m*. CAPTRG[1:0] bits (Trigger edge)	**Trigger condition**		
	T16B_*n*CCCTL*m*.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 CAP*nm* pin input signal	Altering the T16B_*n*CCCTL*m*.CAPIS[1:0] bits from 0x2 to 0x3, or from 0x3 to 0x2	
0x2 (↓)	Falling edge of the CAP*nm* pin input signal	Altering the T16B_*n*CCCTL*m*.CAPIS[1:0] bits from 0x3 to 0x2	
0x1 (↑)	Rising edge of the CAP*nm* pin input signal	Altering the T16B_*n*CCCTL*m*.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 TOUT*nm* 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 T16B_*n*CCCTL*m*.TOUTMT bit is control bit for comparator mode and is ineffective in capture mode.

Bit 5 TOUTO

This bit sets the TOUT*nm* signal output level when software control mode (T16B_*n*CCCTL*m*.TOUTMD[2:0] = 0x0) is selected for the TOUT*nm* output.

1 (R/W): High level output

0 (R/W): Low level output

The T16B_*n*CCCTL*m*.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 TOUT*nm* signal waveform is changed by the comparator MATCH and counter MAX/ZERO signals.

The T16B_*n*CCCTL*m*.TOUTMD[2:0] bits are control bits for comparator mode and are ineffective in capture mode.

Table 16.7.5 TOUT Generation Mode

T16B_*n*CCCTL*m*. TOUTMD[2:0] bits	**TOUT generation mode and operations**			
	T16B_*n*CCCTL*m*. TOUTMT bit	**Count mode**	**Output signal**	**Change in the signal**
0x7	**Reset/set mode**			
	0	Up count mode Up/down count mode	TOUT*nm*	The signal becomes inactive by the MATCH signal and it becomes active by the MAX signal.
		Down count mode	TOUT*nm*	The signal becomes inactive by the MATCH signal and it becomes active by the ZERO signal.
	1	All count modes	TOUT*nm*	The signal becomes inactive by the MATCH*m* signal and it becomes active by the MATCH*m*+1 signal.
			TOUT*nm*+1	The signal becomes inactive by the MATCH*m*+1 signal and it becomes active by the MATCH*m* signal.
0x6	**Toggle/set mode**			
	0	Up count mode Up/down count mode	TOUT*nm*	The signal is inverted by the MATCH signal and it becomes active by the MAX signal.
		Down count mode	TOUT*nm*	The signal is inverted by the MATCH signal and it becomes active by the ZERO signal.
	1	All count modes	TOUT*nm*	The signal is inverted by the MATCH*m* signal and it becomes active by the MATCH*m*+1 signal.
			TOUT*nm*+1	The signal is inverted by the MATCH*m*+1 signal and it becomes active by the MATCH*m* signal.
0x5	**Reset mode**			
	0	All count modes	TOUT*nm*	The signal becomes inactive by the MATCH signal.
	1	All count modes	TOUT*nm*	The signal becomes inactive by the MATCH*m* or MATCH*m*+1 signal.
			TOUT*nm*+1	The signal becomes inactive by the MATCH*m*+1 or MATCH*m* signal.

T16B_*n*CCCTL*m*.TOUTMD[2:0] bits	TOUT generation mode and operations			
	T16B_*n*CCCTL*m*.TOUTMT bit	Count mode	Output signal	Change in the signal
0x4	**Toggle mode**			
	0	All count modes	TOUT*nm*	The signal is inverted by the MATCH signal.
	1	All count modes	TOUT*nm*	The signal is inverted by the MATCH*m* or MATCH*m*+1 signal.
			TOUT*nm*+1	The signal is inverted by the MATCH*m*+1 or MATCH*m* signal.
0x3	**Set/reset mode**			
	0	Up count mode Up/down count mode	TOUT*nm*	The signal becomes active by the MATCH signal and it becomes inactive by the MAX signal.
		Down count mode	TOUT*nm*	The signal becomes active by the MATCH signal and it becomes inactive by the ZERO signal.
	1	All count modes	TOUT*nm*	The signal becomes active by the MATCH*m* signal and it becomes inactive by the MATCH*m*+1 signal.
			TOUT*nm*+1	The signal becomes active by the MATCH*m*+1 signal and it becomes inactive by the MATCH*m* signal.
0x2	**Toggle/reset mode**			
	0	Up count mode Up/down count mode	TOUT*nm*	The signal is inverted by the MATCH signal and it becomes inactive by the MAX signal.
		Down count mode	TOUT*nm*	The signal is inverted by the MATCH signal and it becomes inactive by the ZERO signal.
	1	All count modes	TOUT*nm*	The signal is inverted by the MATCH*m* signal and it becomes inactive by the MATCH*m*+1 signal.
			TOUT*nm*+1	The signal is inverted by the MATCH*m*+1 signal and it becomes inactive by the MATCH*m* signal.
0x1	**Set mode**			
	0	All count modes	TOUT*nm*	The signal becomes active by the MATCH signal.
	1	All count modes	TOUT*nm*	The signal becomes active by the MATCH*m* or MATCH*m*+1 signal.
			TOUT*nm*+1	The signal becomes active by the MATCH*m*+1 or MATCH*m* signal.
0x0	**Software control mode**			
	*	All count modes	TOUT*nm*	The signal becomes active by setting the T16B_*n*CCCTL*m*.TOUTO bit to 1 and it becomes inactive by setting to 0.

Bit 1 TOUTINV

This bit selects the TOUT*nm* signal polarity.

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

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

The T16B_*n*CCCTL*m*.TOUTINV bit is control bit for comparator mode and is ineffective in capture mode.

Bit 0 CCMD

This bit selects the operating mode of the comparator/capture circuit *m*.

1 (R/W): Capture mode (T16B_*n*CCR*m* register = capture register)

0 (R/W): Comparator mode (T16B_*n*CCR*m* register = compare data register)

T16B Ch.*n* Compare/Capture *m* Data Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*CCR*m*	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.

T16B Ch.*n* Counter Max/Zero DMA Request Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*MZDMAEN	15–0	MZDMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 MZDMAEN[15:0]

These bits enable T16B to issue a DMA transfer request to the corresponding DMA controller channel (Ch.0–Ch.15) when the counter value reaches the MAX value or 0x0000.

1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

T16B Ch.*n* Compare/Capture *m* DMA Request Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
T16B_*n*CC*m*DMAEN	15–0	CC*m*DMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 CC*m*DMAEN[15:0]

These bits enable T16B to issue a DMA transfer request to the corresponding DMA controller channel (Ch.0–Ch.15) when the counter value reaches the compare data or is captured.

1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

17 Sound Generator (SNDA)

17.1 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 17.1.1 shows the SNDA configuration.

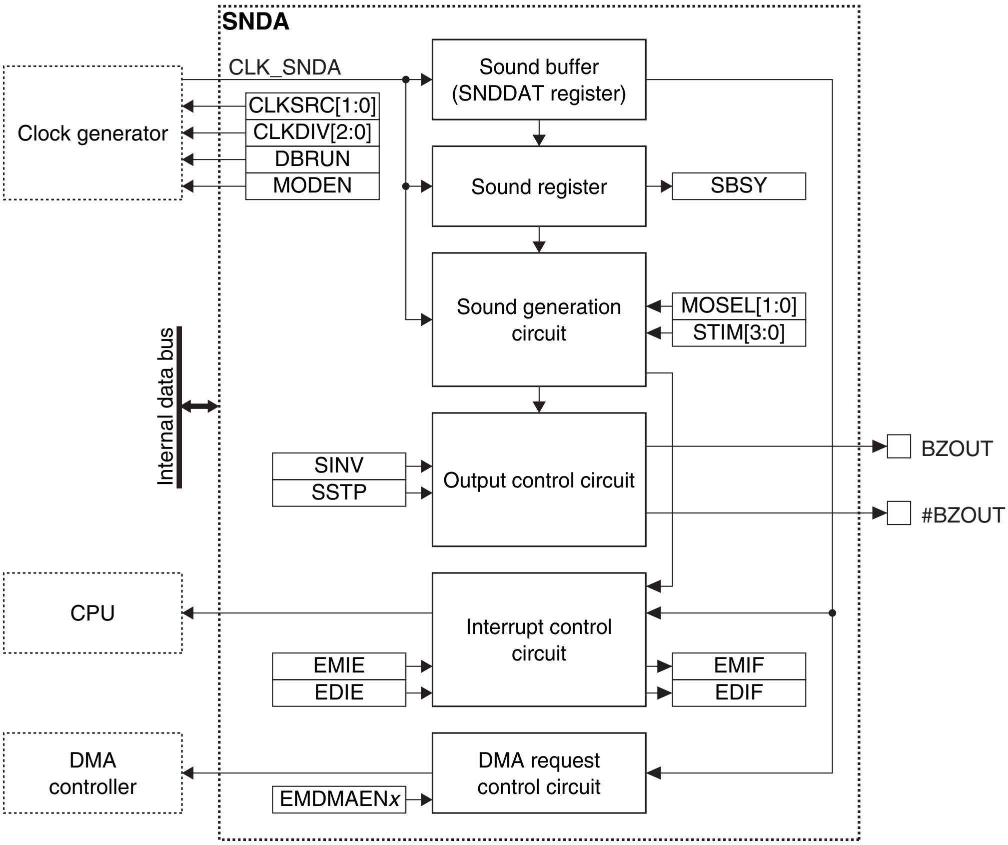


Figure 17.1.1 SNDA Configuration

17.2 Output Pins and External Connections

17.2.1 List of Output Pins

Table 17.2.1.1 lists the SNDA pins.

Table 17.2.1.1 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.

17.2.2 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 SNDASEL.SINV bit.

Direct drive mode (SNDASEL.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 (SNDASEL.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.

17.2.3 External Connections

Figures 17.2.2.1 and 17.2.2.2 show connection diagrams between SNDA and a piezoelectric buzzer.

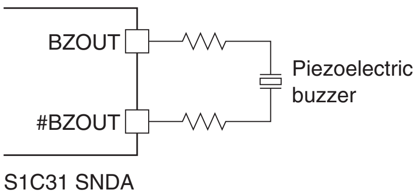


Figure 17.2.2.1 Connection between SNDA and Piezoelectric Buzzer (Direct Drive)

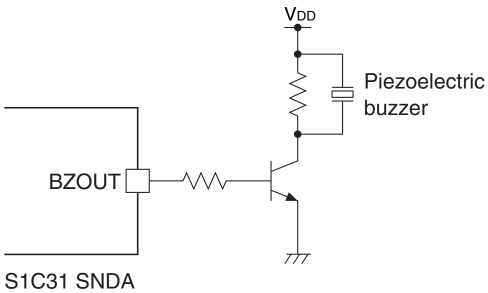


Figure 17.2.2.2 Connection between SNDA and Piezoelectric Buzzer (Single Pin Drive)

17.3 Clock Settings

17.3.1 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 SNDACLK register bits:
   - SNDACLK.CLKSRC[1:0] bits (Clock source selection)
   - SNDACLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)

The CLK_SNDA frequency should be set to around 32,768 Hz.

17.3.2 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.*xxxx*SLPC bit for the CLK_SNDA clock source.
If the CLGOSC.*xxxx*SLPC 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.

17.3.3 Clock Supply in DEBUG Mode

The CLK_SNDA supply during DEBUG mode should be controlled using the SNDACLK.DBRUN bit.
The CLK_SNDA supply to SNDA is suspended when the CPU enters DEBUG mode if the SNDACLK.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 SNDACLK.DBRUN bit = 1, the CLK_SNDA supply is not suspended and SNDA will keep operating in DEBUG mode.

17.4 Operations

17.4.1 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 SNDACTL.MODEN bit to 1. (Enable SNDA operations)
4. Set the following SNDASEL register bits:
   - Set the SNDASEL.SINV bit (Set output pin drive mode)
   - Set the SNDASEL.MOSEL[1:0] bits (Set sound output mode)
5. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the SNDAINTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the SNDAINTE register to 1. (Enable interrupts)
6. Configure the DMA controller and set the following SNDA control bits when using DMA transfer:
   - Write 1 to the DMA transfer request enable bits in the SNDAEMDMAEN register. (Enable DMA transfer requests)

17.4.2 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 SNDASEL.MOSEL[1:0] bits to 0x0. (Set normal buzzer mode)
2. Write data to the following sound buffer (SNDADAT register) bits. (Start buzzer output)
   - SNDADAT.SLEN[5:0] bits (Set buzzer output signal duty ratio)
   - SNDADAT.SFRQ[7:0] bits (Set buzzer output signal frequency)
3. Write 1 to the SNDACTL.SSTP bit after the output period has elapsed. (Stop buzzer output)

Normal buzzer output operations

When data is written to the sound buffer (SNDADAT register), SNDA clears the SNDAINTF.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 SNDAINTF.EMIF bit and SNDAINTF.SBSY bit are both set to 1. The output pin outputs the buzzer signal with the frequency/duty ratio specified.

Writing 1 to the SNDACTL.SSTP bit stops buzzer output and sets the SNDAINTF.EDIF bit (sound output completion interrupt flag) to 1. The SNDAINTF.SBSY bit is cleared to 0.

Figure 17.4.2.1 shows a buzzer output timing chart in normal buzzer mode.

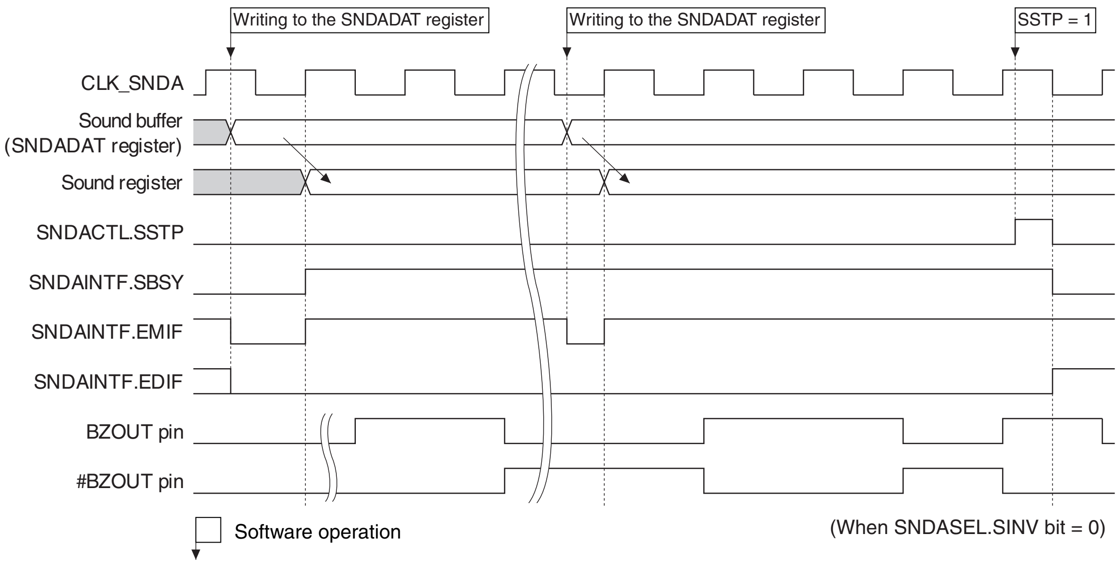


Figure 17.4.2.1 Buzzer Output Timing Chart in Normal Buzzer Mode

Buzzer output waveform configuration (normal buzzer mode/one-shot buzzer mode)

Set the buzzer signal frequency and duty ratio (high period/cycle) using the SNDADAT.SFRQ[7:0] and SNDADAT.SLEN[5:0] bits, respectively. Use the following equations to calculate these setting values.

$$\text{SNDADAT.SFRQ[7:0] bits} = \frac{f_{CLK\_SNDA}}{f_{BZOUT}} - 1 \qquad \text{(Eq. 17.1)}$$

$$\text{SNDADAT.SLEN[5:0] bits} = \left(\frac{f_{CLK\_SNDA}}{f_{BZOUT}} \times \frac{DUTY}{100}\right) - 1 \qquad \text{(Eq. 17.2)}$$

Where

f_{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 SNDADAT.SFRQ[7:0] bits ≤ SNDADAT.SLEN[5:0] bits
- Settings as SNDADAT.SFRQ[7:0] bits = 0x00

Table 17.4.2.1 Buzzer Frequency Settings (when f_{CLK_SNDA} = 32,768 Hz)

SNDADAT. SFRQ[7:0] bits	Frequency [Hz]	SNDADAT. SFRQ[7:0] bits	Frequency [Hz]	SNDADAT. SFRQ[7:0] bits	Frequency [Hz]	SNDADAT. 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 17.4.2.2 Buzzer Duty Ratio Setting Examples (when f_{CLK_SNDA} = 32,768 Hz)

SNDADAT. 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

SNDADAT. 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

17.4.3 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

1. Set the following SNDASEL register bits:
   - Set the SNDASEL.MOSEL[1:0] bits to 0x1. (Set one-shot buzzer mode)
   - SNDASEL.STIM[3:0] bits (Set output duration)
2. Write data to the following sound buffer (SNDADAT register) bits. (Start buzzer output)
   - SNDADAT.SLEN[5:0] bits (Set buzzer output signal duty ratio)
   - SNDADAT.SFRQ[7:0] bits (Set buzzer output signal frequency)

One-shot buzzer output operations

When data is written to the sound buffer (SNDADAT register), SNDA clears the SNDAINTF.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 SNDAINTF.EMIF bit and SNDAINTF.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 SNDASEL.STIM[3:0] bits has elapsed. At the same time, the SNDAINTF.EDIF bit (sound output completion interrupt flag) is set to 1 and the SNDAINTF.SBSY bit is cleared to 0.
Figure 17.4.3.1 shows a buzzer output timing chart in one-shot buzzer mode.

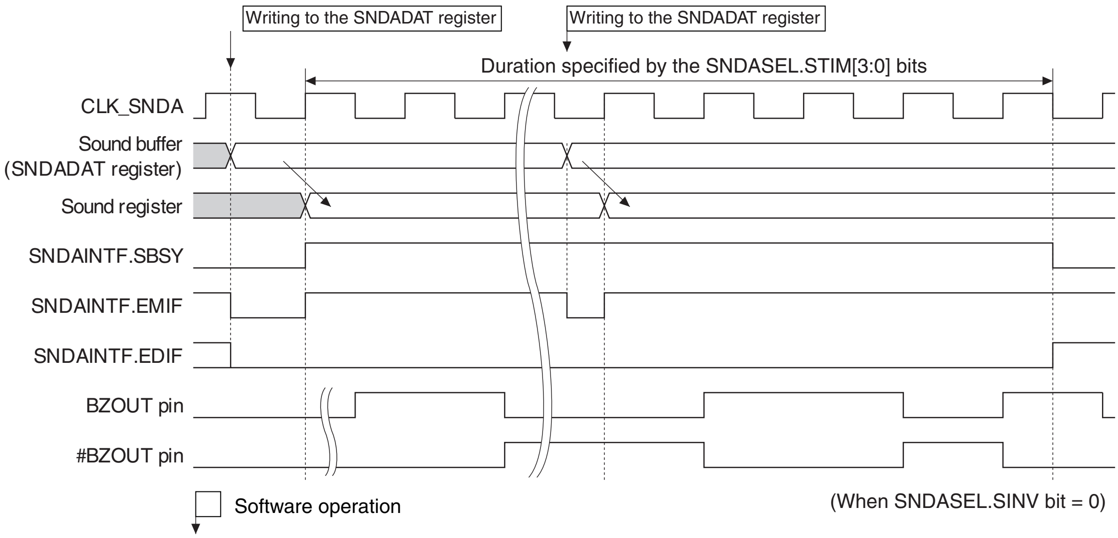


Figure 17.4.3.1 Buzzer Output Timing Chart in One-shot Buzzer Mode

17.4.4 Output in Melody Mode

Melody mode generates the buzzer signal with a melody according to the data written to the sound buffer (SNDADAT 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 SNDASEL register bits:
   - Set the SNDASEL.MOSEL[1:0] bits to 0x2. (Set melody mode)
   - SNDASEL.STIM[3:0] bits (Set tempo)
2. Write data to the following sound buffer (SNDADAT register) bits. (Start sound output)
   - SNDADAT.MDTI bit (Set tie/slur)
   - SNDADAT.MDRS bit (Set note/rest)
   - SNDADAT.SLEN[5:0] bits (Set duration)
   - SNDADAT.SFRQ[7:0] bits (Set scale)
3. Check to see if the SNDAINTF.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 (SNDADAT register), SNDA clears the SNDAINTF.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 SNDAINTF.EMIF bit and SNDAINTF.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 (SNDADAT register) until the next trigger is issued. At the same time, the SNDAINTF.EDIF bit (sound output completion interrupt flag) is set to 1 and the SNDAINTF.SBSY bit is cleared to 0.

Figure 17.4.4.1 shows a melody mode operation timing chart.

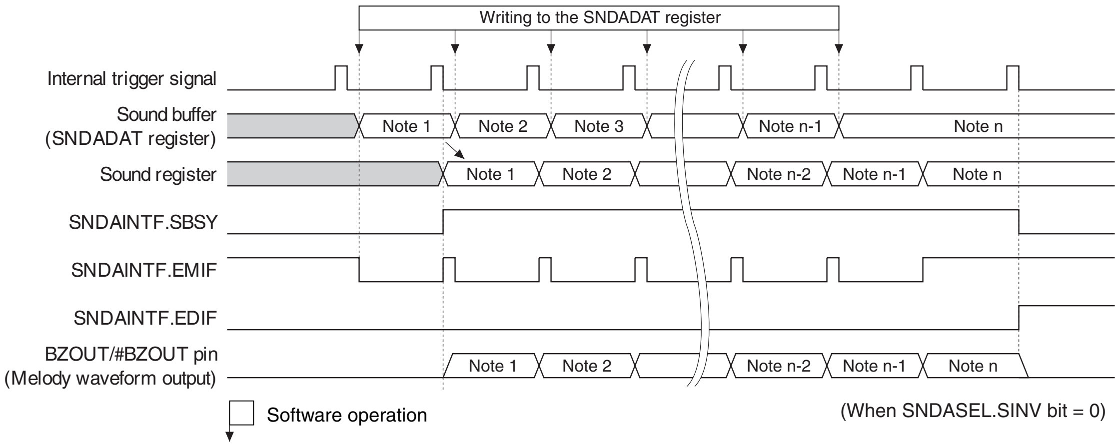


Figure 17.4.4.1 Melody Mode Operation Timing Chart

Melody output using DMA

By setting the SNDAEMDMAEN.EMDMAEN*x* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and melody data is transferred from the specified memory to the sound buffer (SNDADAT register) via DMA Ch.*x* when the SNDAINTF.EMIF bit is set to 1 (sound buffer empty).
This automates the melody output procedure from Steps 2 to 4 described above.
The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance so that transmit data will be transferred to the sound buffer (SNDADAT register). For more information on DMA, refer to the "DMA Controller" chapter.

Table 17.4.4.1 DMA Data Structure Configuration Example (for Melody Output)

Item		Setting example
End pointer	Transfer source	Memory address in which the last melody data is stored
	Transfer destination	SNDADAT register address
Control data	dst_inc	0x3 (no increment)
	dst_size	0x1 (halfword)
	src_inc	0x1 (+2)
	src_size	0x1 (halfword)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

Melody output waveform configuration

Note/rest (duration) specification

Notes and rests can be specified using the SNDADAT.MDRS and SNDADAT.SLEN[5:0] bits.

Table 17.4.4.2 Note/Rest Specification (when f_{CLK_SNDA} = 32,768 Hz)

SNDADAT.SLEN[5:0] bits	SNDADAT.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 SNDADAT.MDTI bit to 1 and the previous note and the current note are played continuously.

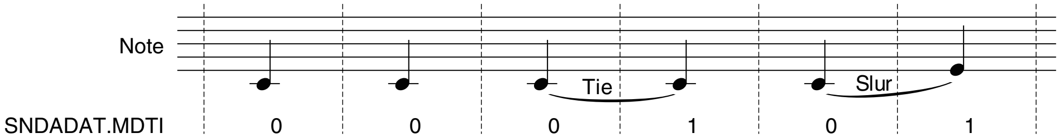


Figure 17.4.4.2 Tie and Slur

Scale specification

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

Table 17.4.4.3 Scale Specification (when f_{CLK_SNDA} = 32,768 Hz)

SNDADAT.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
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

17.5 Interrupts

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

Table 17.5.1 SNDA Interrupt Function

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

SNDA provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the CPU 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" chapter.

17.6 DMA Transfer Requests

The SNDA has a function to generate DMA transfer requests from the causes shown in Table 17.6.1.

Table 17.6.1 DMA Transfer Request Causes of SNDA

Cause to request DMA transfer	DMA transfer request flag	Set condition	Clear condition
Sound buffer empty	Sound buffer empty flag (SNDAINTF.EMIF)	When data in the sound buffer (SNDADAT register) is transferred to the sound register or 1 is written to the SNDACTL.SSTP bit	Writing to the SNDADAT register

The SNDA provides DMA transfer request enable bits corresponding to each DMA transfer request flag shown above for the number of DMA channels. A DMA transfer request is sent to the pertinent channel of the DMA controller only when the DMA transfer request flag, of which DMA transfer has been enabled by the DMA transfer request enable bit, is set. The DMA transfer request flag also serves as an interrupt flag, therefore, both the DMA transfer request and the interrupt cannot be enabled at the same time. After a DMA transfer has completed, disable the DMA transfer to prevent unintended DMA transfer requests from being issued. For more information on the DMA control, refer to the "DMA Controller" chapter.

17.7 Control Registers

SNDA Clock Control Register

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

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 17.7.1 Clock Source and Division Ratio Settings

SNDACLK. CLKDIV[2:0] bits	SNDACLK.CLKSRC[1:0] bits			
	0x0 IOSC	0x1 OSC1	0x2 OSC3	0x3 EXOSC
0x7	Reserved	1/1	Reserved	1/1
0x6	1/1,024		1/1,024	
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 SNDACLK register settings can be altered only when the SNDACTL.MODEN bit = 0.

SNDA Select Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDASEL	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 17.7.2 Tempo/One-shot Buzzer Output Duration Selections (when f_{CLK_SNDA} = 32,768 Hz)

SNDASEL. 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 SNDAINTF.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 17.7.3 Sound Output Mode Selection

SNDASEL.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
SNDACTL	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 SNDACTL.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
SNDADAT	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.

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 (SNDADAT.SFRQ[5:0] bits) are effective within the SNDADAT.SFRQ[7:0] bits. Always set the SNDADAT.SFRQ[7:6] bits to 0x0.
- The SNDADAT 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
SNDAINTF	15–9	–	0x00	–	R	–
	8	SBSY	0	H0	R	
	7–2	–	0x00	–	R	
	1	EMIF	1	H0	R	Cleared by writing to the SNDADAT register.
	0	EDIF	0	H0	R/W	Cleared by writing 1 or writing to the SNDADAT register.

Bits 15–9 Reserved

Bit 8 SBSY

This bit indicates the sound output status. (See Figures 17.4.2.1, 17.4.3.1, and 17.4.4.1.)

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:

SNDAINTF.EMIF bit: Sound buffer empty interrupt

SNDAINTF.EDIF bit: Sound output completion interrupt

SNDA Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
SNDAINTE	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:
SNDAINTE.EMIE bit: Sound buffer empty interrupt
SNDAINTE.EDIE bit: Sound output completion interrupt

SNDA Sound Buffer Empty DMA Request Enable Register

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

Bits 15–0 EMDMAEN[15:0]

These bits enable the SNDA to issue a DMA transfer request to the corresponding DMA controller channel (Ch.0–Ch.15) when a sound buffer empty state has occurred.

1 (R/W): Enable DMA transfer request
0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

18 IR Remote Controller (REMC3)

18.1 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 18.1.1 shows the REMC3 configuration.

Table 18.1.1 REMC3 Channel Configuration of S1C31W65

Item	S1C31W65
Number of channels	1 transmitter channel

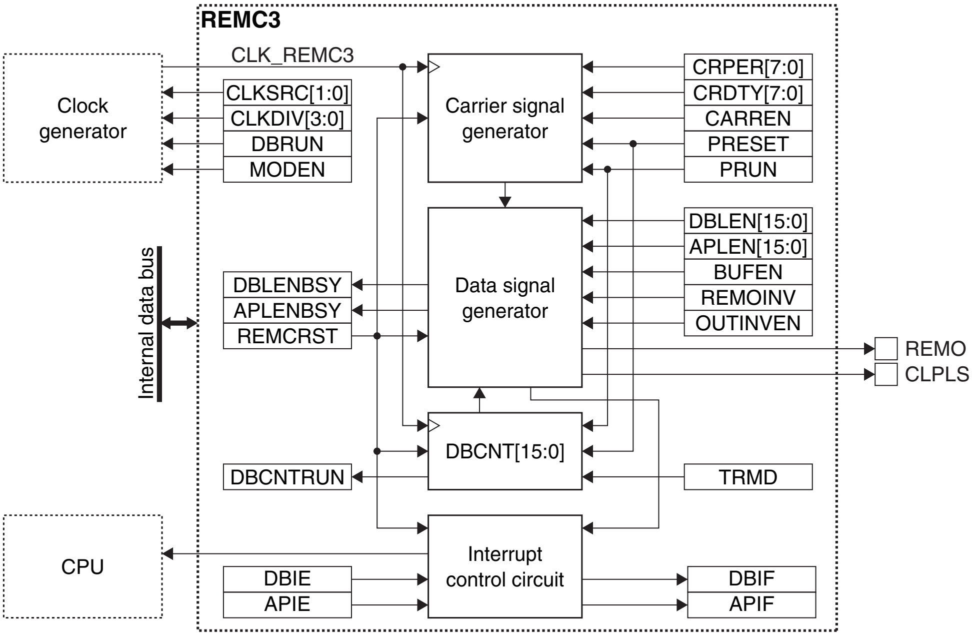


Figure 18.1.1 REMC3 Configuration

18.2 Output Pins and External Connections

18.2.1 List of Output Pins

Table 18.2.1.1 shows the REMC3 pin.

Table 18.2.1.1 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.

18.2.2 External Connections

Figure 18.2.2.1 shows a connection example between the REMC3 and an external infrared module.

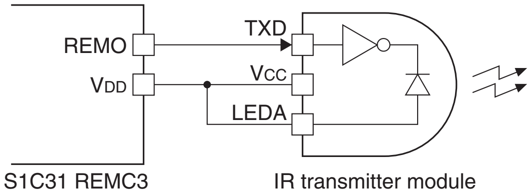


Figure 18.2.2.1 Connection Example Between REMC3 and External Infrared Module

18.3 Clock Settings

18.3.1 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 REMC3CLK register bits:
   - REMC3CLK.CLKSRC[1:0] bits (Clock source selection)
   - REMC3CLK.CLKDIV[3:0] bits (Clock division ratio selection = Clock frequency setting)

18.3.2 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.*xxxx*SLPC bit for the CLK_REMC3 clock source.
If the CLGOSC.*xxxx*SLPC 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.

18.3.3 Clock Supply During Debugging

The CLK_REMC3 supply during debugging should be controlled using the REMC3CLK.DBRUN bit.
The CLK_REMC3 supply to the REMC3 is suspended when the CPU enters debug state if the REMC3CLK.DBRUN bit = 0. After the CPU returns to normal operation, 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 state was entered. If the REMC3CLK.DBRUN bit = 1, the CLK_REMC3 supply is not suspended and the REMC3 will keep operating in debug state.

18.4 Operations

18.4.1 Initialization

The REMC3 should be initialized with the procedure shown below.

1. Write 1 to the REMC3DBCTL.REMCRST bit. (Reset REMC3)
2. Configure the REMC3CLK.CLKSRC[1:0] and REMC3CLK.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 REMC3DBCTL register bits:
   - Set the REMC3DBCTL.MODEN bit to 1. (Enable count operation clock)
   - REMC3DBCTL.TRMD bit (Select repeat mode/one-shot mode)
   - Set the REMC3DBCTL.BUFEN bit to 1. (Enable compare buffer)
   - REMC3DBCTL.REMOINV bit (Configure inverse logic output signal)
5. Configure the following REMC3CARR register bits:
   - REMC3CARR.CRPER[7:0] bit (Set carrier signal cycle)
   - REMC3CARR.CRDTY[7:0] bit (Set carrier signal duty)
6. Configure the following REMC3CCTL register bits:
   - REMC3CCTL.CARREN bit (Enable/disable carrier modulation)
   - REMC3CCTL.OUTINVEN bit (Configure output signal polarity)
7. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the REMC3INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the REMC3INTE register to 1. (Enable interrupts)

18.4.2 Data Transmission Procedures

Starting data transmission

The following shows a procedure to start data transmission.

1. Set the REMC3APLEN.APLEN[15:0] bits. (Set data signal duty)
2. Set the REMC3DBLEN.DBLEN[15:0] bits. (Set data signal cycle)
3. Set the following REMC3DBCTL register bits:
   - Set the REMC3DBCTL.PRESET bit to 1. (Reset internal counters)
   - Set the REMC3DBCTL.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 REMC3APLEN.APLEN[15:0] and REMC3DBLEN.DBLEN[15:0] bits, respectively, before a compare DB interrupt (REMC3INTF.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 (REMC3INTF.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 (REMC3INTF.DBIF bit = 1).
2. Set the REMC3DBCTL.PRUN bit to 0. (Stop counting)
3. Set the REMC3DBCTL.MODEN bit to 0. (Disable count operation clock)

18.4.3 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 18.4.3.1 shows an example of the output waveform.

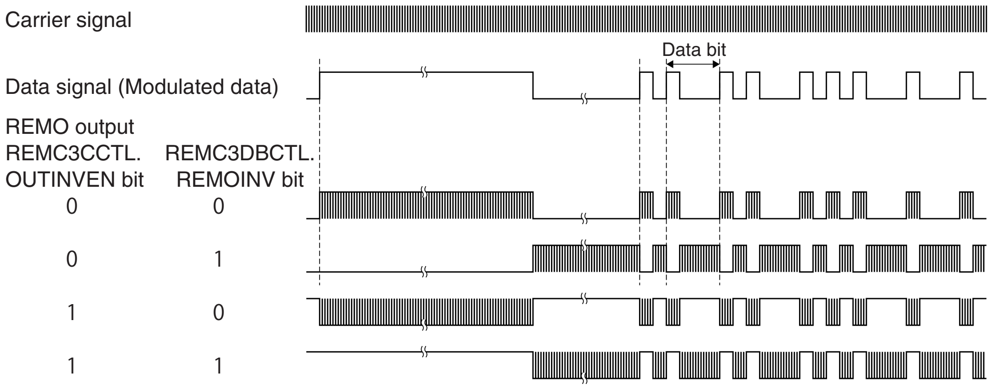


Figure 18.4.3.1 REMO Output Waveform Example

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 REMC3CARR.CRDTY[7:0] and REMC3CARR.CRPER[7:0] bits. Figure 18.4.3.2 shows an example of the carrier signal generated.

Example) REMC3CARR.CRDTY[7:0] bits = 2, REMC3CARR.CRPER[7:0] bits = 8

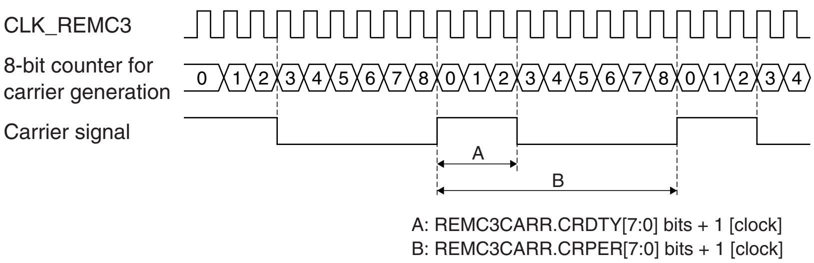


Figure 18.4.3.2 Example of Carrier Signal Generated

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

$$\text{Carrier frequency} = \frac{f_{CLK\_REMC3}}{CRPER + 1} \qquad \text{Duty ratio} = \frac{CRDTY + 1}{CRPER + 1} \qquad \text{(Eq. 18.1)}$$

Where

f_{CLK_REMC3}: CLK_REMC3 frequency [Hz]
CRPER: REMC3CARR.CRPER[7:0] bit-setting value (1–255)
CRDTY: REMC3CARR.CRDTY[7:0] bit-setting value (0–254)
* REMC3CARR.CRDTY[7:0] bits < REMC3CARR.CRPER[7:0] bits

The 8-bit counter for carrier generation is reset by the REMC3DBCTL.PRESET bit and is started/stopped by the REMC3DBCTL.PRUN bit in conjunction with the 16-bit counter for data signal generation. When the counter value is matched with the REMC3CARR.CRDTY[7:0] bits, the carrier signal waveform is inverted. When the counter value is matched with the REMC3CARR.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 (REMC3DBCNT.DBCNT[15:0] bits) that runs with CLK_REMC3 and the setting values of the REMC3APLEN.APLEN[15:0] and REMC3DBLEN.DBLEN[15:0] bits. Figure 18.4.3.3 shows an example of the data signal generated.

Example) REMC3APLEN.APLEN[15:0] bits = 0x0bd0, REMC3DBLEN.DBLEN[15:0] bits = 0x11b8, REMC3DBCTL.TRMD bit = 0 (repeat mode), REMC3DBCTL.REMOINV bit = 0 (signal logic non-inverted)

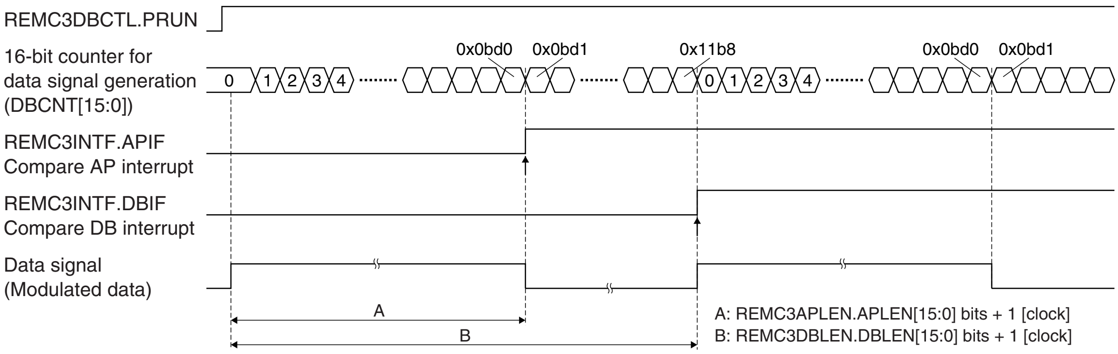


Figure 18.4.3.3 Example of Data Signal Generated

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

$$\text{Data length} = \frac{\text{DBLEN} + 1}{f_{\text{CLK\_REMC3}}} \qquad \text{Duty ratio} = \frac{\text{APLEN} + 1}{\text{DBLEN} + 1} \qquad \text{(Eq. 18.2)}$$

Where

$f_{\text{CLK\_REMC3}}$: CLK_REMC3 frequency [Hz]

DBLEN: REMC3DBLEN.DBLEN[15:0] bit-setting value (1–65,535)

APLEN: REMC3APLEN.APLEN[15:0] bit-setting value (0–65,534)

* REMC3APLEN.APLEN[15:0] bits < REMC3DBLEN.DBLEN[15:0] bits

The 16-bit counter for data signal generation is reset by the REMC3DBCTL.PRESET bit and is started/stopped by the REMC3DBCTL.PRUN bit. When the counter value is matched with the REMC3APLEN.APLEN[15:0] bits (compare AP), the data signal waveform is inverted. When the counter value is matched with the REMC3DBLEN.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 REMC3DBLEN.DBLEN[15:0] and REMC3APLEN.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 (REMC3DBCTL.TRMD bit = 0), the counter keeps operating until it is stopped using the REMC3DBCTL.PRUN bit. When the counter is set to one-shot mode (REMC3DBCTL.TRMD bit = 1), the counter stops automatically when the counter value is matched with the REMC3DBLEN.DBLEN[15:0] bit-setting value.

18.4.4 Continuous Data Transmission and Compare Buffers

Figure 18.4.4.1 shows an operation example of continuous data transmission with the compare buffer enabled.

Example) REMC3DBCTL.TRMD bit = 0 (repeat mode), REMC3DBCTL.BUFEN bit = 1 (compare buffer enabled),
REMC3DBCTL.REMOINV bit = 0 (signal logic non-inverted)

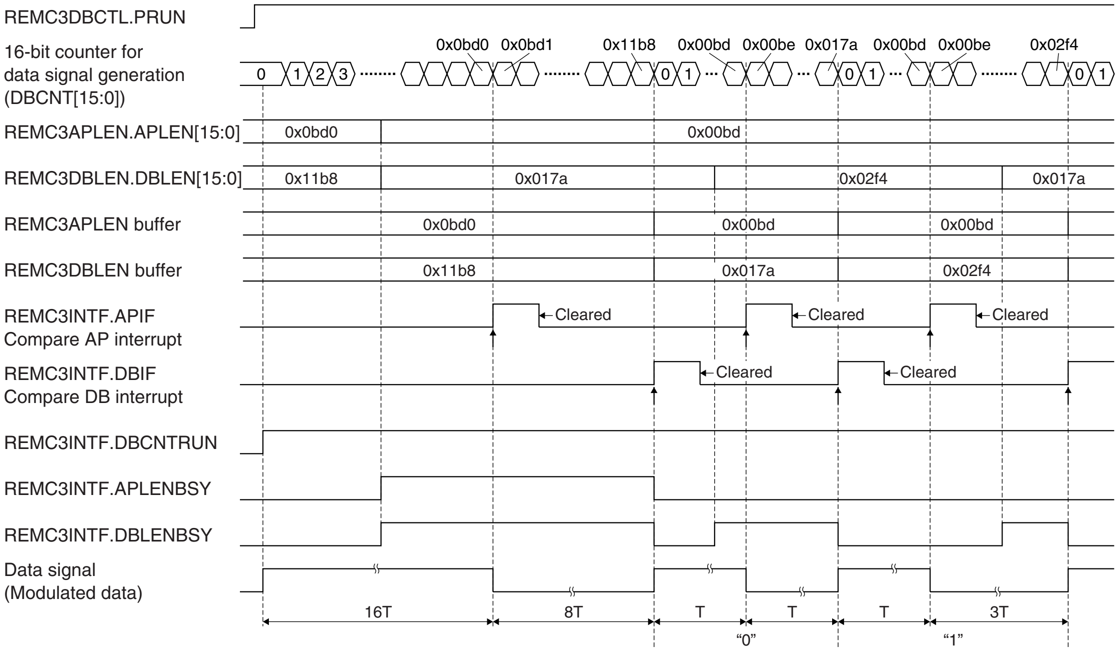


Figure 18.4.4.1 Continuous Data Transmission Example

When the compare buffer is disabled (REMC3DBCTL.BUFEN bit = 0), the 16-bit counter value is directly compared with the REMC3APLEN.APLEN[15:0] and REMC3DBLEN.DBLEN[15:0] bit values. The comparison value is altered immediately after the REMC3APLEN.APLEN[15:0] or REMC3DBLEN.DBLEN[15:0] bits are rewritten.

When the compare buffer is enabled (REMC3DBCTL.BUFEN bit = 1), the REMC3APLEN.APLEN[15:0] and REMC3DBLEN.DBLEN[15:0] bit values are loaded into the compare buffers provided respectively (REMC3APLEN buffer and REMC3DBLEN 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 REMC3DBLEN 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 (REMC3INTF.APLENBSY bit and REMC3INTF.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.

18.5 Interrupts

The REMC3 has a function to generate the interrupts shown in Table 18.5.1.

Table 18.5.1 REMC3 Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Compare AP	REMC3INTF.APIF	When the REMC3APLEN register (or REMC3APLEN buffer) value and the 16-bit counter for data signal generation are matched	Writing 1 to the interrupt flag or the REMC3DBCTL.REMCRST bit
Compare DB	REMC3INTF.DBIF	When the REMC3DBLEN register (or REMC3DBLEN buffer) value and the 16-bit counter for data signal generation are matched	Writing 1 to the interrupt flag or the REMC3DBCTL.REMCRST bit

The REMC3 provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the CPU 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" chapter.

18.6 Application Example: Driving EL Lamp

The REMC3 can be used to simply drive an EL lamp as an application example. Figures 18.6.1 and 18.6.2 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.

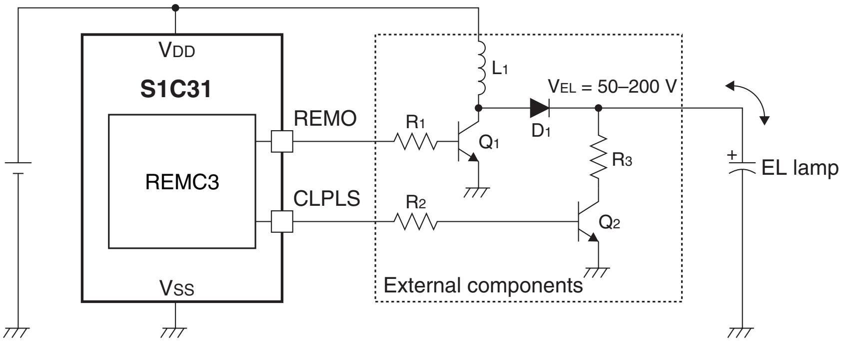


Figure 18.6.1 Example of EL Lamp Drive Circuit

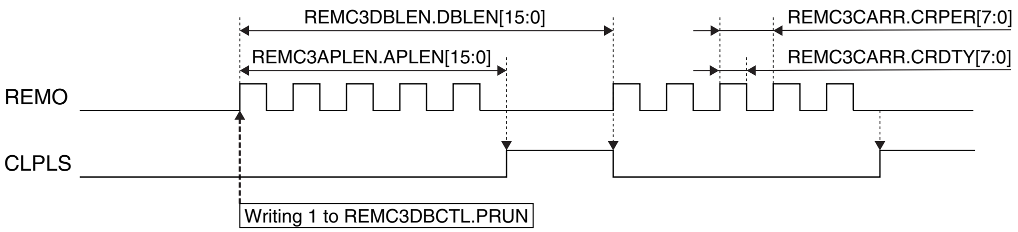


Figure 18.6.2 Example of Generated Drive Waveform

The REMO and CLPLS signals are output from the respective pins while the REMC3DBCTL.PRUN bit = 1. The difference between the setting values of the REMC3DBLEN.DBLEN[15:0] bits and REMC3APLEN.APLEN[15:0] bits becomes the CLPLS pulse width (high period).

18.7 Control Registers

REMC3 Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMC3CLK	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 during debugging or not.

1 (R/W): Clock supplied during debugging

0 (R/W): No clock supplied during debugging

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 18.7.1 Clock Source and Division Ratio Settings

REMC3CLK. CLKDIV[3:0] bits	REMC3CLK.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 REMC3CLK register settings can be altered only when the REMC3DBCTL.MODEN bit = 0.

REMC3 Data Bit Counter Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMC3DBCTL	15–10	–	0x00	–	R	–
	9	PRESET	0	H0/S0	R/W	Cleared by writing 1 to the REMC3DBCTL.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 REMC3DBCTL.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 REMC3DBCTL.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 REMC3DBCTL.MODEN bit must be set to 1. While the counter is running, writing 0 to the REMC3DBCTL.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 18.4.3.1.

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 REMC3DBCTL.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.

Note: After the data signal is output in one-shot mode, set the REMC3DBCTL.REMCRST bit to 1.

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 REMC3DBCTL.MODEN bit is altered from 1 to 0 while sending data, the data being sent cannot be guaranteed. When setting the REMC3DBCTL.MODEN bit to 1 again after that, be sure to write 1 to the REMC3DBCTL.REMCRST bit as well.

REMC3 Data Bit Counter Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMC3DBCNT	15–0	DBCNT[15:0]	0x0000	H0/S0	R	Cleared by writing 1 to the REMC3DBCTL.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
REMC3APLEN	15–0	APLEN[15:0]	0x0000	H0	R/W	Writing enabled when REMC3DBCTL. MODEN bit = 1.

Bits 15–0 APLEN[15:0]

These bits set the active pulse length of the data signal (high period when the REMC3DBCTL.REMOINV bit = 0 or low period when the REMC3DBCTL.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 REMC3APLEN. APLEN[15:0] bit-setting value. The data signal duty ratio is determined by this setting and the REMC3DBLEN.DBLEN[15:0] bit-setting. (See Figure 18.4.3.3.)

Before this register can be rewritten, the REMC3DBCTL.MODEN bit must be set to 1.

REMC3 Data Bit Length Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMC3DBLEN	15–0	DBLEN[15:0]	0x0000	H0	R/W	Writing enabled when REMC3DBCTL. 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 REMC3DBLEN.DBLEN[15:0] bit-setting value. (See Figure 18.4.3.3.)

Before this register can be rewritten, the REMC3DBCTL.MODEN bit must be set to 1.

REMC3 Status and Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMC3INTF	15–11	–	0x00	–	R	–
	10	DBCNTRUN	0	H0/S0	R	Cleared by writing 1 to the REMC3DBCTL.REMCRST bit.
	9	DBLENBSY	0	H0	R	Effective when the REMC3DBCTL. BUFEN bit = 1.
	8	APLENBSY	0	H0	R	
	7–2	–	0x00	–	R	–
	1	DBIF	0	H0/S0	R/W	Cleared by writing 1 to this bit or the REMC3DBCTL.REMCRST bit.
	0	APIF	0	H0/S0	R/W	

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 18.4.4.1.)

1 (R): Running (Counting)

0 (R): Idle

Bit 9 DBLENBSY

This bit indicates whether the value written to the REMC3DBLEN.DBLEN[15:0] bits is transferred to the REMC3DBLEN buffer or not. (See Figure 18.4.4.1.)

1 (R): Transfer to the REMC3DBLEN buffer has not completed.

0 (R): Transfer to the REMC3DBLEN buffer has completed.

While this bit is set to 1, writing to the REMC3DBLEN.DBLEN[15:0] bits is ineffective.

Bit 8 APLENBSY

This bit indicates whether the value written to the REMC3APLEN.APLEN[15:0] bits is transferred to the REMC3APLEN buffer or not. (See Figure 18.4.4.1.)

1 (R): Transfer to the REMC3APLEN buffer has not completed.

0 (R): Transfer to the REMC3APLEN buffer has completed.

While this bit is set to 1, writing to the REMC3APLEN.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:
REMC3INTF.DBIF bit: Compare DB interrupt
REMC3INTF.APIF bit: Compare AP interrupt

These interrupt flags are also cleared to 0 when 1 is written to the REMC3DBCTL.REMCRST bit.

REMC3 Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMC3INTE	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:
REMC3INTE.DBIE bit: Compare DB interrupt
REMC3INTE.APIE bit: Compare AP interrupt

REMC3 Carrier Waveform Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMC3CARR	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 REMC3CARR.CRDTY[7:0] bit-setting value. The carrier signal duty ratio is determined by this setting and the REMC3CARR.CRPER[7:0] bit-setting. (See Figure 18.4.3.2.)

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 REMC3CARR.CRPER[7:0] bit-setting value. (See Figure 18.4.3.2.)

REMC3 Carrier Modulation Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
REMC3CCTL	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 18.4.3.1.

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 REMC3DBCTL.REMOINV bit should be set to 0.

19 12-bit A/D Converter (ADC12A)

19.1 Overview

The ADC12A is a successive approximation type 12-bit A/D converter.
The features of the ADC12A are listed below.

- Conversion method: Successive approximation type
- Resolution: 12 bits
- Analog input voltage range: Reference voltage VREFA to Vss
- Supports two conversion modes: 1. One-time conversion mode
  2. Continuous conversion mode
- Supports three conversion triggers: 1. Software trigger
  2. 16-bit timer underflow trigger
  3. External trigger
- Can convert multiple analog input signals sequentially.
- Can generate conversion completion and overwrite error interrupts.
- Can issue a DMA transfer request when a conversion has completed.

Figure 19.1.1 shows the ADC12A configuration.

Table 19.1.1 ADC12A Configuration of S1C31W65

Item	S1C31W65
Number of channels	1 channel (Ch.0)
Number of analog signal inputs per channel	Ch.0: 8 inputs (ADIN00–ADIN07 *1)
16-bit timer used as conversion clock and trigger sources	Ch.0 ← 16-bit timer Ch.7
VREFA pin (reference voltage input)	Can be input externally or generated internally *2

*1 ADIN07 is connected to the temperature sensor output.
*2 The reference voltage generator output can be input as the reference voltage.
For more information, refer to the “Temperature Sensor/Reference Voltage Generator” chapter.

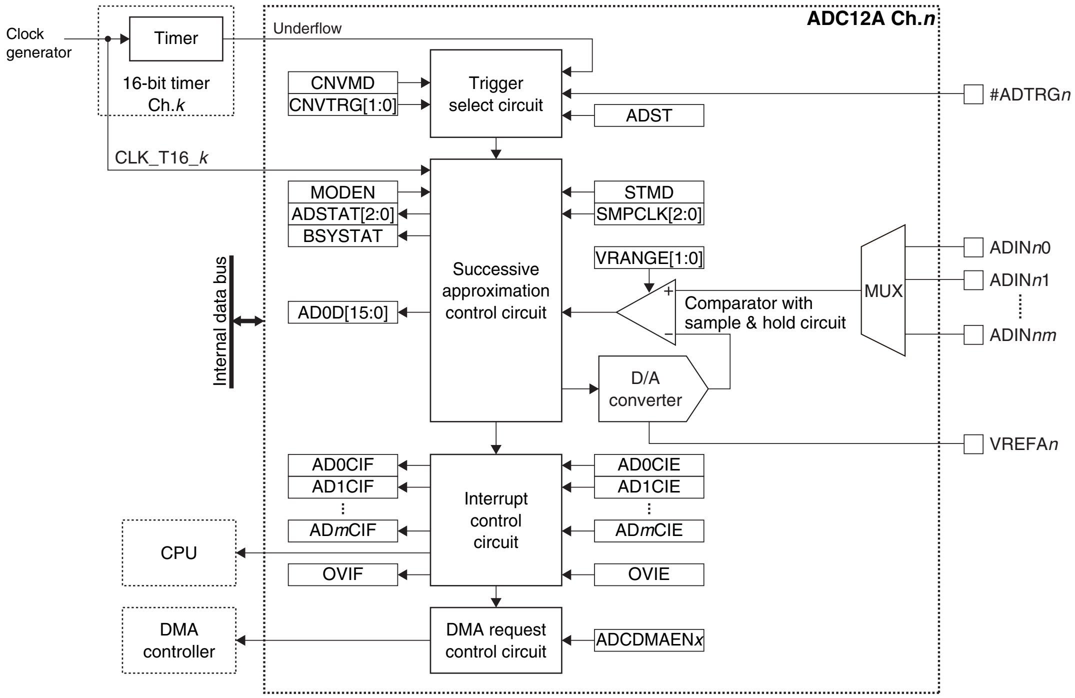


Figure 19.1.1 ADC12A Configuration

Note: In this chapter, *n*, *m*, and *k* refer to an ADC12A channel number, an analog input pin number, and a 16-bit timer channel number, respectively.

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*nm*	A	Hi-Z	Analog signal input
#ADTRG*n*	I	I	External trigger input
VREFA*n*	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.

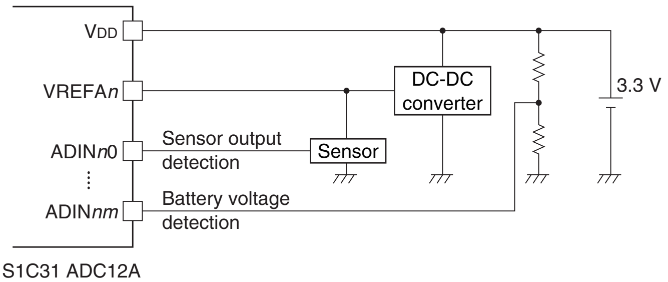


Figure 19.2.2.1 Connections between ADC12A and External Devices

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.

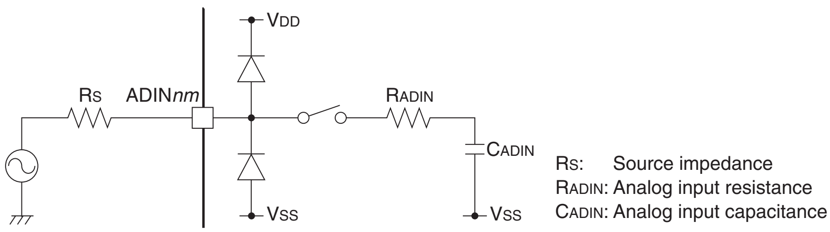


Figure 19.3.2.1 Equivalent Circuit of Analog Input Portion

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 ADC12A_*n*TRG.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 = ADC12A_*n*TRG.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 ADC12A_*n*CTL.MODEN bit to 1. (Enable ADC12A operations)
4. Configure the following ADC12A_*n*TRG register bits:
   - ADC12A_*n*TRG.SMPCLK[2:0] bits (Set sampling time)
   - ADC12A_*n*TRG.CNVTRG[1:0] bits (Select conversion start trigger source)
   - ADC12A_*n*TRG.CNVMD bit (Set conversion mode)
   - ADC12A_*n*TRG.STMD bit (Set data storing mode)
   - ADC12A_*n*TRG.STAAIN[2:0] bits (Set analog input pin to be A/D converted first)
   - ADC12A_*n*TRG.ENDAIN[2:0] bits (Set analog input pin to be A/D converted last)
5. Set the ADC12A_*n*CFG.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 ADC12A_*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the ADC12A_*n*INTE register to 1. (Enable interrupts)
7. Configure the DMA controller and set the following ADC12A control bit when using DMA transfer:
   - Write 1 to the DMA transfer request enable bit in the ADC12A_*n*DMAEN register. (Enable DMA transfer requests)

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 ADC12A_*n*TRG.CNVTRG[1:0] bits.

External trigger (#ADTRG*n* pin)

Writing 1 to the ADC12A_*n*CTL.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 ADC12A_*n*CTL.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 ADC12A_*n*CTL.ADST bit starts A/D conversion.

Trigger inputs can be accepted while the ADC12A_*n*CTL.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 ADC12A_*n*CTL.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 ADC12A_*n*TRG.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 ADC12A_*n*TRG.STAAIN[2:0] bits for specifying the first analog input pin and the ADC12A_*n*TRG.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 ADC12A_*n*CTL.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 ADC12A_*n*CTL.ADST bit.
2. Wait for an ADC12A interrupt.
   i. If the ADC12A_*n*INTF.AD*m*CIF bit = 1 (analog input signal *m* A/D conversion completion interrupt), clear the ADC12A_*n*INTF.AD*m*CIF bit and then go to Step 3.
   ii. If the ADC12A_*n*INTF.OVIF bit = 1 (A/D conversion result overwrite error interrupt), clear the ADC12A_*n*INTF.OVIF bit and terminate as an error or retry A/D conversion.
3. Read the A/D conversion result of the analog input *m* (ADC12A_*n*ADD.ADD[15:0] bits).
   * The 12-bit conversion results are located at the low-order 12 bits or high-order 12-bits within the ADC12A_*n*ADD.ADD[15:0] bits according to the ADC12A_*n*TRG.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 ADC12A_*n*CTL.ADST bit.
   The ADC12A stops operating after the A/D conversion currently being executed has completed.

   The ADC12A_*n*CTL.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 ADC12A_*n*CTL.ADST bit.
2. Wait for an ADC12A interrupt.
   i. If the ADC12A_*n*INTF.AD*m*CIF bit = 1 (analog input signal *m* A/D conversion completion interrupt), clear the ADC12A_*n*INTF.AD*m*CIF bit and then go to Step 3.
   ii. If the ADC12A_*n*INTF.OVIF bit = 1 (A/D conversion result overwrite error interrupt), clear the ADC12A_*n*INTF.OVIF bit and terminate as an error or retry A/D conversion.
3. Read the A/D conversion result of the analog input *m* (ADC12A_*n*ADD.ADD[15:0] bits).
4. Repeat Steps 2 and 3 until terminating A/D conversion.
5. Write 0 to the ADC12A_*n*CTL.ADST bit.
   The ADC12A stops operating after the A/D conversion currently being executed has completed.

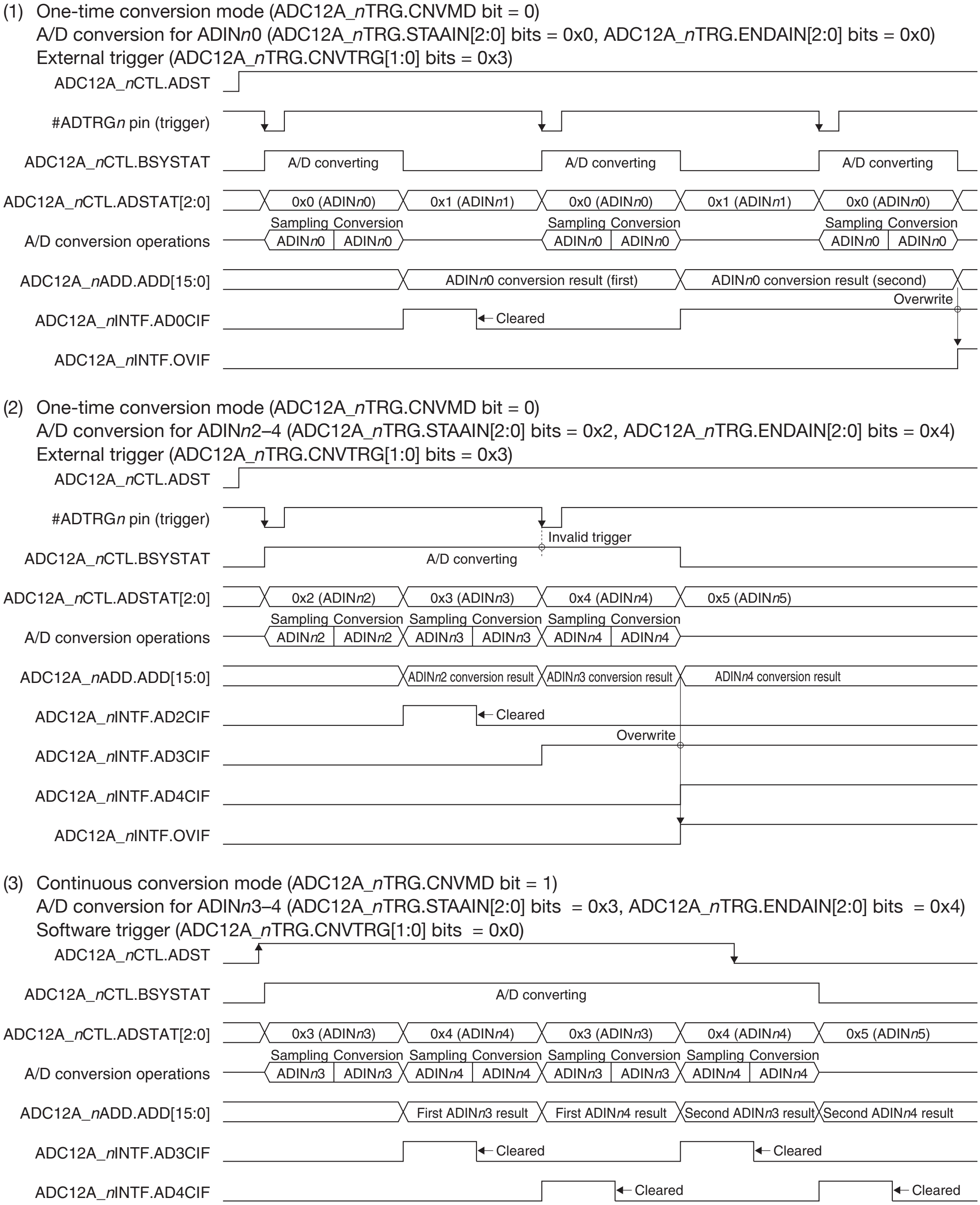


Figure 19.4.4.1 A/D Conversion Operations

A/D converted data transfer using DMA

By setting the ADC12A_*n*DMAEN.ADCDMAEN*x* bit to 1 (DMA transfer request enabled), a DMA transfer request is sent to the DMA controller and the ADC12A_*n*ADD register value is transferred to the specified memory via DMA Ch.*x* when the ADC12A_*n*INTF.AD*m*CIF bit is set to 1 (when A/D conversion for the analog input signal *m* has completed).

This automates reading and saving of A/D converted data.

The transfer source/destination and control data must be set for the DMA controller and the relevant DMA channel must be enabled to start a DMA transfer in advance. For more information on DMA, refer to the "DMA Controller" chapter.

Table 19.4.4.1 DMA Data Structure Configuration Example (Capture Data Transfer)

Item		Setting example
End pointer	Transfer source	ADC12A_*n*ADD register address
	Transfer destination	Memory address to which the last A/D converted data is stored
Control data	dst_inc	0x1 (+2)
	dst_size	0x1 (haflword)
	src_inc	0x3 (no increment)
	src_size	0x1 (halfword)
	R_power	0x0 (arbitrated for every transfer)
	n_minus_1	Number of transfer data
	cycle_ctrl	0x1 (basic transfer)

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 *m* A/D conversion completion	ADC12A_*n*INTF.AD*m*CIF	When an analog input signal *m* A/D conversion result is loaded to the ADC12A_*n*ADD register	Writing 1
A/D conversion result overwrite error	ADC12A_*n*INTF.OVIF	When a new A/D conversion result is loaded to the ADC12A_*n*ADD register while the ADC12A_*n*INTF.AD*m*CIF 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 ADC12A_*n*ADD register has been read or not.

The ADC12A provides interrupt enable bits corresponding to each interrupt flag. An interrupt request is sent to the CPU 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” chapter.

19.6 DMA Transfer Requests

The ADC12A has a function to generate DMA transfer requests from the causes shown in Table 19.6.1.

Table 19.6.1 DMA Transfer Request Causes of ADC12A

Cause to request DMA transfer	DMA transfer request flag	Set condition	Clear condition
Analog input signal *m* A/D conversion completion	A/D conversion completion flag (ADC12A_*n*INTF.AD*m*CIF)	When an analog input signal *m* A/D conversion result is loaded to the ADC12A_*n*ADD register	When the DMA transfer request is accepted

The ADC12A provides DMA transfer request enable bits corresponding to each DMA transfer request flag shown above for the number of DMA channels. A DMA transfer request is sent to the pertinent channel of the DMA controller only when the DMA transfer request flag, of which DMA transfer has been enabled by the DMA transfer request enable bit, is set. The DMA transfer request flag also serves as an interrupt flag, therefore, both the DMA transfer request and the interrupt cannot be enabled at the same time. After a DMA transfer has completed, disable the DMA transfer to prevent unintended DMA transfer requests from being issued. For more information on the DMA control, refer to the “DMA Controller” chapter.

19.7 Control Registers

ADC12A Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12A_*n*CTL	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 19.7.1 Relationship Between Control Bit Value and Analog Input Pin

ADC12A_*n*CTL.ADSTAT[2:0] bits ADC12A_*n*TRG.STAAIN[2:0] bits ADC12A_*n*TRG.ENDAIN[2:0] bits	Analog input pin
0x7	ADIN*n*7
0x6	ADIN*n*6
0x5	ADIN*n*5
0x4	ADIN*n*4
0x3	ADIN*n*3
0x2	ADIN*n*2
0x1	ADIN*n*1
0x0	ADIN*n*0

These bits indicate the last converted analog input pin number after A/D conversion is forcefully terminated by writing 0 to the ADC12A_*n*CTL.ADST bit or automatically terminated in one-time conversion mode (ADC12A_*n*TRG.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 BSYSTAT

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 ADC12A_*n*CTL.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 ADC12A_*n*CTL.MODEN bit, the ADC12A executes a terminate processing. Before the clock source is deactivated, read the ADC12A_*n*CTL.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
ADC12A_*n*TRG	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 ADC12A_*n*CTL.BSYSTAT bit is set to 0 before altering the ADC12A_*n*TRG 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 19.7.1 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 ADC12A_*n*TRG. ENDAIN[2:0] bits ≥ ADC12A_*n*TRG.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 19.7.1 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 register (ADC12A_*n*ADD.ADD[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.

<table>
<tr><td></td><td colspan="16">ADC12A_nADD.ADD[15:0] bits</td></tr>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Left justified (ADC12A_nTRG.STMD bit = 1)</td><td colspan="12">(MSB) 12-bit conversion result (LSB)</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Right justified (ADC12A_nTRG.STMD bit = 0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">(MSB) 12-bit conversion result (LSB)</td></tr>
</table>

Figure 19.7.1 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 19.7.2 Trigger Source Selection

ADC12A_*n*TRG.CNVTRG[1:0] bits	Trigger source
0x3	#ADTRG*n* pin (external trigger)
0x2	Reserved
0x1	16-bit timer Ch.*k* underflow
0x0	ADC12A_*n*CTL.ADST bit (software trigger)

Bit 3 Reserved

Bits 2–0 SMPCLK[2:0]

These bits set the analog input signal sampling time.

Table 19.7.3 Sampling Time Settings

ADC12A_*n*TRG.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
ADC12A_*n*CFG	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1–0	VRANGE[1:0]	0x0	H0	R/W	

Note: Make sure that the ADC12A_*n*CTL.BSYSTAT bit is set to 0 before altering the ADC12A_*n*CFG register.

Bits 15–2 Reserved

Bits 1–0 VRANGE[1:0]

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

Table 19.7.4 A/D Converter Operating Voltage Range Setting

ADC12A_*n*CFG.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_*n*CFG.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_*n*CFG.VRANGE[1:0] bits are set to a value other than 0x0 when the ADC12_*n*CTL.BSYSTAT bit = 1.

ADC12A Ch.*n* Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12A_*n*INTF	15–9	–	0x00	–	R	–
	8	OVIF	0	H0	R/W	Cleared by writing 1.
	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 15–9 Reserved

Bit 8 OVIF

Bits 7–0 AD*m*CIF

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:
ADC12A_*n*INTF.OVIF bit: A/D conversion result overwrite error interrupt
ADC12A_*n*INTF.AD*m*CIF 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
ADC12A_*n*INTE	15–9	–	0x00	–	R	–
	8	OVIE	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 15–9 Reserved

Bit 8 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:
ADC12A_*n*INTE.OVIE bit: A/D conversion result overwrite error interrupt
ADC12A_*n*INTE.AD*m*CIE bit: Analog input signal *m* A/D conversion completion interrupt

ADC12A Ch.*n* DMA Request Enable Register *m*

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12A_*n*DMAEN*m*	15–0	ADCDMAEN[15:0]	0x0000	H0	R/W	–

Bits 15–0 ADCDMAEN[15:0]

These bits enable ADC12A to issue a DMA transfer request to the corresponding DMA controller channel (Ch.0–Ch.15) when the A/D conversion for each analog input has completed.

1 (R/W): Enable DMA transfer request

0 (R/W): Disable DMA transfer request

Each bit corresponds to a DMA controller channel. The high-order bits for the unimplemented channels are ineffective.

ADC12A Ch.*n* Result Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
ADC12A_*n*ADD	15–0	ADD[15:0]	0x0000	H0	R	–

Bits 15–0 ADD[15:0]

The A/D conversion results are set to these bits.

20 Temperature Sensor/Reference Voltage Generator (TSRVR)

20.1 Overview

The TSRVR is a peripheral circuit for the internal A/D converter that outputs the internal temperature sensor detection values and generates the reference voltage. The features of the TSRVR are listed below.

- Includes a temperature sensor that has a linear output characteristic and the sensor output can be measured using the internal A/D converter without external components being attached.
- Can supply a reference voltage (2.0 V, 2.5 V, or VDD selectable) to the internal A/D converter.
- Can supply the reference voltage generated in this circuit to external devices if this IC has the VREFA exclusive pin.

Figure 20.1.1 shows the TSRVR configuration.

Table 20.1.1 TSRVR Configuration of S1C31W65

Item	S1C31W65
Number of channels	1 channel (Ch.0)
Correspondence between TSRVR and internal A/D converter channels	TSRVR Ch.0 → ADC12A Ch.0
A/D converter input connected to temperature sensor	ADIN07
Reference voltage output to external devices	Unavailable

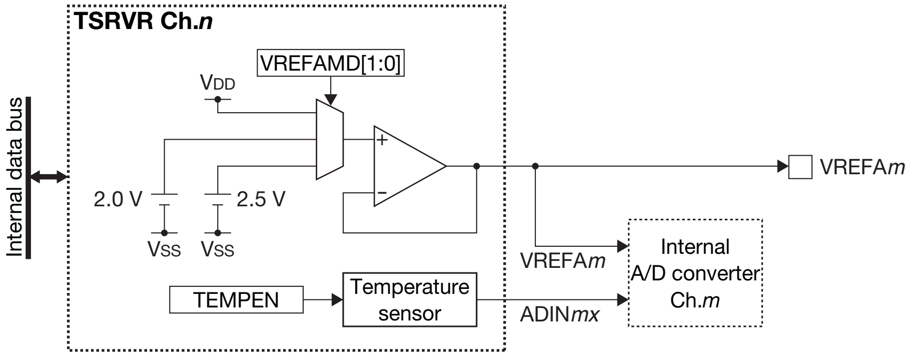


Figure 20.1.1 TSRVR Configuration

Note: In this chapter, *n* and *m* refer to a TSRVR channel number and an internal A/D converter channel number, respectively.

20.2 Output Pin and External Connections

20.2.1 Output Pin

Table 20.2.1.1 shows the TSRVR pin.

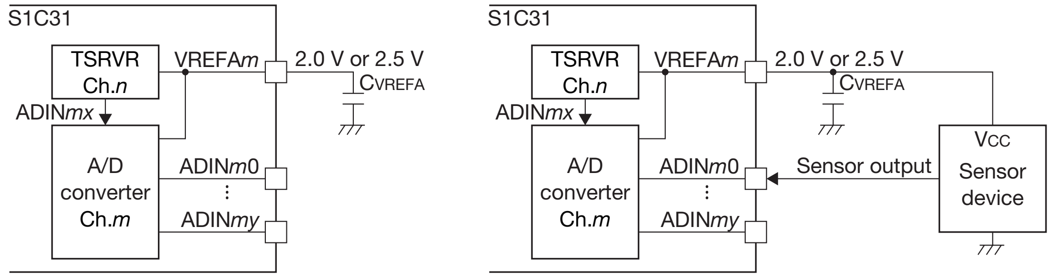
Table 20.2.1.1 TSRVR Pin

Pin name	I/O	Initial status	Function
VREFA*m*	A	Hi-Z	Reference voltage output

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

20.2.2 External Connections

Figure 20.2.2.1 shows connection diagrams between the TSRVR and external components.



(1) When an external device is not connected (2) When an external device is connected

Figure 20.2.2.1 Connections between TSRVR and External Components

20.3 Operations

TSRVR should be configured before starting measurements using the internal A/D converter.

20.3.1 Reference Voltage Setting

The TSRVR output voltage can be supplied to the internal A/D converter as the reference voltage VREFA*m* when it is not supplied externally. The output voltage can be selected using the TSRVR_*n*VCTL.VREFAMD[1:0] bits. Connect C_{VREFA} to the VREFA*m* pin when supplying the reference voltage from TSRVR. A/D conversion by the internal A/D converter should be started after the reference voltage stabilization time t_{VREFA} has elapsed from the time when the output voltage is selected.

20.3.2 Temperature Sensor Setting

The temperature sensor output voltage can be directly measured using the internal A/D converter. The measurement should be started after the temperature sensor output stabilization time t_{TEMP} has elapsed from writing 1 to the TSRVR_*n*TCTL.TEMPEN bit to activate the temperature sensor.

From the temperature sensor output voltage, the measured temperature can be calculated by the equations shown below.

$$T_{SEN} = \frac{(V_{TSEN} - V_{TREF}) \times 1{,}000}{\Delta V_{TEMP}} + T_{REF} \qquad \text{(Eq. 20.1)}$$

Where

T_{SEN}: Actual temperature [°C]

V_{TSEN}: Temperature sensor output voltage at temperature T_{SEN} [V]

T_{REF}: Reference temperature for calibration [°C]

V_{TREF}: Temperature sensor output voltage at temperature T_{REF} [V]

ΔV_{TEMP}: Temperature sensor output voltage temperature coefficient [mV/°C] (Refer to the “Electrical Characteristics” chapter.)

Convert the digital values corresponding to the respective temperatures, that are obtained by the internal A/D converter, into voltage values and assign them to V_{TSEN} and V_{TREF}.

$$V_{(TSEN,\,TREF)} = \frac{ADD}{4{,}096} \times V_{REFA} \qquad \text{(Eq. 20.2)}$$

Where

ADD: A/D conversion result at temperature T_{SEN} or T_{REF} (decimal)

V_{REFA}: A/D converter reference voltage [V]

For details of the internal A/D converter, refer to the “12-bit A/D Converter” chapter.

20.4 Control Registers

TSRVR Ch.*n* Temperature Sensor Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
TSRVR_*n*TCTL	15–8	–	0x00	–	R	–
	7–1	–	0x00	H0	R	
	0	TEMPEN	0	H0	R/W	

Bits 15–1 Reserved

Bit 0 TEMPEN

This bit enables the temperature sensor operation.

1 (R/W): Enable temperature sensor output

0 (R/W): Disable temperature sensor output

TSRVR Ch.*n* Reference Voltage Generator Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
TSRVR_*n*VCTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	H0	R	
	1–0	VREFAMD[1:0]	0x0	H0	R/W	

Bits 15–2 Reserved

Bits 1–0 VREFAMD[1:0]

These bits set the reference voltage generator output voltage.

Table 20.4.1 Output Voltage Settings

TSRVR_*n*VCTL.VREFAMD[1:0] bits	Output voltage
0x3	2.5 V output
0x2	2.0 V output
0x1	V_{DD} level output
0x0	Hi-Z (An external voltage can be applied.)

Notes:
- Be aware that VREFA operating current I_{VREFA} flows when the TSRVR_*n*VCTL.VREFAMD[1:0] bits are set to 0x2 or 0x3.
- When the TSRVR_*n*VCTL.VREFAMD[1:0] bits are not set to 0x0, do not apply an external voltage to the VREFA*m* pin.

21 LCD Driver (LCD8D)

21.1 Overview

LCD8D is an LCD driver to drive an LCD panel. The features of the LCD8D are listed below.

- The frame frequency is configurable into 16 steps.
- Two types of LCD drive waveforms (Waveform A and Waveform B) can be generated.
- Provides all on, all off, and inverse display functions as well as normal display.
- The segment and common pin assignments can be inverted.
- Provides a partial common output drive function.
- Provides an n-segment-line inverse AC drive function.
- The LCD contrast is adjustable. (Note: See the table below.)
- Includes a power supply for 1/2 or 1/3 bias driving (allows external voltages to be applied).
- Provides the frame signal monitoring output pin.
- Can generate interrupts every frame.

Figure 21.1.1 shows the LCD8D configuration.

Table 21.1.1 LCD8D Configuration of S1C31W65

Item	S1C31W65
Number of segments supported	Max. 224 segments (56SEG × 4COM), Max. 416 segments (52SEG × 8COM)
SEG/COM outputs	56SEG × 1–4COM, 52SEG × 5–8COM
LCD drive voltage mode	Internal generation mode and external voltage application mode 1, 2, 3
Drive bias	1/2 or 1/3 bias
LCD contrast	Adjustable into 32 steps (LCD drive voltage internal generation mode only)
Embedded display data RAM	112 bytes

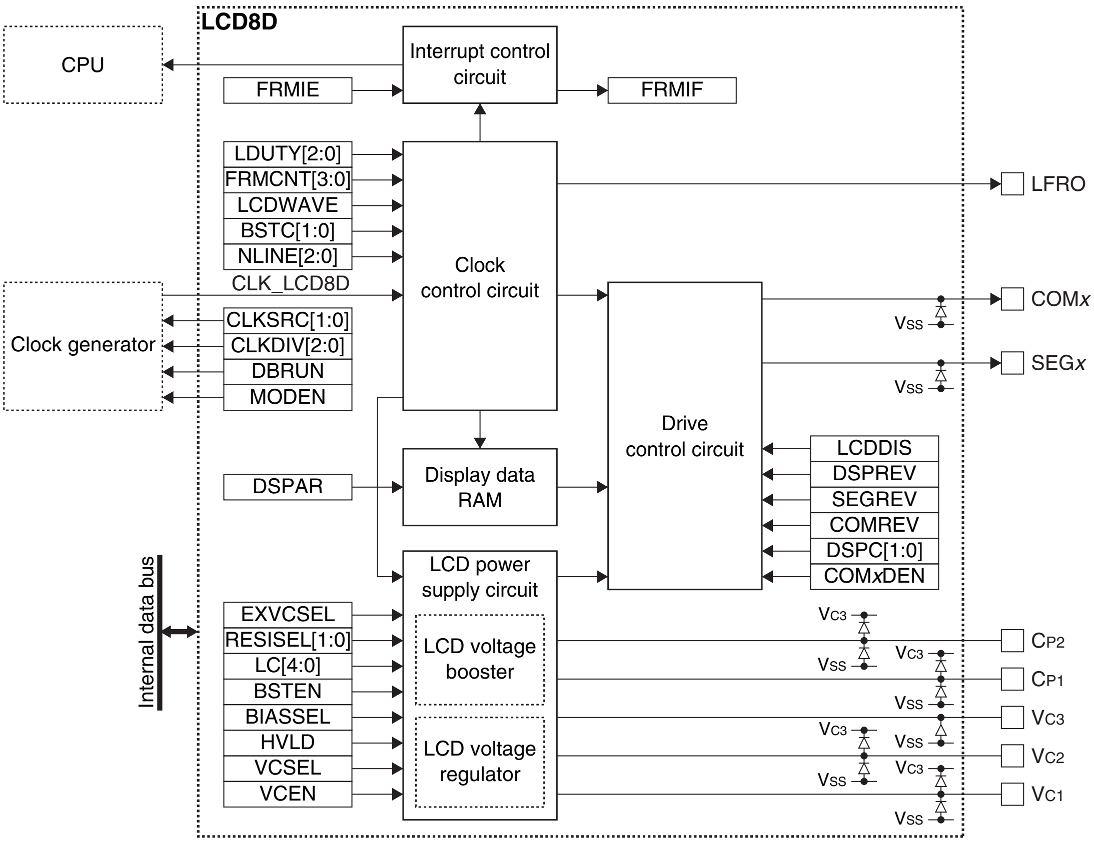


Figure 21.1.1 LCD8D Configuration

21.2 Output Pins and External Connections

21.2.1 List of Output Pins

Table 21.2.1.1 lists the LCD8D pins.

Table 21.2.1.1 List of LCD8D Pins

Pin name	I/O*1	Initial status*1	Function
COM0–3	A	Hi-Z / O (V_{SS})*2	Common data output pins
COM4–7/SEG0–3	A	Hi-Z / O (V_{SS})*2	Common data output/segment data output combination pins
SEG4–55	A	Hi-Z / O (V_{SS})*2	Segment data output pins (See Table 21.2.1.2.)
LFRO	O	O (L)	Frame signal monitoring output pin
V_{C1}	P	–	LCD panel drive power supply pin
V_{C2}	P	–	LCD panel drive power supply pin
V_{C3}	P	–	LCD panel drive power supply pin
C_{P1}	A	–	LCD voltage booster capacitor connecting pin
C_{P2}	A	–	LCD voltage booster capacitor connecting pin

*1: Indicates the status when the pin is configured for LCD8D. *2: When LCD8DCTL.LCDDIS bit = 1

If the port is shared with the LCD8D pin and other functions, the LCD8D output function must be assigned to the port before activating the LCD8D. For more information, refer to the “I/O Ports” chapter.
The COM/SEG combination pin function is switched between the COM pin and the SEG pin according to the COM pin assignment and drive duty selected via software. For the pin configuration, refer to “Drive Duty Switching.”

Notes:
- Be sure to avoid using the V_{C1} to V_{C3} pin outputs for driving external circuits.
- When an LCD panel is connected, set the LCD8DCTL.LCDDIS bit to 1, as activating the LCD panel when it is set to 0 may cause the LCD panel characteristics to fluctuate.

21.2.2 External Connections

Figure 21.2.2.1 shows a connection diagram between LCD8D and an LCD panel.

Note: When the panel is connected, the LCD8DCTL.LCDDIS bit must be set to 1 to bias the panel even if display is turned off.

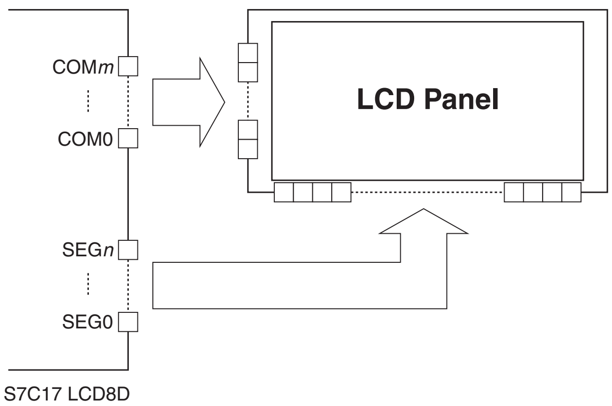


Figure 21.2.2.1 Connections between LCD8D and an LCD Panel

21.3 Clock Settings

21.3.1 LCD8D Operating Clock

When using LCD8D, the LCD8D operating clock CLK_LCD8D must be supplied to LCD8D from the clock generator. The CLK_LCD8D 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 LCD8DCLK register bits:
   - LCD8DCLK.CLKSRC[1:0] bits (Clock source selection)
   - LCD8DCLK.CLKDIV[2:0] bits (Clock division ratio selection = Clock frequency setting)

The CLK_LCD8D frequency should be set to around 32 kHz.

21.3.2 Clock Supply in SLEEP Mode

When using LCD8D during SLEEP mode, the LCD8D operating clock CLK_LCD8D must be configured so that it will keep supplying by writing 0 to the CLGOSC.*xxxx*SLPC bit for the CLK_LCD8D clock source.

21.3.3 Clock Supply in DEBUG Mode

The CLK_LCD8D supply during DEBUG mode should be controlled using the LCD8DCLK.DBRUN bit.
The CLK_LCD8D supply to LCD8D is suspended when the CPU enters DEBUG mode if the LCD8DCLK.DBRUN bit = 0. After the CPU returns to normal mode, the CLK_LCD8D supply resumes. Although LCD8D stops operating and the display is turned off when the CLK_LCD8D supply is suspended, the registers retain the status before DEBUG mode was entered. If the LCD8DCLK.DBRUN bit = 1, the CLK_LCD8D supply is not suspended and LCD8D will keep operating in DEBUG mode.

21.3.4 Frame Frequency

The LCD8D frame signal is generated by dividing CLK_LCD8D. The frame frequency is determined by selecting a division ratio from 16 variations depending on the drive duty using the LCD8DTIM1.FRMCNT[3:0] bits. Use the following equation to calculate the frame frequency.

$$f_{FR} = \frac{f_{CLK\_LCD8D}}{16 \times (FRMCNT + 1) \times (LDUTY + 1)} \qquad \text{(Eq. 21.1)}$$

Where
- f_{FR}: Frame frequency [Hz]
- f_{CLK_LCD8D}: LCD8D operating clock frequency [Hz]
- FRMCNT: LCD8DTIM1.FRMCNT[3:0] setting value (0 to 15)
- LDUTY: LCD8DTIM1.LDUTY[2:0] setting value (0 to 7)

Table 21.3.4.1 lists frame frequency settings when f_{CLK_LCD8D} = 32,768 Hz as an example.

Table 21.3.4.1 Frame Frequency Settings (when f_{CLK_LCD8D} = 32,768 Hz)

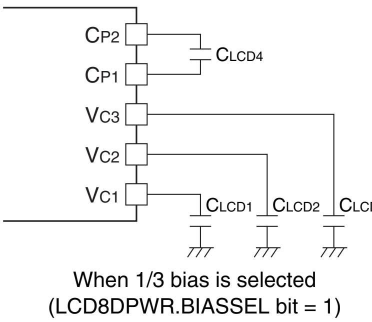
LCD8DTIM1. FRMCNT[3:0] bits	Frame frequency [Hz]							
	1/8 duty	1/7 duty	1/6 duty	1/5 duty	1/4 duty	1/3 duty	1/2 duty	Static
0xf	16.0	18.3	21.3	25.6	32.0	42.7	64.0	128.0
0xe	17.1	19.5	22.8	27.3	34.1	45.5	68.3	136.5
0xd	18.3	20.9	24.4	29.3	36.6	48.8	73.1	146.3
0xc	19.7	22.5	26.3	31.5	39.4	52.5	78.8	157.5
0xb	21.3	24.4	28.4	34.1	42.7	56.9	85.3	170.7
0xa	23.3	26.6	31.0	37.2	46.5	62.1	93.1	186.2
0x9	25.6	29.3	34.1	41.0	51.2	68.3	102.4	204.8
0x8	28.4	32.5	37.9	45.5	56.9	75.9	113.8	227.6
0x7	32.0	36.6	42.7	51.2	64.0	85.3	128.0	256.0
0x6	36.6	41.8	48.8	58.5	73.1	97.5	146.3	292.6
0x5	42.7	48.8	56.9	68.3	85.3	113.8	170.7	341.3
0x4	51.2	58.5	68.3	81.9	102.4	136.5	204.8	409.6
0x3	64.0	73.1	85.3	102.4	128.0	170.7	256.0	512.0
0x2	85.3	97.5	113.8	136.5	170.7	227.6	341.3	682.7
0x1	128.0	146.3	170.7	204.8	256.0	341.3	512.0	1,024.0
0x0	256.0	292.6	341.3	409.6	512.0	682.7	1,024.0	2,048.0

21.4 LCD Power Supply

The LCD drive voltages V_{C1} to V_{C3} can be generated by the internal LCD power supply circuit (LCD voltage regulator and LCD voltage booster). One or all voltages can also be applied from outside the IC.
The drive bias can be selected from 1/3 bias and 1/2 bias using the LCD8DPWR.BIASSEL bit.

21.4.1 Internal Generation Mode

This mode generates all the LCD drive voltages V_{C1} to V_{C3} on the chip. To put LCD8D into internal generation mode, set the LCD8DPWR.EXVCSEL bit to 0 and set both the LCD8DPWR.VCEN and LCD8DPWR.BSTEN bits to 1 to turn both the LCD voltage regulator and LCD voltage booster on. The LCD8DPWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 21.4.1.1 shows an external connection example for internal generation mode.



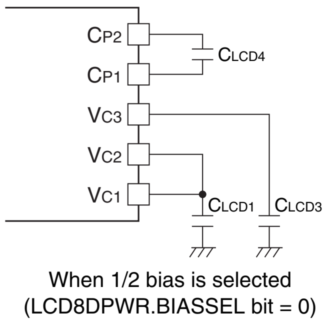


Figure 21.4.1.1 External Connection Example for Internal Generation Mode

21.4.2 External Voltage Application Mode 1

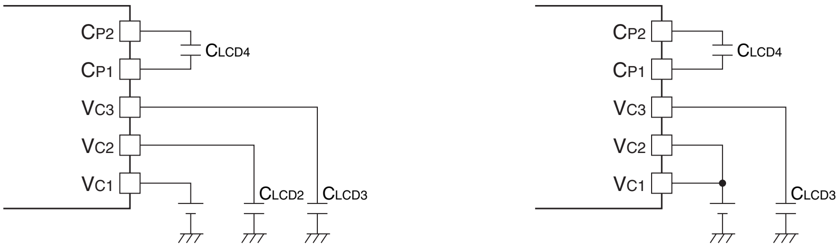
In this mode, all the LCD drive voltages V_{C1} to V_{C3} are applied from outside the IC. To put LCD8D into external voltage application mode 1, set the LCD8DPWR.EXVCSEL bit to 1 and set both the LCD8DPWR.VCEN and LCD8DPWR.BSTEN bits to 0 to turn both the LCD voltage regulator and LCD voltage booster off. The LCD8DPWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 21.4.2.1 shows an external connection example for external voltage application mode 1.

When 1/3 bias is selected (LCD8DPWR.BIASSEL bit = 1) When 1/2 bias is selected (LCD8DPWR.BIASSEL bit = 0)

Figure 21.4.2.1 External Connection Example for External Voltage Application Mode 1 (resistor divider)

21.4.3 External Voltage Application Mode 2

In this mode, the LCD drive voltage V_{C1} or V_{C2} is applied from outside the IC and other voltages are internally generated. To put LCD8D into external voltage application mode 2, set the LCD8DPWR.EXVCSEL bit to 1, set the LCD8DPWR.VCEN bit to 0 to turn the LCD voltage regulator off and the LCD8DPWR.BSTEN bit to 1 to turn the LCD voltage booster on. The LCD8DPWR.RESISEL[1:0] bits should be set to 0x0 to disable the internal LCD voltage dividing resistors. Figure 21.4.3.1 shows an external connection example for external voltage application mode 2.

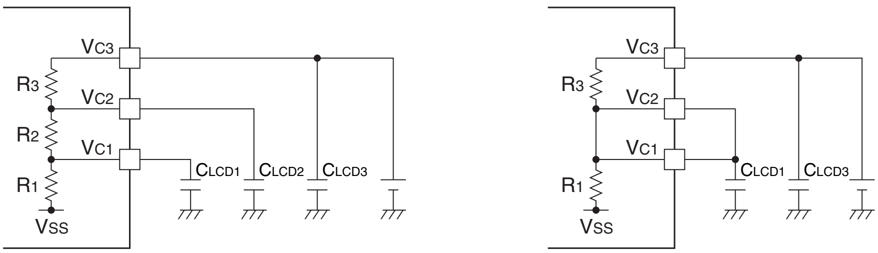


When 1/3 bias is selected (LCD8DPWR.BIASSEL bit = 1) When 1/2 bias is selected (LCD8DPWR.BIASSEL bit = 0)

Figure 21.4.3.1 External Connection Example for External Voltage Application Mode 2 (when V_{C1} is applied)

21.4.4 External Voltage Application Mode 3

In this mode, the LCD drive voltage V_{C3} is applied from outside the IC and the V_{C1} and V_{C2} voltages are generated using the internal LCD voltage dividing resistors. To put LCD8D into external voltage application mode 3, set the LCD8DPWR.EXVCSEL bit to 1 and set both the LCD8DPWR.VCEN and LCD8DPWR.BSTEN bits to 0 to turn both the LCD voltage regulator and LCD voltage booster off. Also set the LCD8DPWR.RESISEL[1:0] bits to 0x1, 0x2, or 0x3 to use the internal LCD voltage dividing resistors according to the LCD panel load. A capacitor should be connected to the V_{C1} to V_{C3} pins while taking fluctuation of LCD load into consideration. Figure 21.4.4.1 shows an external connection example for external voltage application mode 3.



When 1/3 bias is selected (LCD8DPWR.BIASSEL bit = 1) When 1/2 bias is selected (LCD8DPWR.BIASSEL bit = 0)

Figure 21.4.4.1 External Connection Example for External Voltage Application Mode 3

Note: When "Display off" is selected by setting the LCD8DDSP.DSPC[1:0] bits to 0x0 while the external LCD drive voltages are being supplied in an external voltage application mode, the electric charges of V_{C3} can be discharged in the following procedure.

1. Turn the external power supply off.
2. Set the LCD8DPWR.EXVCSEL bit to 0. (Select internal generation mode)
3. Set the LCD8DPWR.EXVCSEL bit to 1. (Select external voltage application mode)

21.4.5 LCD Power Supply Circuit Settings

When using internal generation mode

Select the reference voltage for boosting voltage generated by the LCD voltage regulator according to the power supply voltage V_{DD}. Refer to "LCD Driver (LCD8D) Characteristics" in the "Electrical Characteristics" chapter and set the LCD8DPWR.VCSEL bit. Current consumption can be reduced by selecting reference voltage V_{C2} as compared with reference voltage V_{C1}. By setting the LCD8DPWR.HVLD bit to 1, the LCD voltage regulator enters heavy load protection mode and ensures stable V_{C1} to V_{C3} outputs. Heavy load protection mode should be set when the display has inconsistencies in density. Current consumption increases in heavy load protection mode, therefore do not set heavy load protection mode if unnecessary.

When using internal generation mode or external voltage application mode 2

Set the booster clock frequency used in the LCD voltage booster using the LCD8DTIM2.BSTC[1:0] bits. Set it to the frequency that provides the best V_{C1}–V_{C3} output stability after being evaluated using the actual circuit board.

When using external voltage application mode 3

LCD8D includes voltage dividing resistors to generate the LCD drive voltages V_{C1} and V_{C2} from the V_{C3} that is applied externally. The resistance values can be adjusted according to the external LCD panel load by setting the LCD8DPWR.REGISEL[1:0] bits.

LCD contrast adjustment

The LCD panel contrast can only be adjusted in internal generation mode using the LCD8DPWR.LC[4:0] bits. For the adjustment range, refer to "LCD Driver (LCD8D) Characteristics" in the "Electrical Characteristics" chapter.

21.5 Operations

21.5.1 Initialization

The LCD8D should be initialized with the procedure shown below.

1. Assign the LCD8D output function to the ports. (Refer to the "I/O Ports" chapter.)
2. Configure the LCD8DCLK.CLKSRC[1:0] and LCD8DCLK.CLKDIV[2:0] bits. (Configure operating clock)
3. Configure the following LCD8DCTL register bits:
   - Write 1 to the LCD8DCTL.MODEN bit. (Enable LCD8D operating clock)
   - Write 1 to the LCD8DCTL.LCDDIS bit. (Enable LCD driver pin discharge at display off)
4. Configure the following LCD8DTIM1 register bits:
   - LCD8DTIM1.LDUTY[2:0] bits (Set drive duty)
   - LCD8DTIM1.FRMCNT[3:0] bits (Set frame frequency)
5. Configure the following LCD8DTIM2 register bits:
   - LCD8DTIM2.LCDWAVE bit (Select drive waveform)
   - LCD8DTIM2.NLINE[2:0] bits (Set n-line inverse AC drive)
   - LCD8DTIM2.BSTC[1:0] bits (Set booster clock frequency)
6. Configure the LCD8DPWR.EXVCSEL bit. (Select external voltage application mode/internal generation mode)
7. Configure the following LCD8DPWR register bits:
   - LCD8DPWR.RESISEL[1:0] bits (Select internal voltage dividing resistors)
   - LCD8DPWR.LC[4:0] bits (Set LCD contrast initial value)
   - LCD8DPWR.BSTEN bit (Enable LCD voltage booster)
   - LCD8DPWR.BIASSEL bit (Select drive bias)
   - LCD8DPWR.VCSEL bit (Set reference voltage for boosting)
   - LCD8DPWR.VCEN bit (Enable LCD voltage regulator)

8. Configure the following LCD8DDSP register bits:
   - LCD8DDSP.DSPAR bit (Select display area)
   - LCD8DDSP.COMREV bit (Select COM pin assignment direction)
   - LCD8DDSP.SEGREV bit (Select SEG pin assignment direction)
9. Write display data to the display data RAM.
10. Set the following bits when using the interrupt:
    - Write 1 to the LCD8DINTF.FRMIF bit. (Clear interrupt flag)
    - Set the LCD8DINTE.FRMIE bit to 1. (Enable LCD8D interrupt)

21.5.2 Display On/Off

The LCD display state is controlled using the LCD8DDSP.DSPC[1:0] bits.

Table 21.5.2.1 LCD Display Control

LCD8DDSP.DSPC[1:0] bits	LCD display
0x3	All off (static drive)
0x2	All on
0x1	Normal display
0x0	Display off

Selecting “Display off” stops the drive voltage supply and the LCD driver pin outputs are all set to Vss level when the LCD8DCTL.LCDDIS bit = 1.
Since “All on” and “All off” directly control the driving waveform output by the LCD driver, data in the display data RAM is not altered. The common pins are set to dynamic drive for “All on” and to static drive for “All off.” This function can be used to make the display flash on and off without altering the display memory.

21.5.3 Inverted Display

The LCD panel display can be inverted (black/white inversion) using merely control bit manipulation, without rewriting the display data RAM. Setting the LCD8DDSP.DSPREV bit to 0 inverts the display; setting it to 1 returns the display to normal status. Note that the display will not be inverted when the LCD8DDSP.DSPC[1:0] bits = 0x3 (All off).

21.5.4 Drive Duty Switching

Drive duty can be set to 1/8 to 1/2 or static drive using the LCD8DTIM1.LDUTY[2:0] bits. Table 21.5.4.1 shows the correspondence between the LCD8DTIM1.LDUTY[2:0] bit settings, drive duty, and maximum number of display segments.

Table 21.5.4.1 Drive Duty Settings

LCD8DTIM1. LDUTY[2:0] bits	Duty	Valid COM pins	Valid SEG pins	Max. number of display dots/segments
0x7	1/8	COM0–COM7	SEG4–SEG55	416
0x6	1/7	COM0–COM6		364
0x5	1/6	COM0–COM5		312
0x4	1/5	COM0–COM4		260
0x3	1/4	COM0–COM3	SEG0–SEG55	224
0x2	1/3	COM0–COM2		168
0x1	1/2	COM0–COM1		112
0x0	Static	COM0		56

Unused common pins output an OFF waveform that turns the segments off.

The some pins are shared with a SEG output and a COM output, and they are configured to the SEG or COM pin according to the drive duty selected.

Table 21.5.4.2 SEG/COM Pin Configuration

Pin	**Duty**							
	1/8	**1/7**	**1/6**	**1/5**	**1/4**	**1/3**	**1/2**	**Static**
COM0	COM0							
COM1	COM1							Unused
COM2	COM2						Unused	
COM3	COM3					Unused		
COM4/SEG0/P67	COM4				SEG0 (P67)			
COM5/SEG1/P66	COM5			Unused (66)	SEG1 (P66)			
COM6/SEG2/P65	COM6		Unused (P65)		SEG2 (P65)			
COM7/SEG3/P64	COM7	Unused (P64)			SEG3 (P64)			
SEG4–12, 25–31, 48–52	SEG4–12, 25–31, 48–52							
SEG13–24, 32–47, 53–55/P*xx*	SEG13–24, 32–47, 53–55 (P*xx*)							

21.5.5 Drive Waveforms

LCD8D supports two types (Waveform A, Waveform B) of drive waveform outputs. The waveform type can be selected using the LCD8DTIM2.LCDWAVE bit. The following shows drive waveform examples.

Waveform A

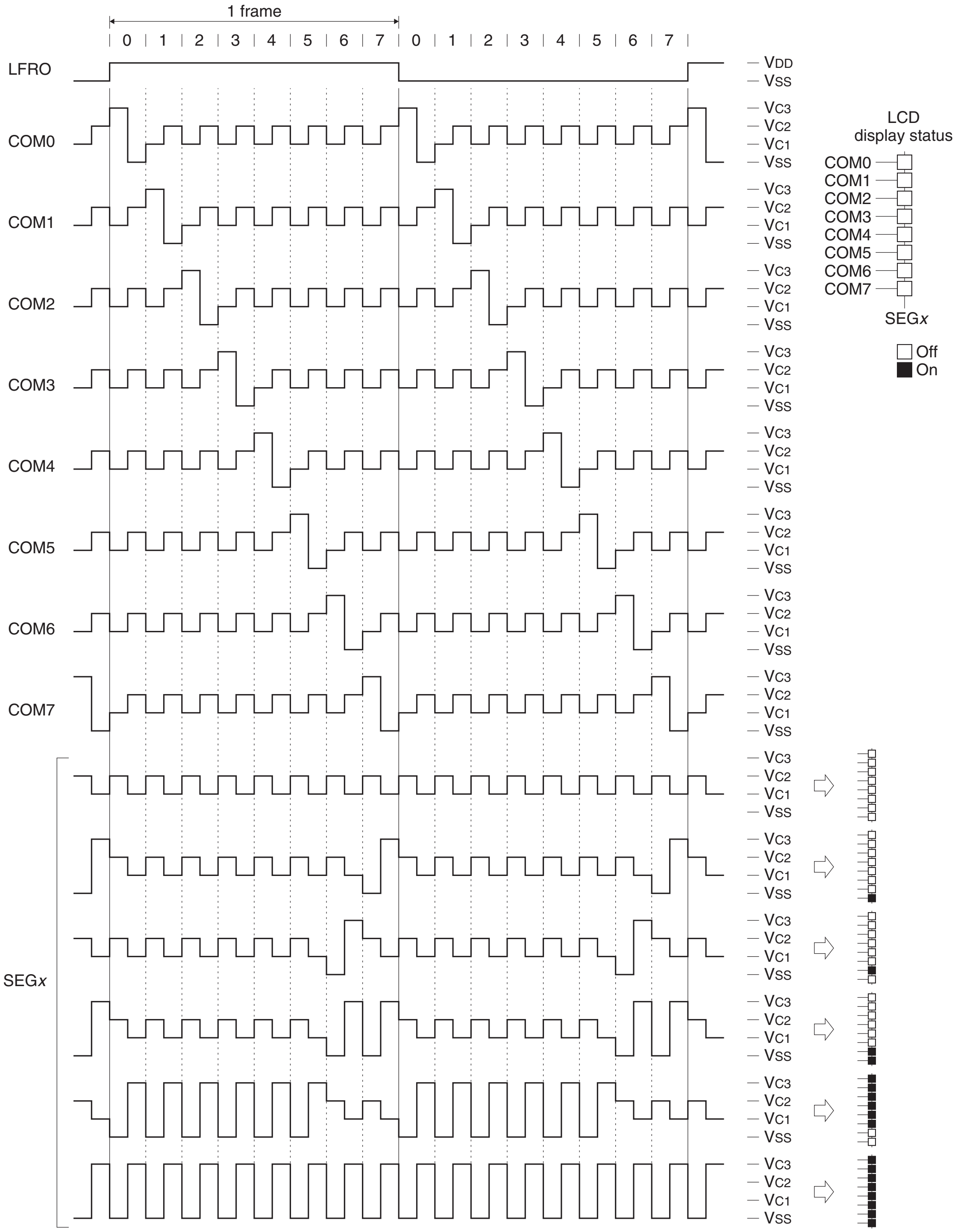


Figure 21.5.5.1 1/8 Duty Drive Waveform (Waveform A, 1/3 bias)

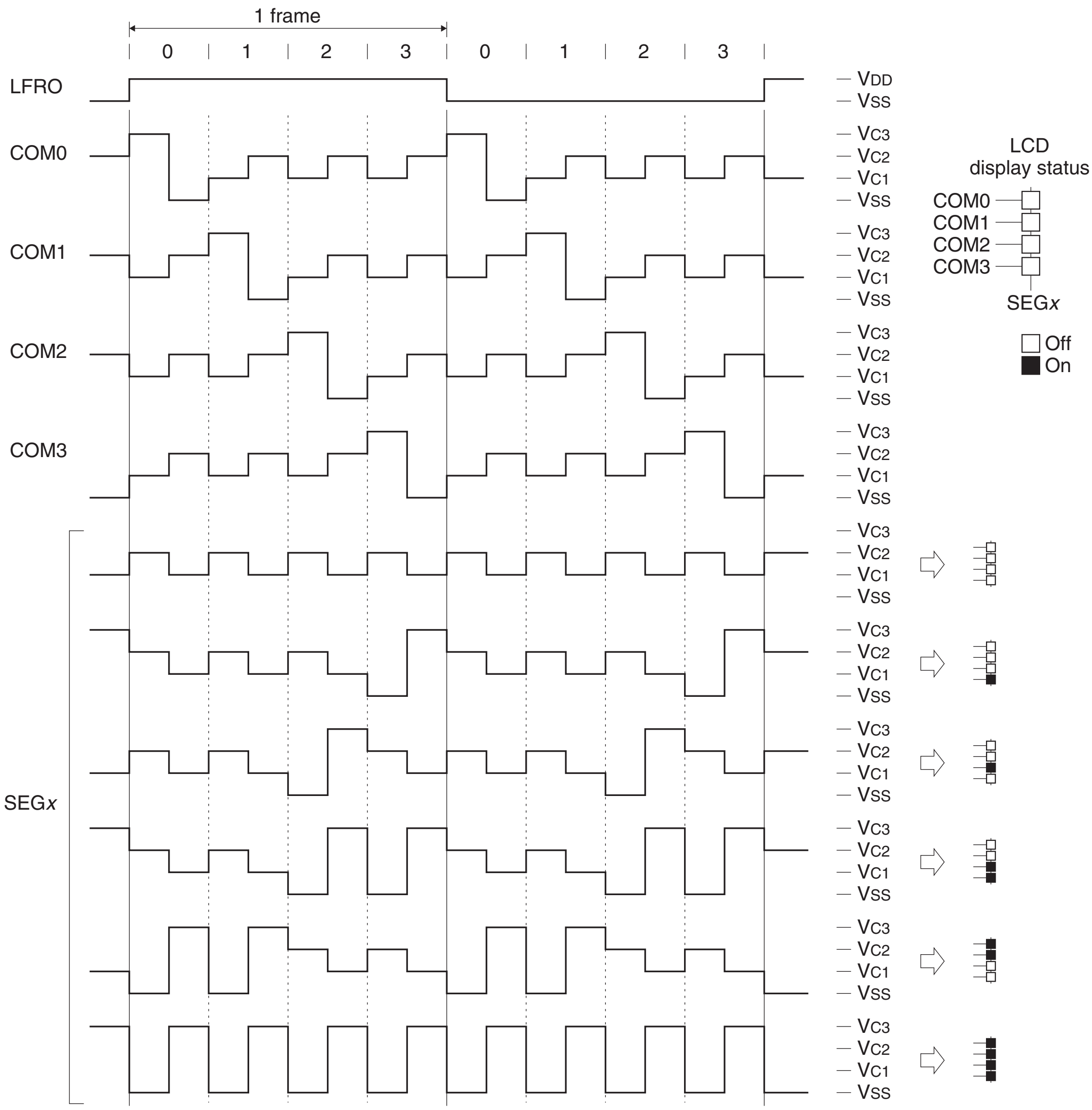


Figure 21.5.5.2 1/4 Duty Drive Waveform (Waveform A, 1/3 bias)

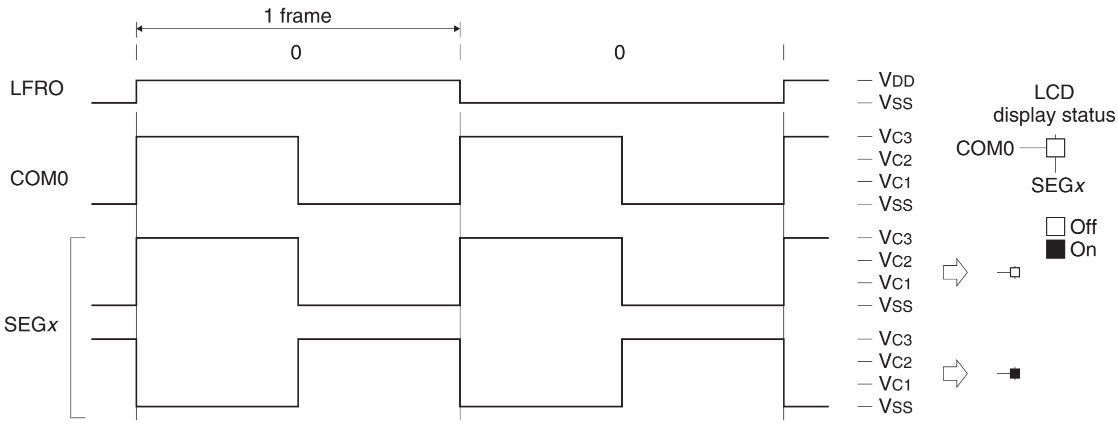


Figure 21.5.5.3 Static Drive Waveform (Waveform A, 1/3 bias)

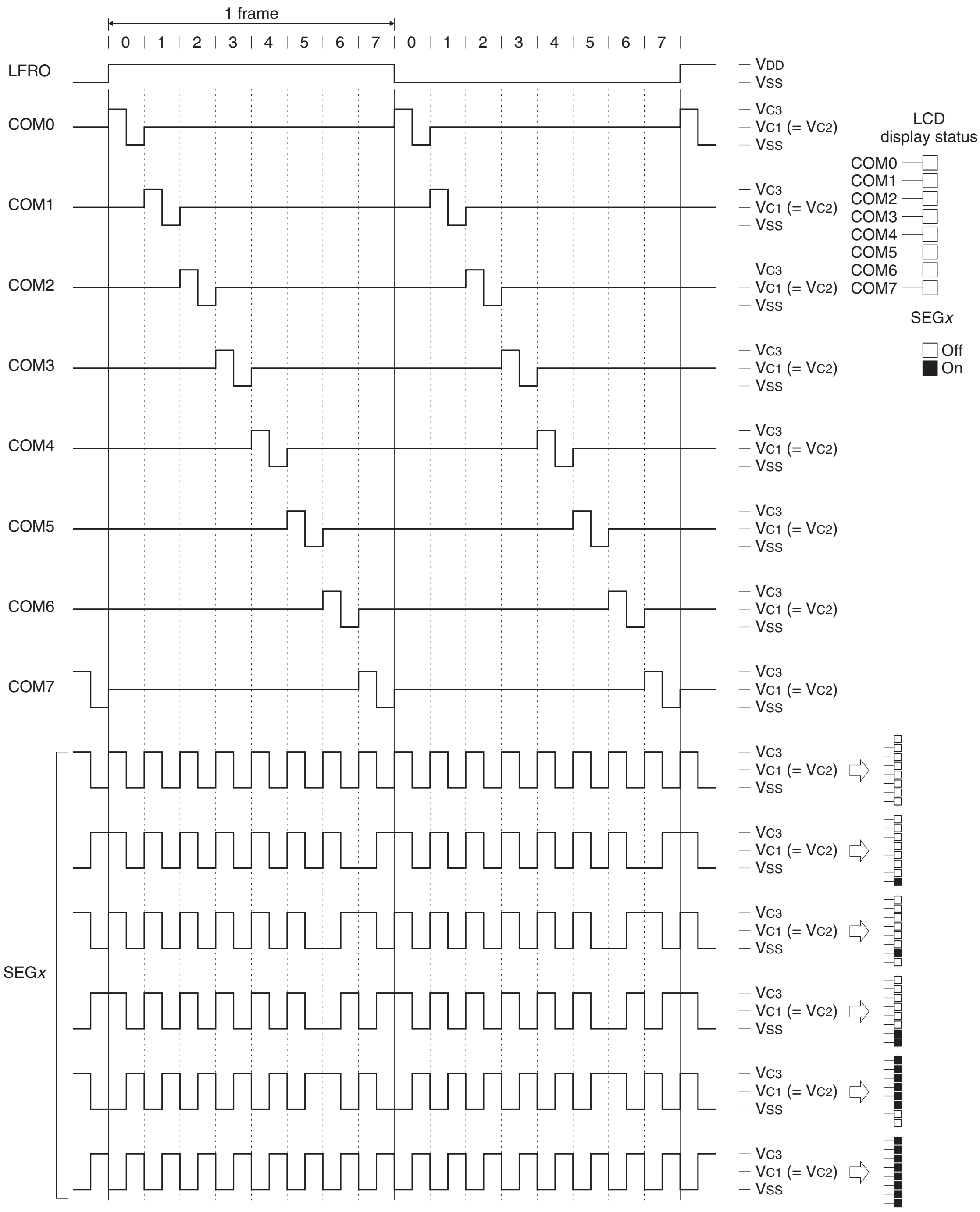


Figure 21.5.5.4 1/8 Duty Drive Waveform (Waveform A, 1/2 bias)

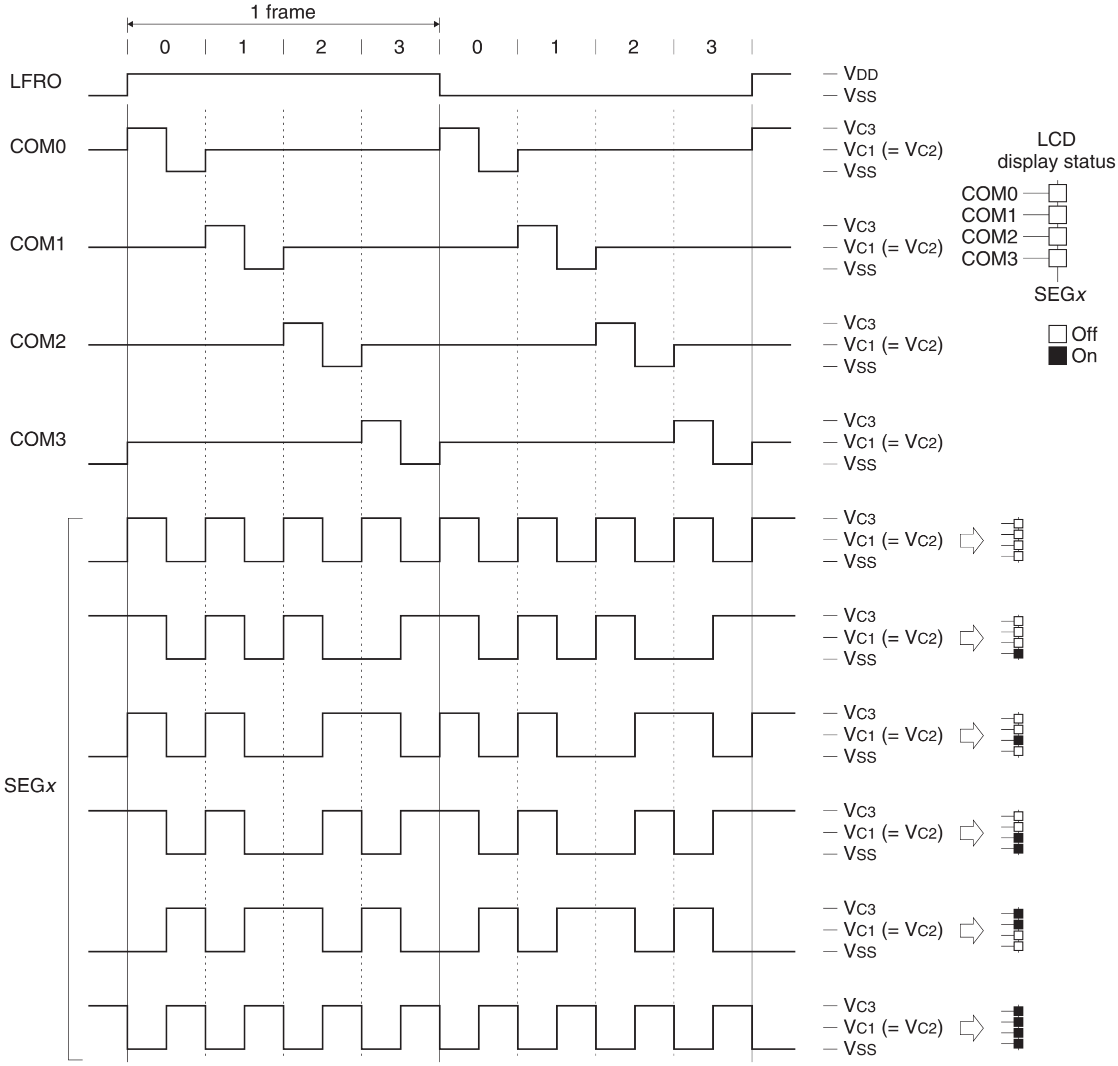


Figure 21.5.5.5 1/4 Duty Drive Waveform (Waveform A, 1/2 bias)

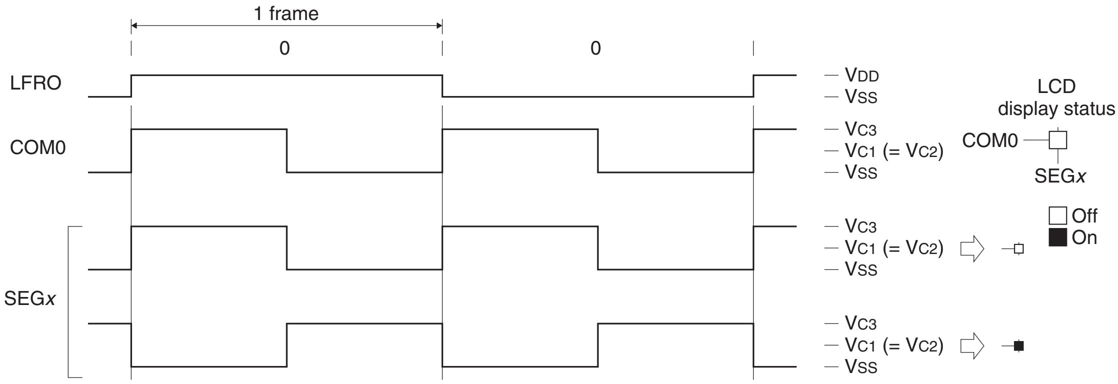


Figure 21.5.5.6 Static Drive Waveform (Waveform A, 1/2 bias)

Waveform B

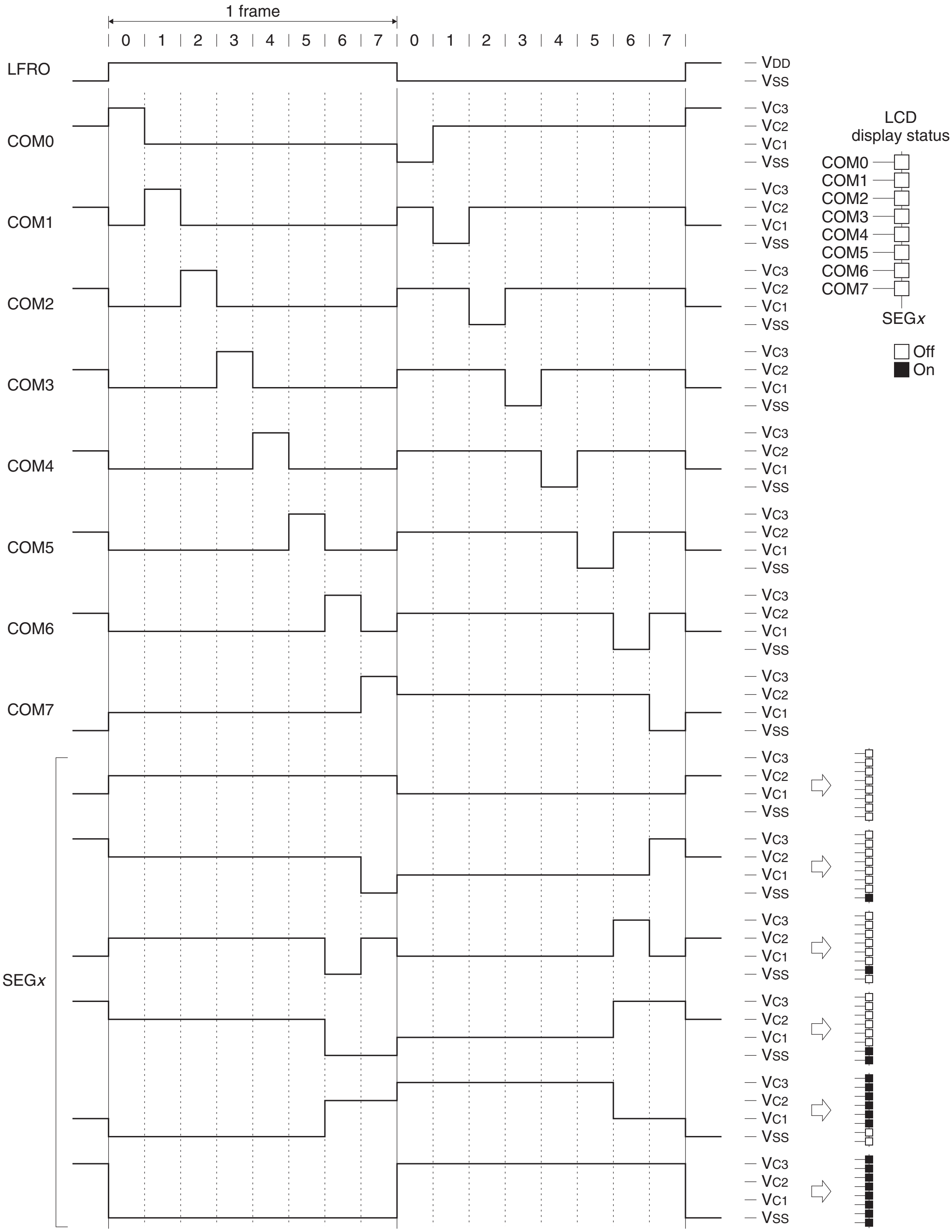


Figure 21.5.5.7 1/8 Duty Drive Waveform (Waveform B, 1/3 bias)

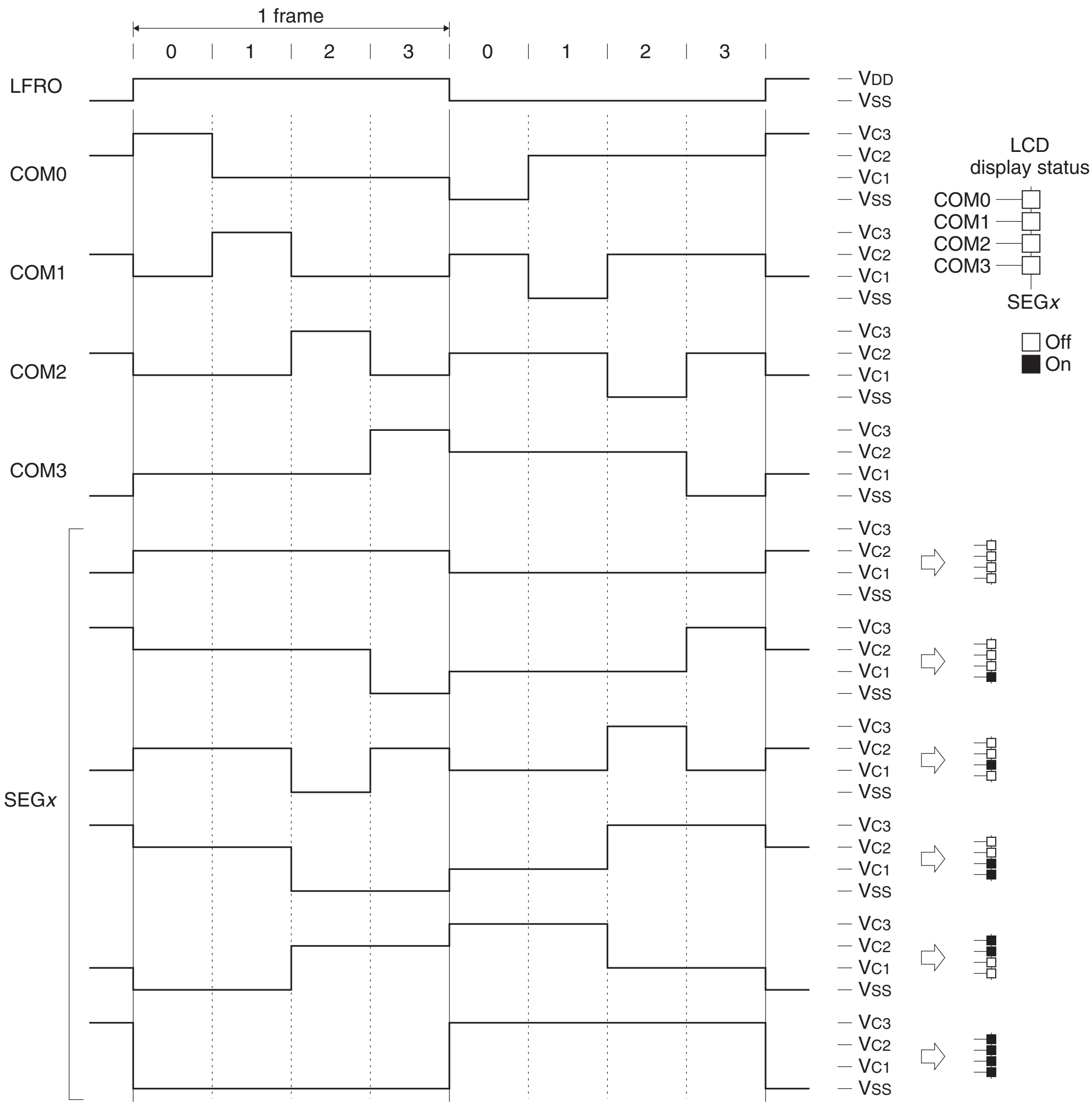


Figure 21.5.5.8 1/4 Duty Drive Waveform (Waveform B, 1/3 bias)

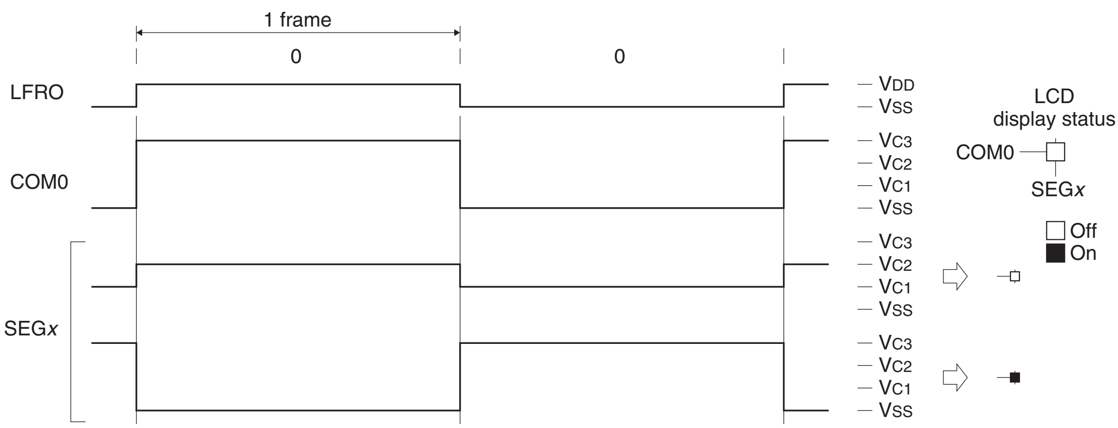


Figure 21.5.5.9 Static Drive Waveform (Waveform B, 1/3 bias)

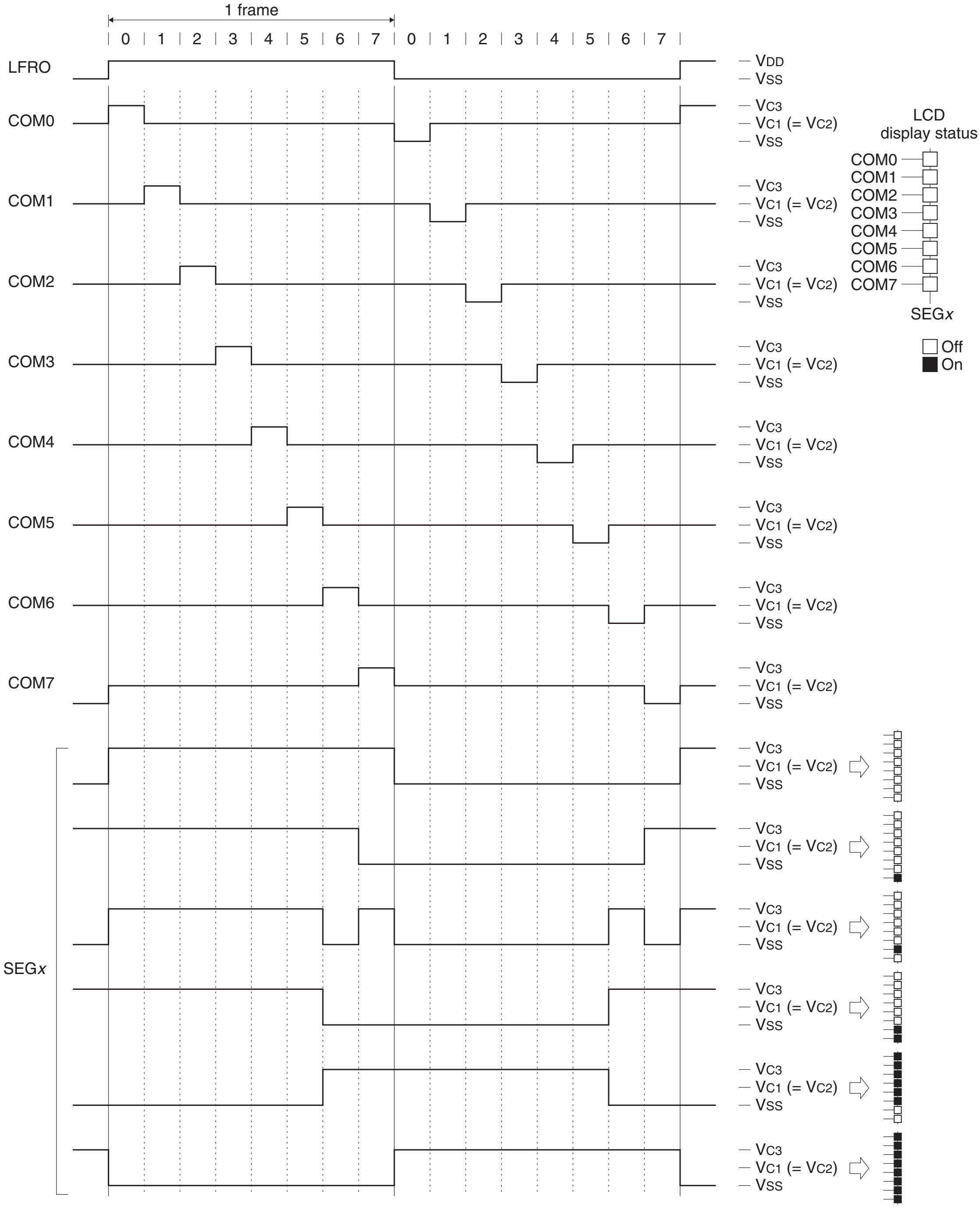


Figure 21.5.5.10 1/8 Duty Drive Waveform (Waveform B, 1/2 bias)

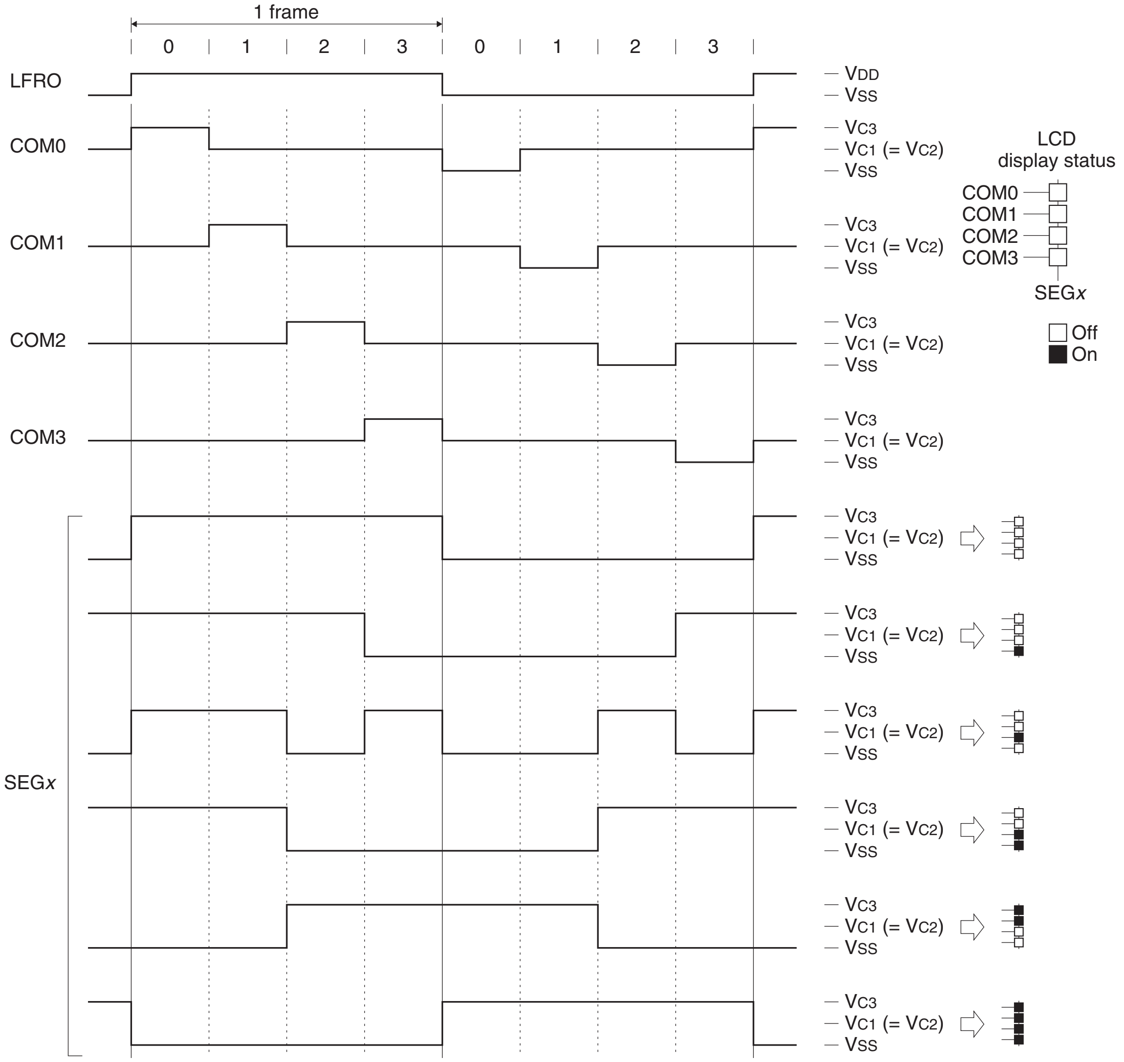


Figure 21.5.5.11 1/4 Duty Drive Waveform (Waveform B, 1/2 bias)

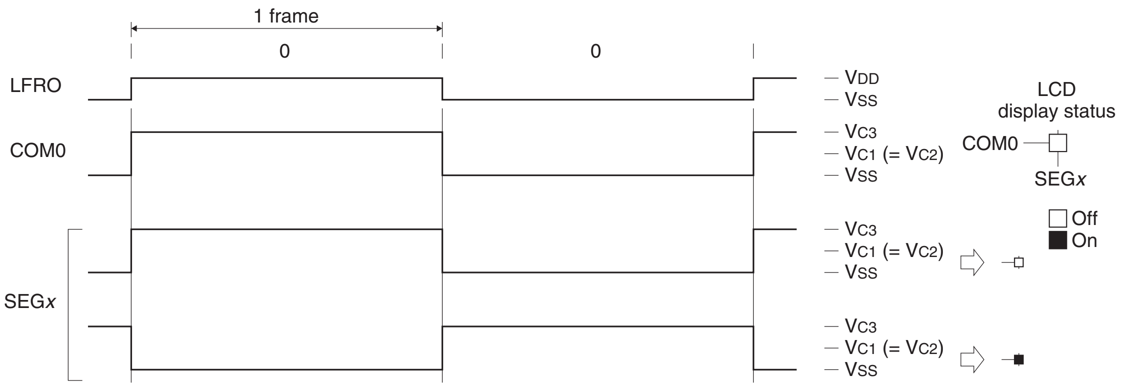


Figure 21.5.5.12 Static Drive Waveform (Waveform B, 1/2 bias)

21.5.6 Partial Common Output Drive

By setting the LCD8DOMC∗.COM*x*DEN bit (*x* = COM No.) to 0, any common outputs can be set to off waveform regardless of the display data RAM contents. The partial common output drive function limits the display to the required area only to reduce power consumption.

21.5.7 n-Segment-Line Inverse AC Drive

The n-line inverse AC drive function may improve the display quality when being reduced such as when crosstalk occurs. To activate the n-line inverse AC drive function, select the number of lines to be inverted using the LCD8DTIM2.NLINE[2:0] bits. The setting value should be determined after being evaluated using the actual circuit board. Note that using the n-line inverse AC drive function increases current consumption.

Table 21.5.7.1 Selecting Number of Inverse Lines

LCD8DTIM2.NLINE[2:0] bits	Number of inverse lines
0x7	7 lines
:	:
0x1	1 line
0x0	Normal drive

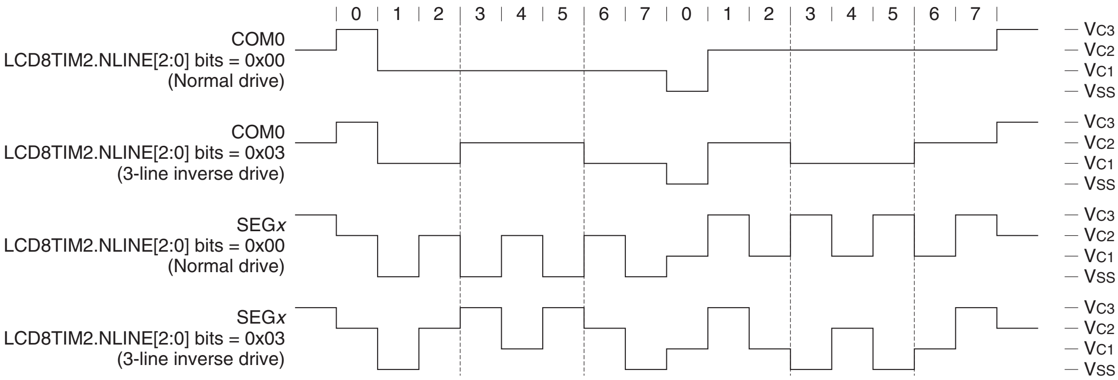


Figure 21.5.7.1 1/8 Duty (Waveform B, 1/3 bias) Normal Drive Waveform and 3-line Inverse Drive Waveform

Note: Do not use the n-line inverse AC drive function when Waveform A is selected.

21.6 Display Data RAM

The display data RAM is located beginning with address 0x7000.
The correspondence between the memory bits of the display data RAM and the common/segment pins varies depending on the selected conditions below.

- Drive duty (1/8 to 1/2 or static drive)
- Segment pin assignment (normal or inverse)
- Common pin assignment (normal or inverse)

Figures 21.6.3.1 to 21.6.3.4 show the correspondence between display data RAM and the common/segment pins in some drive duties.
Writing 1 to the display data RAM bit corresponding to a segment on the LCD panel turns the segment on, while writing 0 turns the segment off. Since the display memory is a RAM allowing reading and writing, bits can be controlled individually using logic operation instructions (read-modify-write instructions).
The area unused for display can be used as general-purpose RAM.

21.6.1 Display Area Selection

In the display data RAM, two screen areas can be allocated and the LCD8DDSP.DSPAR bit can be used to switch between the screens. Setting the LCD8DDSP.DSPAR bit to 0 selects display area 0; setting to 1 selects display area 1.

21.6.2 Segment Pin Assignment

The display data RAM address assignment for the segment pins can be inverted using the LCD8DDSP.SEGREV bit. When the LCD8DDSP.SEGREV bit is set to 1, memory addresses are assigned to segment pins in ascending order. When the LCD8DDSP.SEGREV bit is set to 0, memory addresses are assigned to segment pins in descending order.

21.6.3 Common Pin Assignment

The display data RAM bit assignment for the common pins can be inverted using the LCD8DDSP.COMREV bit. When the LCD8DDSP.COMREV bit is set to 1, memory bits are assigned to common pins in ascending order. When the LCD8DDSP.COMREV bit is set to 0, memory bits are assigned to common pins in descending order.

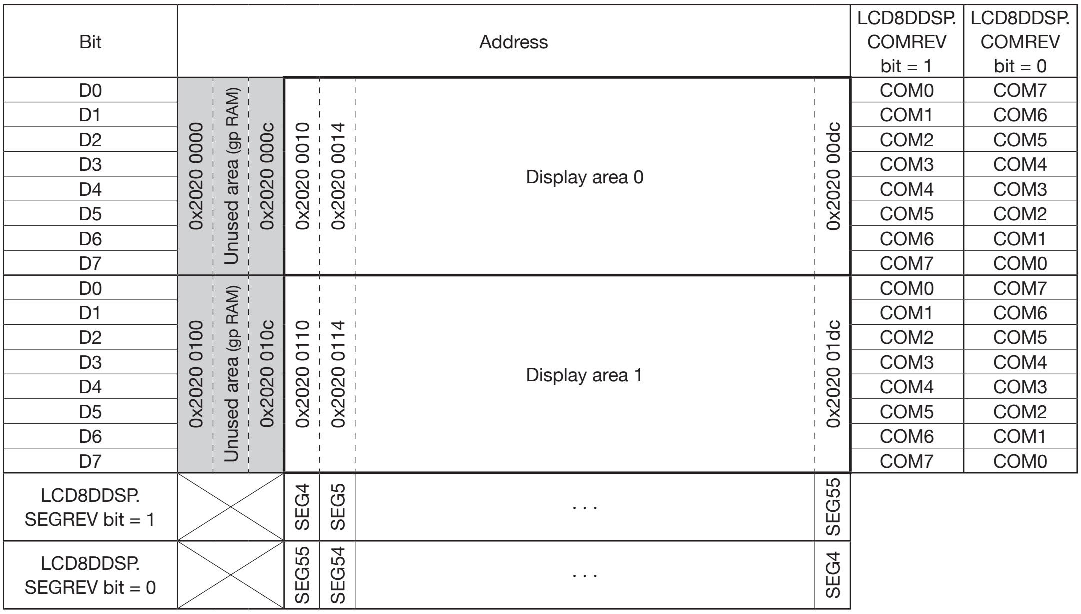


Figure 21.6.3.1 Display Data RAM Map (1/8 duty)

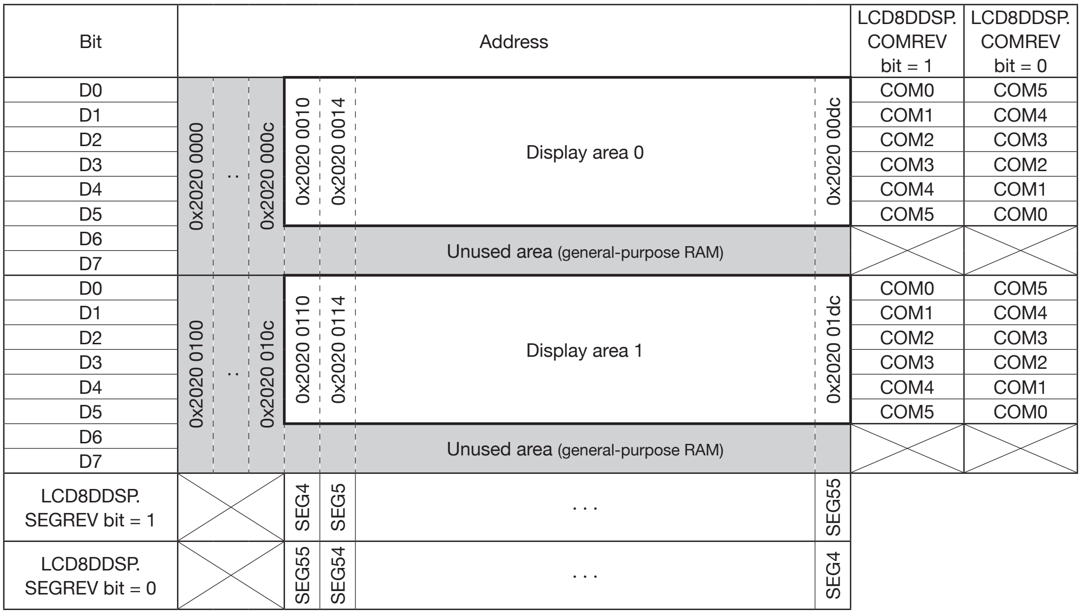


Figure 21.6.3.2 Display Data RAM Map (1/6 duty)

Bit	Address				LCD8DDSP. COMREV bit = 1	LCD8DDSP. COMREV bit = 0
D0	0x2020 0000	0x2020 0004	Display area 0	0x2020 00dc	COM0	COM3
D1					COM1	COM2
D2					COM2	COM1
D3					COM3	COM0
D4			Unused area (general-purpose RAM)			
D5						
D6						
D7						
D0	0x2020 0100	0x2020 0104	Display area 1	0x2020 01dc	COM0	COM3
D1					COM1	COM2
D2					COM2	COM1
D3					COM3	COM0
D4			Unused area (general-purpose RAM)			
D5						
D6						
D7						
LCD8DDSP. SEGREV bit = 1	SEG0	SEG1	. . .	SEG55		
LCD8DDSP. SEGREV bit = 0	SEG55	SEG54	. . .	SEG0		

Figure 21.6.3.3 Display Data RAM Map (1/4 duty)

Bit	Address				LCD8DDSP. COMREV bit = 1	LCD8DDSP. COMREV bit = 0
D0	0x2020 0000	0x2020 0004	Display area 0	0x2020 00dc	COM0	COM0
D1			Unused area (general-purpose RAM)			
D2						
D3						
D4						
D5						
D6						
D7						
D0	0x2020 0100	0x2020 0104	Display area 1	0x2020 01dc	COM0	COM0
D1			Unused area (general-purpose RAM)			
D2						
D3						
D4						
D5						
D6						
D7						
LCD8DDSP. SEGREV bit = 1	SEG0	SEG1	. . .	SEG55		
LCD8DDSP. SEGREV bit = 0	SEG55	SEG54	. . .	SEG0		

Figure 21.6.3.4 Display Data RAM Map (static drive)

Note: No physical memory is allocated to D8 through D31 of each address.

21.7 Interrupt

The LCD8D has a function to generate the interrupt shown in Table 21.7.1.

Table 21.7.1 LCD8D Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Frame	LCD8DINTF.FRMIF	Frame switching	Writing 1

The LCD8D provides an interrupt enable bit corresponding to the interrupt flag. An interrupt request is sent to the interrupt controller only when the interrupt flag, of which interrupt has been enabled by the interrupt enable bit, is set. For more information on interrupt control, refer to the “Interrupt Controller” chapter.

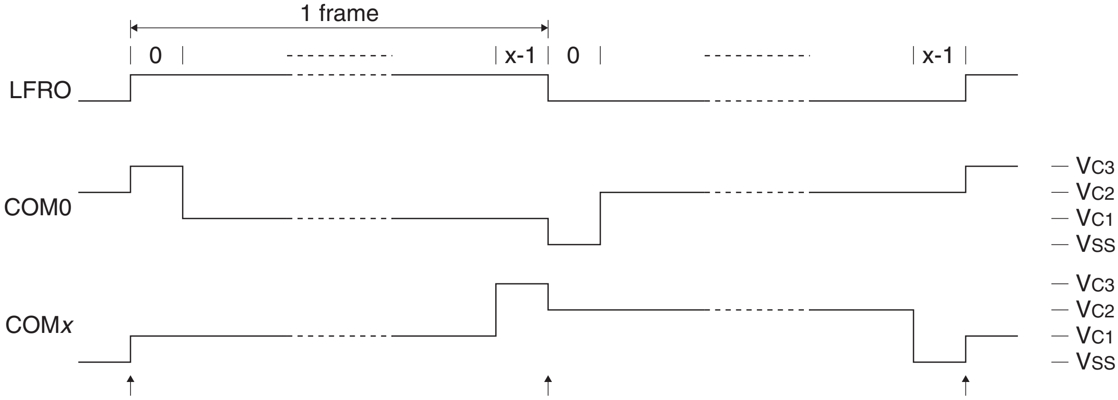


Figure 21.7.1 Frame Interrupt Timings (Waveform B, 1/x duty, 1/3 bias)

21.8 Control Registers

LCD8D Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DCLK	15–9	–	0x00	–	R	–
	8	DBRUN	1	H0	R/W	
	7	–	0	–	R	
	6–4	CLKDIV[2:0]	0x0	H0	R/W	
	3–2	–	0x0	–	R	
	1–0	CLKSRC[1:0]	0x0	H0	R/W	

Bits 15–9 Reserved

Bit 8 DBRUN

This bit sets whether the LCD8D operating clock is supplied in DEBUG mode or not.

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

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

Bit 7 Reserved

Bits 6–4 CLKDIV[2:0]

These bits select the division ratio of the LCD8D operating clock.

Bits 3–2 Reserved

Bits 1–0 CLKSRC[1:0]

These bits select the clock source of the LCD8D.

Table 21.8.1 Clock Source and Division Ratio Settings

LCD8DCLK. CLKDIV[2:0] bits	LCD8DCLK.CLKSRC[1:0] bits			
	0x0	0x1	0x2	0x3
	IOSC	OSC1	OSC3	EXOSC
0x7	Reserved	1/1	Reserved	1/1
0x6	1/1,024		1/1,024	
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 LCD8DCLK register settings can be altered only when the LCD8DCTL.MODEN bit = 0.

LCD8D Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DCTL	15–8	–	0x00	–	R	–
	7–2	–	0x00	–	R	
	1	LCDDIS	0	H0	R/W	
	0	MODEN	0	H0	R/W	

Bits 15–2 Reserved

Bit 1 LCDDIS

This bit enables the SEG/COM-pin discharge operations when "Display off" is selected.

1 (R/W): Enable SEG/COM-pin discharge operations

0 (R/W): Disable SEG/COM-pin discharge operations

Setting this bit to 1 configures the SEG/COM pins to output a low level when "Display off" is selected. Setting to 0 configures the SEG/COM pins to enter Hi-Z status when "Display off" is selected.

Bit 0 MODEN

This bit enables the LCD8D operations.

1 (R/W): Enable LCD8D operations

0 (R/W): Disable LCD8D operations

Setting this bit to 1 starts supplying the operating clock to LCD8D.

Note: If the LCD8DCTL.MODEN bit is altered from 1 to 0 while the LCD panel is displaying, the LCD display is automatically turned off and the LCD8DDSP.DSPC[1:0] bits are also set to 0x0. Also the LCD voltage regulator is automatically turned off and the LCD8DPWR.VCEN bit is set to 0.

LCD8D Timing Control Register 1

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DTIM1	15–12	–	0x0	–	R	–
	11–8	FRMCNT[3:0]	0x3	H0	R/W	
	7–6	–	0x0	–	R	
	5	(reserved)	0	H0	R/W	
	4–3	–	0x0	–	R	
	2–0	LDUTY[2:0]	0x7	H0	R/W	

Bits 15–12 Reserved

Bits 11–8 FRMCNT[3:0]

These bits set the frame frequency. For more information, refer to "Frame Frequency."

Bits 7–3 Reserved

Bits 2–0 LDUTY[2:0]

These bits set the drive duty. For more information, refer to "Drive Duty Switching."

LCD8D Timing Control Register 2

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DTIM2	15	LCDWAVE	0	H0	R/W	–
	14–10	–	0x00	–	R	
	9–8	BSTC[1:0]	0x1	H0	R/W	
	7–3	–	0x00	–	R	
	2–0	NLINE[2:0]	0x0	H0	R/W	

Bit 15 LCDWAVE

This bit selects the drive waveform.

1 (R/W): Waveform A

0 (R/W): Waveform B

Bits 14–10 Reserved

Bits 9–8 BSTC[1:0]

These bits select the booster clock frequency for the LCD voltage booster.

Table 21.8.2 Booster Clock Frequency

LCD8DTIM2.BSTC[1:0] bits	Booster clock frequency [Hz]
0x3	f_{CLK_LCD8D}/64
0x2	f_{CLK_LCD8D}/32
0x1	f_{CLK_LCD8D}/16
0x0	f_{CLK_LCD8D}/4

f_{CLK_LCD8D}: LCD8D operating clock frequency [Hz]

Note: Do not alter the LCD8DTIM2.BSTC[1:0] bits from the initial value when using a model that does not have an LCD power supply.

Bits 7–3 Reserved

Bits 2–0 NLINE[2:0]

These bits enable the n-line inverse AC drive function and set the number of inverse lines. However, these bits should be fixed at 0x0 when the drive waveform is set to Waveform A. For more information, refer to "n-Segment-Line Inverse AC Drive."

LCD8D Power Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DPWR	15	EXVCSEL	1	H0	R/W	–
	14–13	RESISEL[1:0]	0x0	H0	R/W	
	12–8	LC[4:0]	0x00	H0	R/W	
	7–5	–	0x0	–	R	
	4	BSTEN	0	H0	R/W	
	3	BIASSEL	1	H0	R/W	
	2	HVLD	0	H0	R/W	
	1	VCSEL	0	H0	R/W	
	0	VCEN	0	H0	R/W	

Bit 15 EXVCSEL

This bit selects the LCD drive power supply mode (external voltage application mode or internal generation mode).

1 (R/W): External voltage application mode

0 (R/W): Internal generation mode

Note: Be sure to avoid applying voltages to the V_{C1} to V_{C3} pins when the LCD8DPWR.EXVCSEL bit is set to 0, as the LCD power supply pins are short-circuited to GND.

Bits 14–13 RESISEL[1:0]

These bits select the internal LCD voltage dividing resistor value.

Table 21.8.3 Internal LCD Voltage Divider Resistor Value Adjustment

LCD8DPWR.RESI[1:0] bits	Internal resistor value
0x3	Large
0x2	↕
0x1	Small
0x0	Internal voltage dividing resistors are not used.

Bits 12–8 LC[4:0]

These bits set the LCD panel contrast.

Table 21.8.4 LCD Contrast Adjustment

LCD8DPWR.LC[4:0] bits	Contrast
0x1f	High (dark)
0x1e	↑
:	:
0x01	↓
0x00	Low (light)

Bits 7–5 Reserved

Bit 4 BSTEN

This bit turns the LCD voltage booster on and off.

1 (R/W): LCD voltage booster on

0 (R/W): LCD voltage booster off

For more information, refer to “LCD Power Supply.”

Bit 3 BIASSEL

This bit selects the LCD drive bias.

1 (R/W): 1/3 bias

0 (R/W): 1/2 bias

Bit 2 HVLD

This bit sets the LCD voltage regulator into heavy load protection mode.

1 (R/W): Heavy load protection mode

0 (R/W): Normal mode

For more information, refer to “LCD Voltage Regulator Settings.”

Bit 1 VCSEL

This bit sets the LCD voltage regulator output (reference voltage for boosting).

1 (R/W): V_{C2}

0 (R/W): V_{C1}

For more information, refer to “LCD Voltage Regulator Settings.”

Note: The LCD8DPWR.VCSEL bit must be set to 0 in an external voltage application mode.

Bit 0 VCEN

This bit turns the LCD voltage regulator on and off.

1 (R/W): LCD voltage regulator on

0 (R/W): LCD voltage regulator off

For more information, refer to “LCD Power Supply.”

LCD8D Display Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DDSP	15–8	–	0x00	–	R	–
	7	–	0	–	R	
	6	SEGREV	1	H0	R/W	
	5	COMREV	1	H0	R/W	
	4	DSPREV	1	H0	R/W	
	3	–	0	–	R	
	2	DSPAR	0	H0	R/W	
	1–0	DSPC[1:0]	0x0	H0	R/W	

Bits 15–7 Reserved

Bit 6 SEGREV

This bit selects the segment pin assignment direction.

1 (R/W): Normal assignment

0 (R/W): Inverse assignment

For more information, see Figures 21.6.3.1 to 21.6.3.4.

Bit 5 COMREV

This bit selects the common pin assignment direction.

1 (R/W): Normal assignment

0 (R/W): Inverse assignment

For more information, see Figures 21.6.3.1 to 21.6.3.4.

Note: Do not set the LCD8DDSP.COMREV bit to 0 when the LCD8DTIM1.LDUTY[2:0] bits = 0x4–0x6.

Bit 4 DSPREV

This bit controls black/white inversion on the LCD display.

1 (R/W): Normal display

0 (R/W): Inverted display

Bit 3 Reserved

Bit 2 DSPAR

This bit switches the display area in the display data RAM.

1 (R/W): Display area 1

0 (R/W): Display area 0

Bits 1–0 DSPC[1:0]

These bits control the LCD display on/off and select a display mode. For more information, refer to "Display On/Off."

LCD8D COM Pin Control Register 0

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DCOMC0	15–8	–	0x00	–	R	–
	7	COM7DEN	1	H0	R/W	
	6	COM6DEN	1	H0	R/W	
	5	COM5DEN	1	H0	R/W	
	4	COM4DEN	1	H0	R/W	
	3	COM3DEN	1	H0	R/W	
	2	COM2DEN	1	H0	R/W	
	1	COM1DEN	1	H0	R/W	
	0	COM0DEN	1	H0	R/W	

Bits 15–8 Reserved

Bits 7–0 **COM*x*DEN**

These bits configure the partial drive of the COM*x* pins.

1 (R/W): Normal output

0 (R/W): Off waveform output

LCD8D Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DINTF	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	FRMIF	0	H0	R/W	Cleared by writing 1.

Bits 15–1 **Reserved**

Bit 0 **FRMIF**

This bit indicates the frame interrupt cause occurrence status.

1 (R): Cause of interrupt occurred

0 (R): No cause of interrupt occurred

1 (W): Clear flag

0 (W): Ineffective

LCD8D Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
LCD8DINTE	15–8	–	0x00	–	R	–
	7–1	–	0x00	–	R	
	0	FRMIE	0	H0	R/W	

Bits 15–1 **Reserved**

Bit 0 **FRMIE**

This bit enables the frame interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt

22 R/F Converter (RFC)

22.1 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 22.1.1 shows the RFC configuration.

Table 22.1.1 RFC Channel Configuration of S1C31W65

Item	S1C31W65
Number of channels	1 channel (Ch.0) **Note**: DC oscillation mode for resistive sensor measurements can only be used.

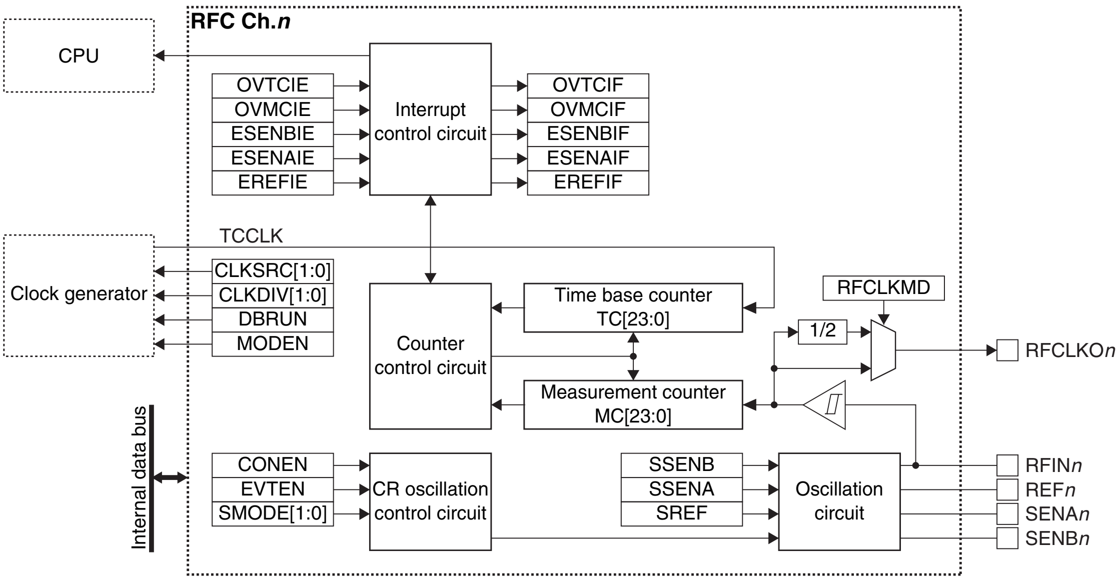


Figure 22.1.1 RFC Configuration

22.2 Input/Output Pins and External Connections

22.2.1 List of Input/Output Pins

Table 22.2.1.1 lists the RFC pins.

Table 22.2.1.1 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	VSS	RFCLK input or oscillation control pin
RFCLKO*n*	O	Hi-Z	RFCLK monitoring output pin RFCLK is output to monitor the oscillation frequency.

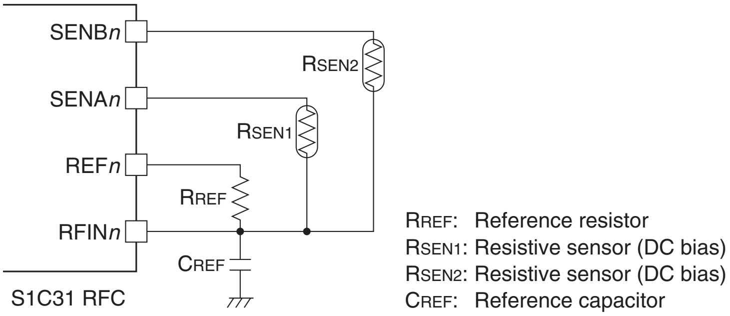
* Indicates the status when the pin is configured for the RFC.

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 VSS level when the port is switched. Be aware that large current may flow if the pin is biased by an external circuit.

22.2.2 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 22.2.2.1 Connection Example in Resistive Sensor DC Oscillation Mode

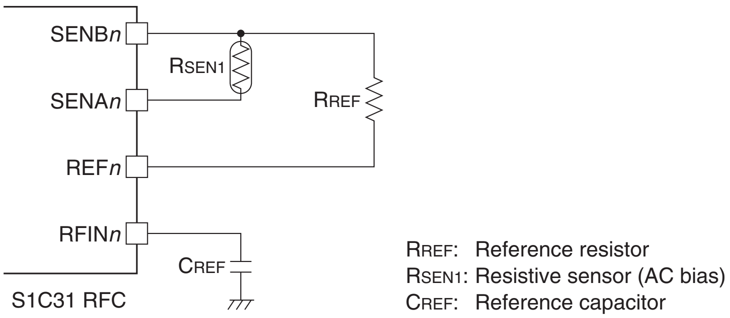
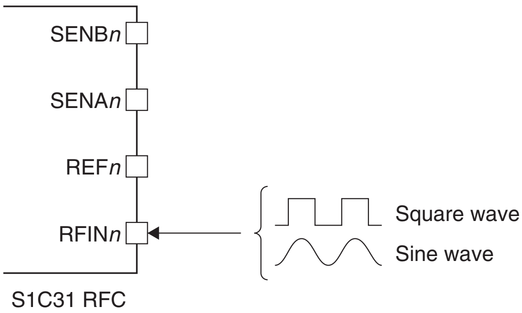


Figure 22.2.2.2 Connection Example in Resistive Sensor AC Oscillation Mode



∗ Leave the unused pins open.

Figure 22.2.2.3 External Clock Input in External Clock Input Mode

22.3 Clock Settings

22.3.1 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 RFC_*n*CLK register bits:
   - RFC_*n*CLK.CLKSRC[1:0] bits (Clock source selection)
   - RFC_*n*CLK.CLKDIV[1:0] bits (Clock division ratio selection = Clock frequency setting)

The time base counter performs counting with TCCLK set here. Selecting a higher clock results in higher conversion accuracy, note, however, that the frequency should be determined so that the time base counter will not overflow during reference oscillation.

22.3.2 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.*xxxx*SLPC bit for the TCCLK clock source.

22.3.3 Clock Supply in DEBUG Mode

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

22.4 Operations

22.4.1 Initialization

The RFC should be initialized with the procedure shown below.

1. Configure the RFC_*n*CLK.CLKSRC[1:0] and RFC_*n*CLK.CLKDIV[1:0] bits. (Configure operating clock)
2. Set the following bits when using the interrupt:
   - Write 1 to the interrupt flags in the RFC_*n*INTF register. (Clear interrupt flags)
   - Set the interrupt enable bits in the RFC_*n*INTE register to 1. (Enable interrupts)
3. Assign the RFC input/output function to the ports. (Refer to the "I/O Ports" chapter.)

4. Configure the following RFC_*n*CTL register bits:
   - RFC_*n*CTL.EVTEN bit (Enable/disable external clock input mode)
   - RFC_*n*CTL.SMODE[1:0] bits (Select oscillation mode)
   - Set the RFC_*n*CTL.MODEN bit to 1. (Enable RFC operations)

22.4.2 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 RFC_*n*CTL.SMODE[1:0] bits.

DC oscillation mode for resistive sensor measurements

This mode performs measurements by DC driving the reference resistor and the resistive sensor to oscillate. Set the 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 RFC_*n*CTL.EVTEN bit to 1. The measurement procedure is the same as when the internal oscillation circuit is used.

22.4.3 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 <Initial value: n> = <Counter value at the end of sensor oscillation: m> (if error = 0). Setting a large <Initial value: n> 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.

22.4.4 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 RFC_nMCH and RFC_nMCL registers (measurement counter).
2. Clear the RFC_nTCH and RFC_nTCL registers (time base counter) to 0x000000.
3. Clear both the RFC_nINTF.EREFIF and RFC_nINTF.OVTCIF bits by writing 1.
4. Set the RFC_nTRG.SREF bit to 1 to start reference oscillation.
5. Wait for an RFC interrupt.
   i. If the RFC_nINTF.EREFIF bit = 1 (reference oscillation completion), clear the RFC_nINTF.EREFIF bit and then go to Step 6.
   ii. If the RFC_nINTF.OVTCIF bit = 1 (time base counter overflow error), clear the RFC_nINTF.OVTCIF bit and terminate measurement as an error or retry after altering the measurement counter initial value.
6. Clear the RFC_nINTF.ESENAIF, RFC_nINTF.ESENBIF, and RFC_nINTF.OVMCIF bits by writing 1.
7. Set the RFC_nTRG.SSENA bit (sensor A) or the RFC_nTRG.SSENB bit (sensor B) corresponding to the sensor to be measured to 1 to start sensor oscillation (use the RFC_nTRG.SSENA bit in AC oscillation mode).
8. Wait for an RFC interrupt.
   i. If the RFC_nINTF.ESENAIF bit = 1 (sensor A oscillation completion) or the RFC_nINTF.ESENBIF bit = 1 (sensor B oscillation completion), clear the RFC_nINTF.ESENAIF or RFC_nINTF.ESENBIF bit and then go to Step 9.
   ii. If the RFC_nINTF.OVMCIF bit = 1 (measurement counter overflow error), clear the RFC_nINTF.OVMCIF bit and terminate measurement as an error or retry after altering the measurement counter initial value.
9. Read the RFC_nMCH and RFC_nMCL registers (measurement counter) and correct the results depending on the sensor to obtain the detected value.

R/F converting operations

Reference oscillation

When the RFC_nTRG.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 (0xffffff → 0x000000), the RFC_nTRG.SREF bit is cleared to 0 and the reference oscillation stops automatically.
The measurement counter overflow sets the RFC_nINTF.EREFIF bit to 1 indicating that the reference oscillation has been terminated normally. If the RFC_nINTE.EREFIE bit = 1, a reference oscillation completion interrupt request occurs at this point.

The time base counter overflow sets the RFC_*n*INTF.OVTCIF bit to 1 indicating that the reference oscillation has been terminated abnormally. If the RFC_*n*INTE.OVTCIE bit = 1, a time base counter overflow error interrupt request occurs at this point.

Sensor oscillation

When the RFC_*n*TRG.SSENA bit (sensor A) or the RFC_*n*TRG.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 (0xffffff → 0x000000), the RFC_*n*TRG.SSENA bit or the RFC_*n*TRG.SSENB bit that started oscillation is cleared to 0 and the sensor oscillation stops automatically.

The time base counter reaching 0x000000 sets the RFC_*n*INTF.ESENAIF bit (sensor A) or the RFC_*n*INTF.ESENBIF bit (sensor B) to 1 indicating that the sensor oscillation has been terminated normally. If the RFC_*n*INTE.ESENAIE bit = 1 or the RFC_*n*INTE.ESENBIE bit = 1, a sensor A or sensor B oscillation completion interrupt request occurs at this point.

The measurement counter overflow sets the RFC_*n*INTF.OVMCIF to 1 indicating that the sensor oscillation has been terminated abnormally. If the RFC_*n*INTE.OVMCIE bit = 1, a measurement counter overflow error interrupt request occurs at this point.

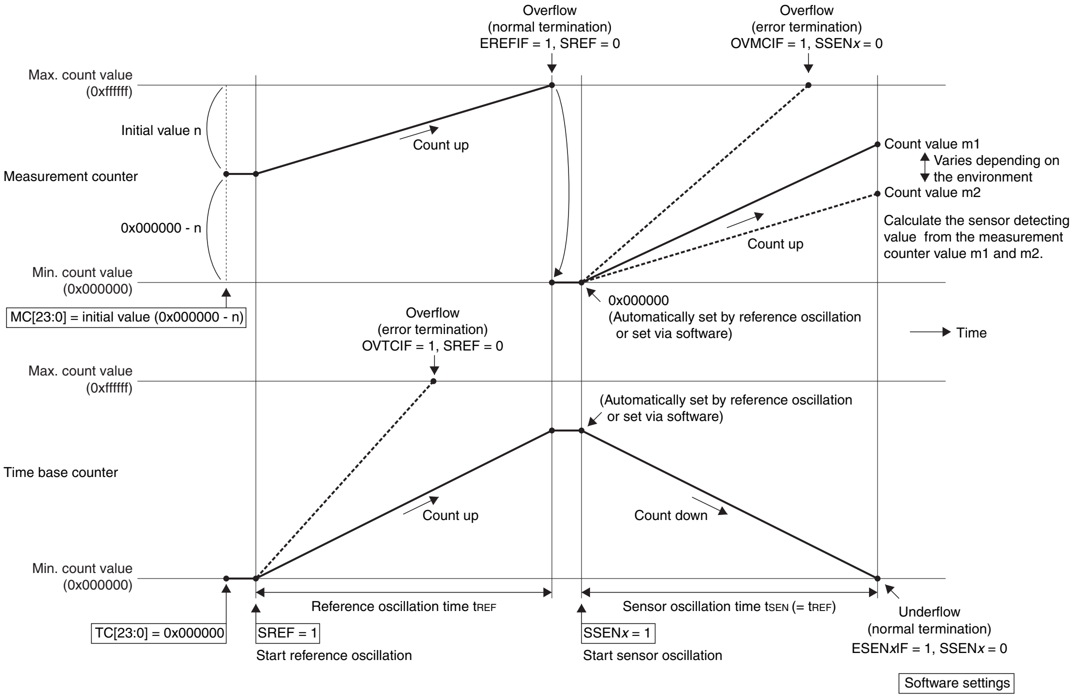


Figure 22.4.4.1 Counter Operations During Reference/Sensor Oscillation

Forced termination

To abort reference oscillation or sensor oscillation, write 0 to the RFC_*n*TRG.SREF bit (reference oscillation), the RFC_*n*TRG.SSENA bit (sensor A oscillation), or the RFC_*n*TRG.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 ≈ m. The difference between n and m is a conversion error. Table 22.4.4.1 lists the error factors. (n: measurement counter initial value, m: measurement counter value at the end of sensor oscillation)

Table 22.4.4.1 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

22.4.5 CR Oscillation Frequency Monitoring Function

The CR oscillation clock (RFCLK) generated during converting operation can be output from the RFCLKO*n* pin for monitoring. By setting the RFC_*n*CTL.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 RFC_*n*TRG.SREF bit (reference oscillation), the RFC_*n*TRG.SSENA bit (sensor A oscillation), or the RFC_*n*TRG.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 RFC_*n*CTL.RFCLKMD bit to 1 changes the output clock to the divided-by-two RFCLK clock.

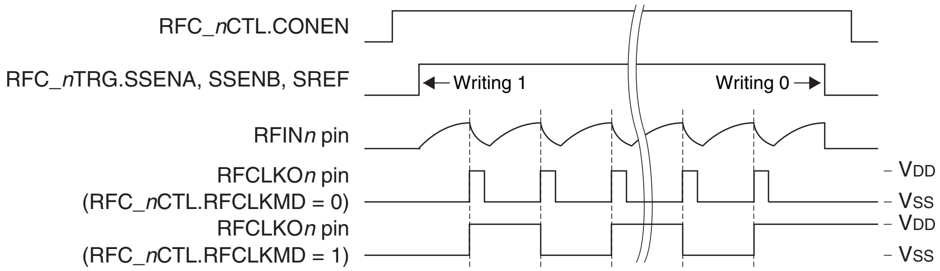


Figure 22.4.5.1 CR Oscillation Clock (RFCLK) Waveform

22.5 Interrupts

The RFC has a function to generate the interrupts shown in Table 22.5.1.

Table 22.5.1 RFC Interrupt Function

Interrupt	Interrupt flag	Set condition	Clear condition
Reference oscillation completion	RFC_*n*INTF.EREFIF	When reference oscillation has been completed normally due to a measurement counter overflow	Writing 1
Sensor A oscillation completion	RFC_*n*INTF.ESENAIF	When sensor A oscillation has been completed normally due to the time base counter reaching 0x000000	Writing 1
Sensor B oscillation completion	RFC_*n*INTF.ESENBIF	When sensor B oscillation has been completed normally due to the time base counter reaching 0x000000	Writing 1
Measurement counter overflow error	RFC_*n*INTF.OVMCIF	When sensor oscillation has been terminated abnormally due to a measurement counter overflow	Writing 1
Time base counter overflow error	RFC_*n*INTF.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 CPU 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” chapter.

22.6 Control Registers

RFC Ch.*n* Clock Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC_*n*CLK	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 22.6.1 Clock Source and Division Ratio Settings

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

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

Note: The RFC_*n*CLK register settings can be altered only when the RFC_*n*CTL.MODEN bit = 0.

RFC Ch.*n* Control Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC_*n*CTL	15–9	–	0x00	–	R	–
	8	RFCLKMD	0	H0	R/W	
	7	CONEN	0	H0	R/W	
	6	EVTEN	0	H0	R/W	
	5–4	SMODE[1:0]	0x0	H0	R/W	
	3–1	–	0x0	–	R	
	0	MODEN	0	H0	R/W	

Bits 15–9 Reserved

Bit 8 RFCLKMD

This bit sets the RFCLKO*n* pin to output the divided-by-two oscillation clock.

1 (R/W): Divided-by-two clock output

0 (R/W): Oscillation clock output

For more information, refer to "CR Oscillation Frequency Monitoring Function."

Bit 7 CONEN

This bit disables the automatic CR oscillation stop function to enable continuous oscillation function.

1 (R/W): Enable continuous oscillation

0 (R/W): Disable continuous oscillation

For more information, refer to "CR Oscillation Frequency Monitoring Function."

Bit 6 EVTEN

This bit enables external clock input mode (event counter mode).

1 (R/W): External clock input mode

0 (R/W): Normal mode

For more information, refer to "Operating Modes."

Note: Do not input an external clock before the RFC_*n*CTL.EVTEN bit is set to 1. The RFIN*n* pin is pulled down to VSS 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 22.6.2 Oscillation Mode Selection

RFC_*n*CTL.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 RFC_*n*CTL.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
RFC_*n*TRG	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 RFC_*n*TRG.SSENB bit does not start oscillation when the RFC_*n*CTL.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 RFC_*n*CTL.MODEN bit = 0 (RFC operation disabled).
- When writing 1 to the RFC_*n*TRG.SREF bit, the RFC_*n*TRG.SSENA bit, or the RFC_*n*TRG.SSENB bit to start oscillation, be sure to avoid having more than one bit set to 1.
- Be sure to clear the interrupt flags (RFC_*n*INTF.EREFIF bit, RFC_*n*INTF.ESENAIF bit, RFC_*n*INTF.ESENBIF bit, RFC_*n*INTF.OVMCIF bit, and RFC_*n*INTF.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
RFC_*n*MCL	15–0	MC[15:0]	0x0000	H0	R/W	–
RFC_*n*MCH	15–8	–	0x00	–	R	–
	7–0	MC[23:16]	0x00	H0	R/W	

Or

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC_*n*MCL RFC_*n*MCH	31–24	–	0x00	–	R	–
	23–0	MC[23:0]	0x000000	H0	R/W	

Bits 31–24 Reserved

Bits 23–0 MC[23:0]

Measurement counter data can be read and written through these bits.

Note: The measurement counter must be set from the low-order value (RFC_*n*MCL.MC[15:0] bits) first when data is set using a 16-bit access instruction. The counter may not be set to the correct value if the high-order value (RFC_*n*MCH.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
RFC_*n*TCL	15–0	TC[15:0]	0x0000	H0	R/W	–
RFC_*n*TCH	15–8	–	0x00	–	R	–
	7–0	TC[23:16]	0x00	H0	R/W	

Or

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC_*n*TCL RFC_*n*TCH	31–24	–	0x00	–	R	–
	23–0	TC[23:0]	0x000000	H0	R/W	

Bits 31–24 Reserved

Bits 23–0 TC[23:0]

Time base counter data can be read and written through these bits.

Note: The time base counter must be set from the low-order value (RFC_*n*TCL.TC[15:0] bits) first when data is set using a 16-bit access instruction. The counter may not be set to the correct value if the high-order value (RFC_*n*TCH.TC[23:16] bits) is written first.

RFC Ch.*n* Interrupt Flag Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC_*n*INTF	15–8	–	0x00	–	R	–
	7–5	–	0x0	–	R	
	4	OVTCIF	0	H0	R/W	Cleared by writing 1.
	3	OVMCIF	0	H0	R/W	
	2	ESENBIF	0	H0	R/W	
	1	ESENAIF	0	H0	R/W	
	0	EREFIF	0	H0	R/W	

Bits 15–5 Reserved

Bit 4 OVTCIF
Bit 3 OVMCIF
Bit 2 ESENBIF
Bit 1 ESENAIF
Bit 0 EREFIF

These bits indicate the 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:

RFC_*n*INTF.OVTCIF bit: Time base counter overflow error interrupt
RFC_*n*INTF.OVMCIF bit: Measurement counter overflow error interrupt
RFC_*n*INTF.ESENBIF bit: Sensor B oscillation completion interrupt
RFC_*n*INTF.ESENAIF bit: Sensor A oscillation completion interrupt
RFC_*n*INTF.EREFIF bit: Reference oscillation completion interrupt

RFC Ch.*n* Interrupt Enable Register

Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
RFC_*n*INTE	15–8	–	0x00	–	R	–
	7–5	–	0x0	–	R	
	4	OVTCIE	0	H0	R/W	
	3	OVMCIE	0	H0	R/W	
	2	ESENBIE	0	H0	R/W	
	1	ESENAIE	0	H0	R/W	
	0	EREFIE	0	H0	R/W	

Bits 15–5 Reserved

Bit 4 OVTCIE
Bit 3 OVMCIE
Bit 2 ESENBIE
Bit 1 ESENAIE
Bit 0 EREFIE

These bits enable RFC interrupts.

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

The following shows the correspondence between the bit and interrupt:

RFC_*n*INTE.OVTCIE bit: Time base counter overflow error interrupt
RFC_*n*INTE.OVMCIE bit: Measurement counter overflow error interrupt
RFC_*n*INTE.ESENBIE bit: Sensor B oscillation completion interrupt
RFC_*n*INTE.ESENAIE bit: Sensor A oscillation completion interrupt
RFC_*n*INTE.EREFIE bit: Reference oscillation completion interrupt

23 Electrical Characteristics

23.1 Absolute Maximum Ratings

(V_{SS} = 0 V)

Item	Symbol	Condition	Rated value	Unit
Power supply voltage	V_{DD}		-0.3 to 7.0	V
LCD power supply voltage	V_{C1}		-0.3 to 7.0	V
	V_{C2}		-0.3 to 7.0	V
	V_{C3}		-0.3 to 7.0	V
Input voltage	V_I	#RESET, P07, P10–17, PD2–D3	-0.3 to V_{DD} + 0.5	V
		PD6–D7	-0.3 to V_{C3} + 0.5	V
		P00–06, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D1, PD5	-0.3 to 7.0	V
Output voltage	V_O	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D5	-0.3 to V_{DD} + 0.5	V
		PD6–D7	-0.3 to V_{C3} + 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	Ta		-40 to 105	°C
Storage temperature	Tstg		-65 to 125	°C

23.2 Recommended Operating Conditions

(V_{SS} = 0 V) *1

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Power supply voltage	V_{DD}	For normal operation	1.8	–	5.5	V
		For Flash programming	2.2	–	5.5	V
		For LCD driver operation	1.8	–	5.5	V
	V_{C3}	When PD6 and PD7 are used as GPIO	1.8	–	5.5	V
LCD power supply voltage (1/3 bias)	V_{C1}	When an external voltage is applied	–	1	1.9	V
	V_{C2}	$V_{C1} \leq V_{C2} \leq V_{C3}$	–	2	3.8	V
	V_{C3}		–	3	5.5	V
LCD power supply voltage (1/2 bias)	V_{C1}	When an external voltage is applied	–	1.5	2.8	V
	V_{C3}	V_{C1} (= V_{C2}) $\leq V_{C3}$	–	3	5.5	V
OSC1 oscillator oscillation frequency	f_{OSC1}	Crystal oscillator	–	32.768	–	kHz
OSC3 oscillator oscillation frequency	f_{OSC3}	Crystal/ceramic oscillator	0.2	–	33	MHz
EXOSC external clock frequency	f_{EXOSC}	When supplied from an external oscillator	0.016	–	33	MHz
Bypass capacitor between V_{SS} and V_{DD}	C_{PW1}		–	3.3	–	µF
Capacitor between V_{SS} and V_{D1}	C_{PW2}		–	1	1.2	µF
Capacitor between V_{SS} and V_{C1}	C_{LCD1}	*2, *3	–	1	–	µF
Capacitors between V_{SS} and V_{C2-3}	C_{LCD2-3}	*2, *3	–	1	–	µF
Capacitor between C_{P1} and C_{P2}	C_{LCD4}	*2, *3	–	1	–	µF
Gate capacitor for OSC1 oscillator	C_{G1}	*4	0	–	25	pF
Drain capacitor for OSC1 oscillator	C_{D1}	*4	–	0	–	pF
Gate capacitor for OSC3 oscillator	C_{G3}	When crystal/ceramic oscillator is used *4	0	–	100	pF
Drain capacitor for OSC3 oscillator	C_{D3}	When crystal/ceramic oscillator is used *4	0	–	100	pF
Debug pin pull-up resistors	R_{DBG1-2}	*5	–	100	–	kΩ
Capacitor between V_{SS} and V_{PP}	C_{VPP}		–	0.1	–	µF

*1 The potential variation of the V_{SS} voltage should be suppressed to within ±0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count).
*2 The V_{C1}–V_{C3} and C_{P1}–C_{P2} pins can be left open when the LCD driver is not used.
*3 Connect between the V_{C1} and V_{C2} pins when the LCD power supply circuit is configured for 1/2 bias.
*4 The component values should be determined after performing matching evaluation of the resonator mounted on the printed circuit board actually used.
*5 R_{DBG1-2} are not required when using the debug pins as general-purpose I/O ports.
*6 The component values should be determined after evaluating operations using an actual mounting board.

23.3 Current Consumption

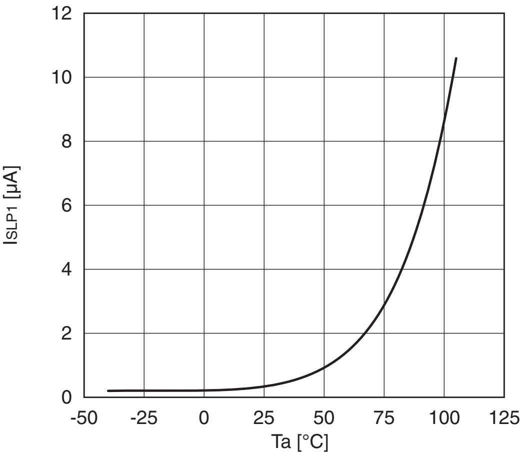
Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = 25°C, EXOSC = OFF, PWGACTL.REGMODE[1:0] bits = 0x0 (automatic mode), PWGACTL.REGSEL bit = 1 (mode0), FLASHCWAIT.RDWAIT[1:0] bits = 0x1 (2 cycles)

Item	Symbol	Condition	Ta	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I_{SLP1}	IOSC = OFF, OSC1 = OFF, OSC3 = OFF	25°C	–	0.3	10.0	μA
			105°C	–	9.5	190	μA
	I_{SLP2}	IOSC = OFF, OSC1 = OFF, OSC3 = OFF, PWGACTL.REGSEL bit = 0 (mode1)	25°C	–	0.25	7.5	μA
			105°C	–	8.2	130	μA
	I_{SLP3}	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = OFF, RTCA = ON	–	–	0.8	13.0	μA
	I_{SLP4}	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = OFF, RTCA = ON, PWGACTL.REGSEL bit = 0 (mode1)	–	–	0.7	8.5	μA
Current consumption in HALT mode	I_{HALT1}	IOSC = 32 MHz, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = IOSC	–	–	1,450	2,230	μA
	I_{HALT2}	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = 20 MHz (ceramic oscillator)*3, SYSCLK = OSC3	–	–	815	1,620	μA
	I_{HALT3}	IOSC = OFF, OSC1 = 32 kHz*2, OSC3 = OFF, SYSCLK = OSC1	–	–	2.0	15.0	μA
	I_{HALT4}	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = OSC1	–	–	1.5	14.0	μA
	I_{HALT5}	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = OSC1, PWGACTL.REGSEL bit = 0 (mode1)	–	–	1.0	9.5	μA
Current consumption in RUN mode	I_{RUN1}*4	IOSC = 32 MHz, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = IOSC	–	–	5,360	8,250	μA
	I_{RUN2}*4	IOSC = 16 MHz, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = IOSC	–	–	3,100	4,800	μA
	I_{RUN3}*4	IOSC = 8 MHz, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = IOSC	–	–	1,600	2,500	μA
	I_{RUN4}*4	IOSC = 2 MHz, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = IOSC, PWGACTL.REGSEL bit = 0 (mode1)	–	–	260	400	μA
	I_{RUN5}*4	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = 20 MHz (ceramic oscillator)*3, SYSCLK = OSC3	–	–	3,720	5,950	μA
	I_{RUN6}*4	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = OSC1, FLASHCWAIT.RDWAIT[1:0] bits = 0x0 (1 cycle)	–	–	7.5	21.0	μA
	I_{RUN7}*4	IOSC = OFF, OSC1 = 32.768 kHz*1, OSC3 = OFF, SYSCLK = OSC1, FLASHCWAIT.RDWAIT[1:0] bits = 0x0 (1 cycle), PWGACTL.REGSEL bit = 0 (mode1)	–	–	5.5	17.0	μA
	I_{RUN8}*4	IOSC = OFF, OSC1 = 32 kHz*2, OSC3 = OFF, SYSCLK = OSC1, FLASHCWAIT.RDWAIT[1:0] bits = 0x0 (1 cycle)	–	–	8.0	22.0	μA
	I_{RUN9}*4	IOSC = OFF, OSC1 = 32 kHz*2, OSC3 = OFF, SYSCLK = OSC1, FLASHCWAIT.RDWAIT[1:0] bits = 0x0 (1 cycle), PWGACTL.REGSEL bit = 0 (mode1)	–	–	6.0	18.0	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, R_1 = 50 kΩ (Max.), C_L = 7 pF)
*2 OSC1 oscillator: CLGOSC1.OSC1SELCR bit = 1
*3 OSC3 oscillator: CLGOSC3.OSC3INV[1:0] bits = 0x2, C_{G3} = C_{D3} = 10 pF
*4 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.

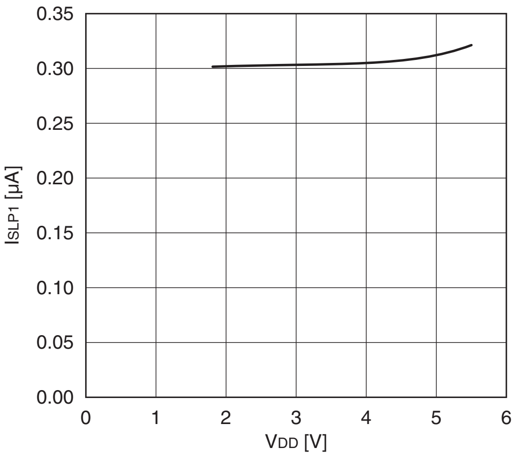
Current consumption-temperature characteristic in SLEEP mode

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



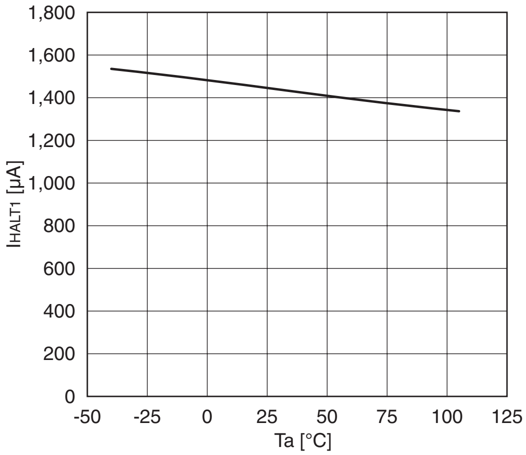
Current consumption-power supply voltage characteristic in SLEEP mode

IOSC = OFF, OSC1 = OFF, OSC3 = OFF, Ta = 25°C, Typ. value



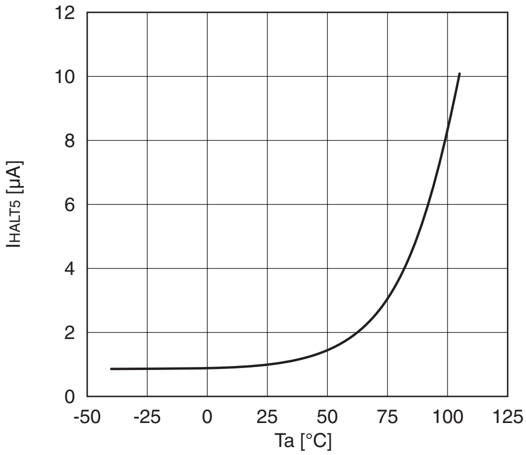
Current consumption-temperature characteristic in HALT mode (IOSC operation)

IOSC = 32 MHz, OSC1 = 32.768 kHz, OSC3 = OFF, Typ. value



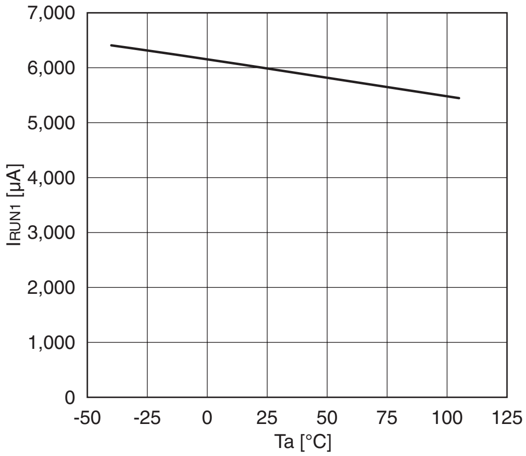
Current consumption-temperature characteristic in HALT mode (OSC1 operation)

IOSC = OFF, OSC1 = 32.768 kHz, OSC3 = OFF, PWGACTL.REGSEL bit = 0 (mode1), Typ. value



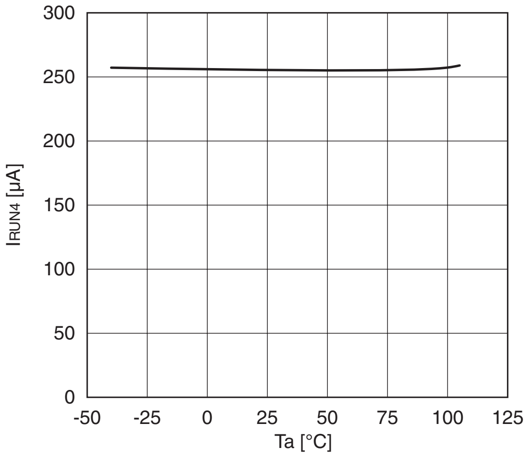
Current consumption-temperature characteristic in RUN mode (IOSC operation)

IOSC = 32 MHz, OSC1 = 32.768 kHz, OSC3 = OFF, Typ. value



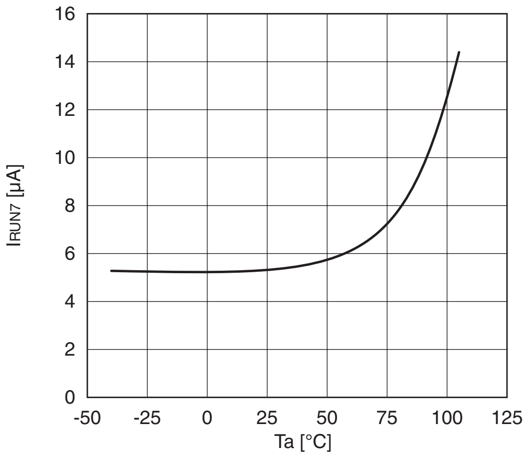
Current consumption-temperature characteristic in RUN mode (IOSC operation)

IOSC = 2 MHz, OSC1 = 32.768 kHz, OSC3 = OFF PWGACTL.REGSEL bit = 0 (mode1), Typ. value



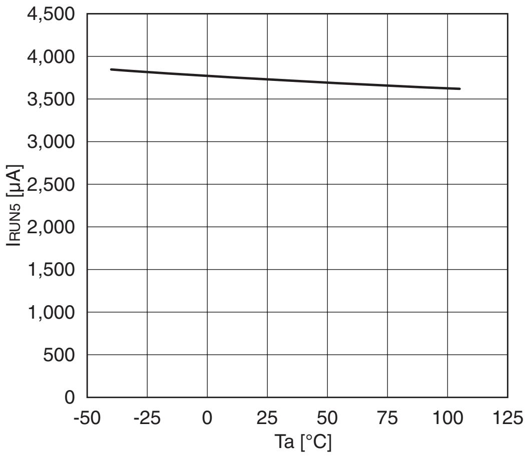
Current consumption-temperature characteristic in RUN mode (OSC1 operation)

IOSC = OFF, OSC1 = 32.768 kHz, OSC3 = OFF, PWGACTL.REGSEL bit = 0 (mode1), Typ. value



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

IOSC = OFF, OSC1 = 32.768 kHz, OSC3 = 20 MHz (ceramic oscillator), CLGOSC3.OSC3INV[1:0] bits = 0x2, Typ. value

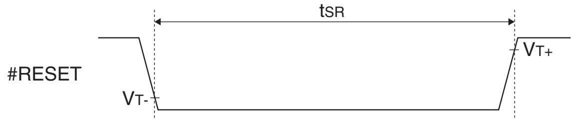


23.4 System Reset Controller (SRC) Characteristics

#RESET pin characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

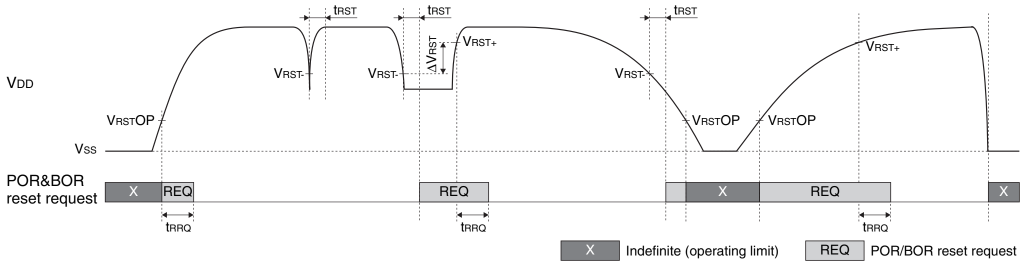
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}		0.5 × V_{DD}	–	0.8 × V_{DD}	V
Low level Schmitt input threshold voltage	V_{T-}		0.2 × V_{DD}	–	0.5 × V_{DD}	V
Schmitt input hysteresis voltage	ΔV_T		230	–	–	mV
Input pull-up resistance	R_{IN}		100	200	500	kΩ
Leakage current	I_{LEAK}	SRCRESETPCTL.PORT_PLUP_EN bit = 0	-1	–	1	µA
Pin capacitance	C_{IN}		–	–	15	pF
Reset Low pulse width	t_{SR}		20	–	–	µs



POR/BOR characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
POR/BOR canceling voltage	V_{RST+}		1.10	–	1.75	V
POR/BOR detection voltage	V_{RST-}		1.00	–	1.70	V
POR/BOR hysteresis voltage	ΔV_{RST}		20	80	–	mV
POR/BOR detection response time	t_{RST}		–	–	1,000	µs
POR/BOR operating limit voltage	$V_{RST}OP$		–	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 $V_{RST}OP$ or less.

Reset hold circuit characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Reset hold time*1	t_{RSTR}		0.9	1	2	ms

*1 Time until the internal reset signal is negated after the reset request is canceled.

23.5 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.

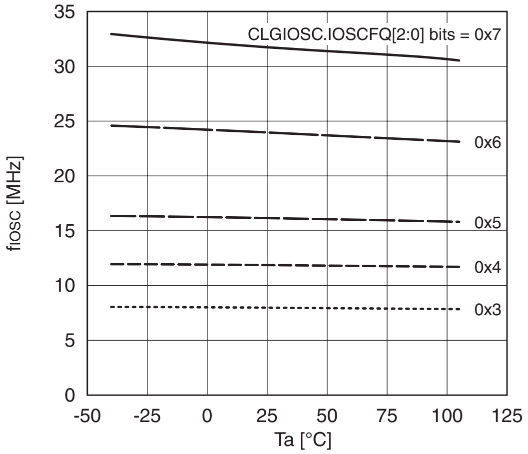
IOSC oscillator circuit characteristics

Unless otherwise specified: V_{DD} = 2.0 to 5.5 V, V_{SS} = 0 V, PWGACTL.REGSEL bit = 1, Ta = -40 to 105°C

Item	Symbol	Condition	Ta	Min.	Typ.	Max.	Unit
Oscillation start time	t_{stal}		–	–	–	3	µs
Oscillation frequency	f_{IOSC}	CLGIOSC.IOSCFQ[2:0] bits = 0x7	25°C	31.0	32	33.0	MHz
			-40 to 105°C	29.1	32	33.3	MHz
		CLGIOSC.IOSCFQ[2:0] bits = 0x6	25°C	23.3	24	24.7	MHz
			-40 to 105°C	21.8	24	26.2	MHz
		CLGIOSC.IOSCFQ[2:0] bits = 0x5	25°C	15.5	16	16.5	MHz
			-40 to 105°C	14.9	16	17.1	MHz
		CLGIOSC.IOSCFQ[2:0] bits = 0x4	25°C	11.6	12	12.4	MHz
			-40 to 105°C	11.3	12	12.7	MHz
		CLGIOSC.IOSCFQ[2:0] bits = 0x3	25°C	7.8	8	8.2	MHz
			-40 to 105°C	7.6	8	8.4	MHz
		CLGIOSC.IOSCFQ[2:0] bits = 0x1	25°C	2.11	2.2	2.29	MHz
			-40 to 105°C	2.00	2.2	2.40	MHz
		CLGIOSC.IOSCFQ[2:0] bits = 0x0	25°C	0.97	1.1	1.23	MHz
			-40 to 105°C	0.92	1.1	1.28	MHz
		CLGIOSC.IOSCFQ[2:0] bits = 0x1, PWGACTL.REGSEL bit = 0	25°C	1.94	2	2.06	MHz
			-40 to 105°C	1.84	2	2.16	MHz
		CLGIOSC.IOSCFQ[2:0] bits = 0x0, PWGACTL.REGSEL bit = 0	25°C	0.88	1	1.12	MHz
			-40 to 105°C	0.84	1	1.16	MHz

IOSC oscillation frequency-temperature characteristic

V_{DD} = 2.0 to 5.5 V, PWGACTL.REGSEL bit = 1, Typ. value



OSC1 oscillator circuit characteristics

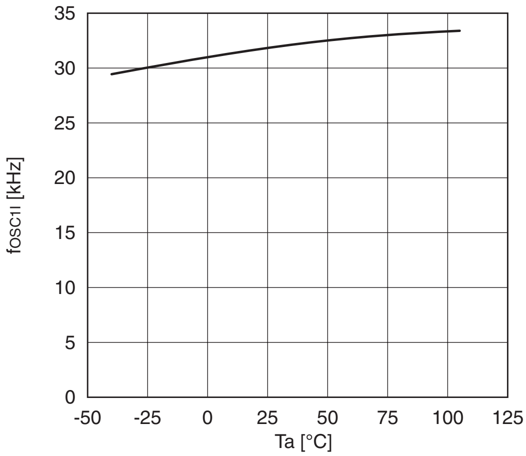
Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = 25°C

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	–	–	700	µs
Internal oscillator oscillation frequency	f_{OSC1I}	CLGOSC1.OSC1SELCR bit = 1	31.04	32	32.96	kHz
		CLGOSC1.OSC1SELCR bit = 1, Ta = -40 to 105°C	27.84	32	36.16	kHz

*1 CLGOSC1.CGI1[2:0] bits = 0x0, Crystal resonator = C-002RX (manufactured by Seiko Epson Corporation, R_1 = 50 kΩ (Max.), C_L = 7 pF)

OSC1 internal oscillation frequency-temperature characteristic

V_{DD} = 1.8 to 5.5 V, Typ. value



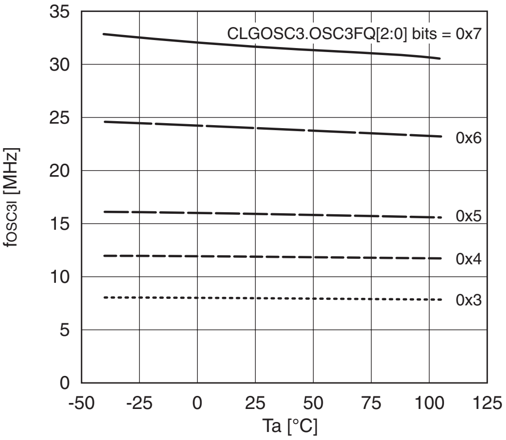
OSC3 oscillator circuit characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, CLGOSC3.OSC3MD bit = 0, PWGACTL.REGSEL bit = 1, Ta = 25°C

Item	Symbol	Condition	Ta	Min.	Typ.	Max.	Unit
Crystal/ceramic oscillation start time	t_{sta3C}	Crystal resonator, CLGOSC3.OSC3MD bit = 1	–	–	–	20	ms
		Ceramic resonator, CLGOSC3.OSC3MD bit = 1	–	–	–	1	ms
Crystal/ceramic oscillator internal gate capacitance	C_{GI3C}	CLGOSC3.OSC3MD bit = 1	–	–	5	–	pF
Crystal/ceramic oscillator internal drain capacitance	C_{DI3C}	CLGOSC3.OSC3MD bit = 1	–	–	5	–	pF
Internal oscillator oscillation start time	t_{sta3I}		–	–	–	3	µs
Internal oscillator oscillation frequency	f_{OSC3I}	CLGOSC3.OSC3FQ[2:0] bits = 0x7, V_{DD} = 2.0 to 5.5 V	25°C	31.0	32	33.0	MHz
			-40 to 105°C	29.1	32	33.3	MHz
		CLGOSC3.OSC3FQ[2:0] bits = 0x6, V_{DD} = 2.0 to 5.5 V	25°C	23.3	24	24.7	MHz
			-40 to 105°C	21.8	24	26.2	MHz
		CLGOSC3.OSC3FQ[2:0] bits = 0x5, V_{DD} = 2.0 to 5.5 V	25°C	15.5	16	16.5	MHz
			-40 to 105°C	14.9	16	17.1	MHz
		CLGOSC3.OSC3FQ[2:0] bits = 0x4, V_{DD} = 2.0 to 5.5 V	25°C	11.6	12	12.4	MHz
			-40 to 105°C	11.3	12	12.7	MHz
		CLGOSC3.OSC3FQ[2:0] bits = 0x3–0x0, V_{DD} = 2.0 to 5.5 V	25°C	7.8	8	8.2	MHz
			-40 to 105°C	7.6	8	8.4	MHz

OSC3 internal oscillation frequency-temperature characteristic

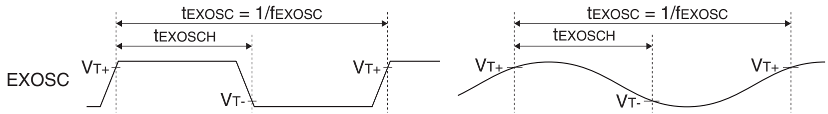
V_{DD} = 2.0 to 5.5 V, Typ. value



EXOSC external clock input characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°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 × V_{DD}	–	0.8 × V_{DD}	V
Low level Schmitt input threshold voltage	V_{T-}		0.2 × V_{DD}	–	0.5 × V_{DD}	V
Schmitt input hysteresis voltage	ΔV_T		165	–	–	mV



23.6 Flash Memory Characteristics

Unless otherwise specified: V_{DD} = 2.2 to 5.5 V, V_{SS} = 0 V*1, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Programming count *2	C_{FEP}	Programmed data is guaranteed to be retained for 10 years.	1,000	–	–	times
Programming current *3	I_{FLASH}		–	16	33	mA

*1 The potential variation of the V_{SS} voltage should be suppressed to within ±0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count).

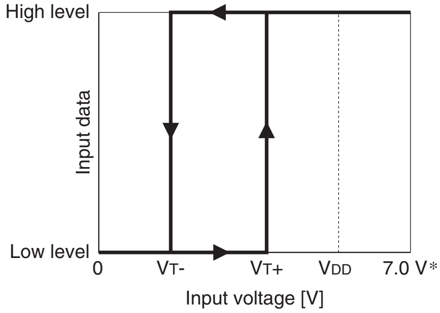
*2 Assumed that Erasing + Programming as count of 1. The count includes programming in the factory for shipment with ROM data programmed.

*3 The value is added to the current consumption in the current operating mode.

23.7 Input/Output Port (PPORT) Characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

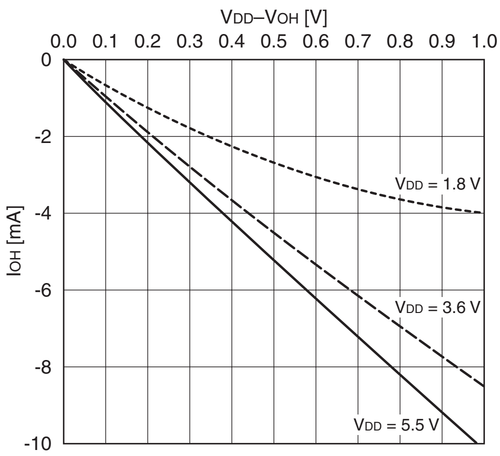
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D3, PD5	0.5 × V_{DD}	–	0.8 × V_{DD}	V
		PD6–D7	0.5 × V_{C3}	–	0.8 × V_{C3}	V
Low level Schmitt input threshold voltage	V_{T-}	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D3, PD5	0.2 × V_{DD}	–	0.5 × V_{DD}	V
		PD6–D7	0.2 × V_{C3}	–	0.5 × V_{C3}	V
Schmitt input hysteresis voltage	ΔV_T		165	–	–	mV
High level output current	I_{OH}	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D5, V_{OH} = 0.9 × V_{DD}	–	–	-0.5	mA
		PD6–D7, V_{OH} = 0.9 × V_{C3}	–	–	-0.5	mA
Low level output current	I_{OL}	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D5, V_{OL} = 0.1 × V_{DD}	0.5	–	–	mA
		PD6–D7, V_{OL} = 0.1 × V_{C3}	0.5	–	–	mA
Leakage current	I_{LEAK}	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D3, PD5–D7	-1	–	1	μA
Input pull-up resistance	R_{INU}	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D3, PD5–D7	100	200	500	kΩ
Input pull-down resistance	R_{IND}	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D3, PD5–D7	100	200	500	kΩ
Pin capacitance	C_{IN}	P00–07, P10–17, P20–27, P30–37, P40–47, P50–57, P60–67, PD0–D3, PD6–D7	–	–	15	pF
		PD5	–	–	20	pF



(* For over voltage tolerant fail-safe type port)

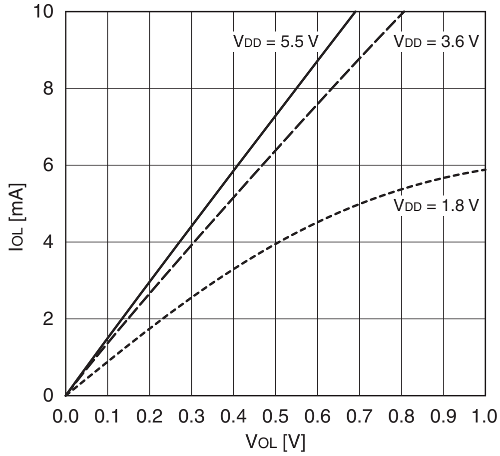
High-level output current characteristic

Ta = 105°C, Max. value



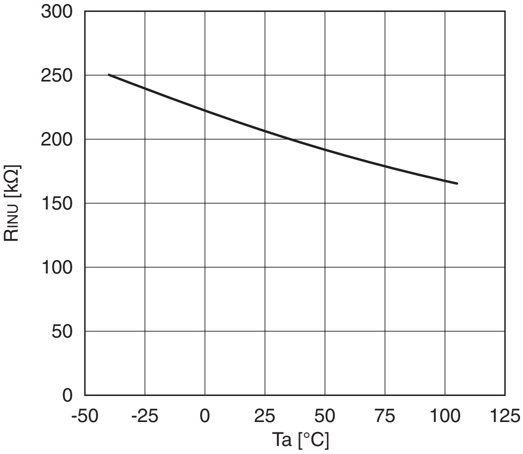
Low-level output current characteristic

Ta = 105°C, Min. value



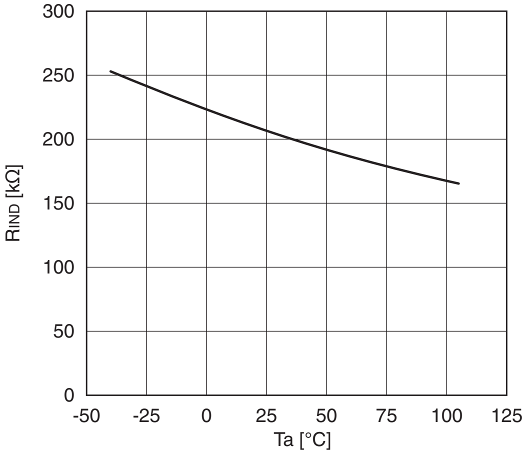
Pull-up resistance-temperature characteristic

V_{DD} = 1.8 to 5.5 V, Typ. value



Pull-down resistance-temperature characteristic

V_{DD} = 1.8 to 5.5 V, Typ. value



23.8 Supply Voltage Detector (SVD4) Characteristics

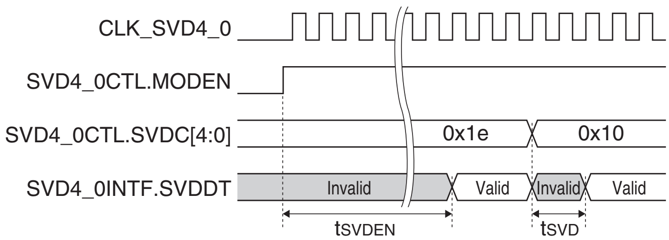
Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EXSVD0*x* pin input voltage range	V_{EXSVD}		0	–	5.5	V
EXSVD0*x* input impedance	R_{EXSVD}	SVD4_0CTL.SVDC[4:0] bits = 0x00	366	407	448	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x01	388	431	474	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x02	409	455	500	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x03	431	479	527	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x04	452	503	553	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x05	474	527	579	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x06	495	550	606	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x07	517	574	632	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x08	539	598	658	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x09	560	622	685	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x0a	582	646	711	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x0b	603	670	737	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x0c	625	694	763	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x0d	646	718	790	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x0e	668	742	816	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x0f	689	766	842	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x10	711	790	869	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x11	754	838	921	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x12	775	862	948	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x13	797	886	974	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x14	819	909	1,000	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x15	840	933	1,027	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x16	862	957	1,053	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x17	883	981	1,079	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x18	905	1,005	1,106	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x19	926	1,029	1,132	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x1a	969	1,077	1,185	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x1b	991	1,101	1,211	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x1c	1,012	1,125	1,237	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x1d	1,034	1,149	1,264	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x1e	1,055	1,173	1,290	kΩ
		SVD4_0CTL.SVDC[4:0] bits = 0x1f	1,077	1,197	1,316	kΩ

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EXSVD0*x* detection voltage/ SVD detection voltage (Ta = -40 to 85°C)	V_{SVD_EXT}/ V_{SVD}	SVD4_0CTL.SVDC[4:0] bits = 0x00	1.65	1.7	1.75	V
		SVD4_0CTL.SVDC[4:0] bits = 0x01	1.75	1.8	1.85	V
		SVD4_0CTL.SVDC[4:0] bits = 0x02	1.84	1.9	1.96	V
		SVD4_0CTL.SVDC[4:0] bits = 0x03	1.94	2.0	2.06	V
		SVD4_0CTL.SVDC[4:0] bits = 0x04	2.04	2.1	2.16	V
		SVD4_0CTL.SVDC[4:0] bits = 0x05	2.13	2.2	2.27	V
		SVD4_0CTL.SVDC[4:0] bits = 0x06	2.23	2.3	2.37	V
		SVD4_0CTL.SVDC[4:0] bits = 0x07	2.33	2.4	2.47	V
		SVD4_0CTL.SVDC[4:0] bits = 0x08	2.43	2.5	2.58	V
		SVD4_0CTL.SVDC[4:0] bits = 0x09	2.52	2.6	2.68	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0a	2.62	2.7	2.78	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0b	2.72	2.8	2.88	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0c	2.81	2.9	2.99	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0d	2.91	3.0	3.09	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0e	3.01	3.1	3.19	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0f	3.10	3.2	3.30	V
		SVD4_0CTL.SVDC[4:0] bits = 0x10	3.20	3.3	3.40	V
		SVD4_0CTL.SVDC[4:0] bits = 0x11	3.40	3.5	3.61	V
		SVD4_0CTL.SVDC[4:0] bits = 0x12	3.49	3.6	3.71	V
		SVD4_0CTL.SVDC[4:0] bits = 0x13	3.59	3.7	3.81	V
		SVD4_0CTL.SVDC[4:0] bits = 0x14	3.69	3.8	3.91	V
		SVD4_0CTL.SVDC[4:0] bits = 0x15	3.78	3.9	4.02	V
		SVD4_0CTL.SVDC[4:0] bits = 0x16	3.88	4.0	4.12	V
		SVD4_0CTL.SVDC[4:0] bits = 0x17	3.98	4.1	4.22	V
		SVD4_0CTL.SVDC[4:0] bits = 0x18	4.07	4.2	4.33	V
		SVD4_0CTL.SVDC[4:0] bits = 0x19	4.17	4.3	4.43	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1a	4.37	4.5	4.64	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1b	4.46	4.6	4.74	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1c	4.56	4.7	4.84	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1d	4.66	4.8	4.94	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1e	4.75	4.9	5.05	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1f	4.85	5.0	5.15	V
EXSVD0*x* detection voltage/ SVD detection voltage (Ta = -40 to 105°C)	V_{SVD_EXT}/ V_{SVD}	SVD4_0CTL.SVDC[4:0] bits = 0x00	1.63	1.7	1.77	V
		SVD4_0CTL.SVDC[4:0] bits = 0x01	1.73	1.8	1.87	V
		SVD4_0CTL.SVDC[4:0] bits = 0x02	1.82	1.9	1.98	V
		SVD4_0CTL.SVDC[4:0] bits = 0x03	1.92	2.0	2.08	V
		SVD4_0CTL.SVDC[4:0] bits = 0x04	2.02	2.1	2.18	V
		SVD4_0CTL.SVDC[4:0] bits = 0x05	2.11	2.2	2.29	V
		SVD4_0CTL.SVDC[4:0] bits = 0x06	2.21	2.3	2.39	V
		SVD4_0CTL.SVDC[4:0] bits = 0x07	2.30	2.4	2.50	V
		SVD4_0CTL.SVDC[4:0] bits = 0x08	2.40	2.5	2.60	V
		SVD4_0CTL.SVDC[4:0] bits = 0x09	2.50	2.6	2.70	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0a	2.59	2.7	2.81	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0b	2.69	2.8	2.91	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0c	2.78	2.9	3.02	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0d	2.88	3.0	3.12	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0e	2.98	3.1	3.22	V
		SVD4_0CTL.SVDC[4:0] bits = 0x0f	3.07	3.2	3.33	V
		SVD4_0CTL.SVDC[4:0] bits = 0x10	3.17	3.3	3.43	V
		SVD4_0CTL.SVDC[4:0] bits = 0x11	3.36	3.5	3.64	V
		SVD4_0CTL.SVDC[4:0] bits = 0x12	3.46	3.6	3.74	V
		SVD4_0CTL.SVDC[4:0] bits = 0x13	3.55	3.7	3.85	V
		SVD4_0CTL.SVDC[4:0] bits = 0x14	3.65	3.8	3.95	V
		SVD4_0CTL.SVDC[4:0] bits = 0x15	3.74	3.9	4.06	V
		SVD4_0CTL.SVDC[4:0] bits = 0x16	3.84	4.0	4.16	V
		SVD4_0CTL.SVDC[4:0] bits = 0x17	3.94	4.1	4.26	V
		SVD4_0CTL.SVDC[4:0] bits = 0x18	4.03	4.2	4.37	V
		SVD4_0CTL.SVDC[4:0] bits = 0x19	4.13	4.3	4.47	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1a	4.32	4.5	4.68	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1b	4.42	4.6	4.78	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1c	4.51	4.7	4.89	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1d	4.61	4.8	4.99	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1e	4.70	4.9	5.10	V
		SVD4_0CTL.SVDC[4:0] bits = 0x1f	4.80	5.0	5.20	V
SVD circuit enable response time	t_{SVDEN}	*1	–	–	500	µs
SVD circuit response time	t_{SVD}		–	–	60	µs

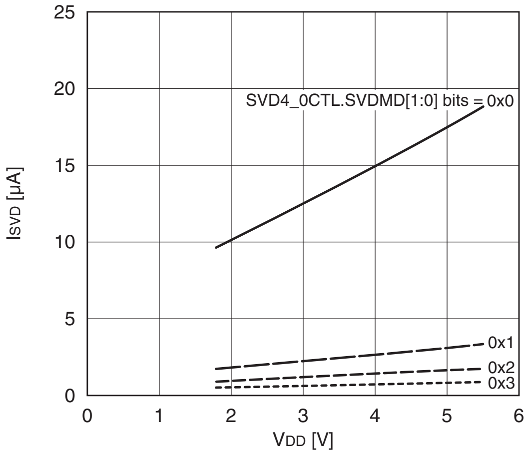
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD circuit current	I_{SVD}	SVD4_0CTL.SVDMD[1:0] bits = 0x0, SVD4_0CTL.SVDC[4:0] bits = 0x00, CLK_SVD4_0 = 32 kHz, Ta = 25°C	–	19	35	µA
		SVDCTL.SVDMD[1:0] bits = 0x1, SVD4_0CTL.SVDC[4:0] bits = 0x00, CLK_SVD4_0 = 32 kHz, Ta = 25°C	–	3.5	7.7	µA
		SVD4_0CTL.SVDMD[1:0] bits = 0x2, SVD4_0CTL.SVDC[4:0] bits = 0x00, CLK_SVD4_0 = 32 kHz, Ta = 25°C	–	1.8	4.1	µA
		SVD4_0CTL.SVDMD[1:0] bits = 0x3, SVD4_0CTL.SVDC[4:0] bits = 0x00, CLK_SVD4_0 = 32 kHz, Ta = 25°C	–	1	2.4	µA

*1 If CLK_SVD4_0 is configured in the neighborhood of 32 kHz, the SVD4_0INTF.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, SVD4_0CTL.SVDC[4:0] bits = 0x00, CLK_SVD4_0 = 32 kHz, Typ. value



23.9 UART (UART3) Characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Transfer baud rate	U_{BRT1}	Normal mode	150	–	460,800	bps
	U_{BRT2}	IrDA mode	150	–	115,200	bps

23.10 Synchronous Serial Interface (SPIA) Characteristics

SPIA Ch.0 master mode

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

Item	Symbol	Condition	V_{D1} output	Min.	Typ.	Max.	単位
SPICLK0 cycle time	t_{SCYC}		mode0	200	–	–	ns
			mode1	480	–	–	ns
SPICLK0 High pulse width	t_{SCKH}		mode0	80	–	–	ns
			mode1	190	–	–	ns
SPICLK0 Low pulse width	t_{SCKL}		mode0	80	–	–	ns
			mode1	190	–	–	ns
SDI0 setup time	t_{SDS}		mode0	70	–	–	ns
			mode1	180	–	–	ns
SDI0 hold time	t_{SDH}		mode0	10	–	–	ns
			mode1	40	–	–	ns
SDO0 output delay time	t_{SDO}	C_L = 15 pF *1	mode0	–	–	25	ns
			mode1	–	–	80	ns

*1 C_L = Pin load

SPIA Ch.0 slave mode

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

Item	Symbol	Condition	V_{D1} output	Min.	Typ.	Max.	単位
SPICLK0 cycle time	t_{SCYC}		mode0	200	–	–	ns
			mode1	480	–	–	ns
SPICLK0 High pulse width	t_{SCKH}		mode0	80	–	–	ns
			mode1	190	–	–	ns
SPICLK0 Low pulse width	t_{SCKL}		mode0	80	–	–	ns
			mode1	190	–	–	ns
SDI0 setup time	t_{SDS}		mode0	10	–	–	ns
			mode1	40	–	–	ns
SDI0 hold time	t_{SDH}		mode0	15	–	–	ns
			mode1	50	–	–	ns
#SPISS0 setup time	t_{SSS}		mode0	10	–	–	ns
			mode1	40	–	–	ns
#SPISS0 High pulse width	t_{SSH}		mode0	80	–	–	ns
			mode1	190	–	–	ns
SDO0 output delay time	t_{SDO}	C_L = 15 pF *1	mode0	–	–	65	ns
			mode1	–	–	210	ns
SDO0 output start time	t_{SDD}	C_L = 15 pF *1	mode0	–	–	65	ns
			mode1	–	–	210	ns
SDO0 output stop time	t_{SDZ}	C_L = 15 pF *1	mode0	–	–	65	ns
			mode1	–	–	210	ns

*1 C_L = Pin load

SPIA Ch.1 master mode

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

Item	Symbol	Condition	V_{DD}	V_{D1} output	Min.	Typ.	Max.	単位
SPICLK1 cycle time	t_{SCYC}		3.0 to 5.5 V	mode0	100	–	–	ns
			1.8 to 3.0 V	mode0	120	–	–	ns
			–	mode1	480	–	–	ns
SPICLK1 High pulse width	t_{SCKH}		3.0 to 5.5 V	mode0	40	–	–	ns
			1.8 to 3.0 V	mode0	50	–	–	ns
			–	mode1	190	–	–	ns
SPICLK1 Low pulse width	t_{SCKL}		3.0 to 5.5 V	mode0	40	–	–	ns
			1.8 to 3.0 V	mode0	50	–	–	ns
			–	mode1	190	–	–	ns
SDI1 setup time	t_{SDS}		3.0 to 5.5 V	mode0	30	–	–	ns
			1.8 to 3.0 V	mode0	40	–	–	ns
			–	mode1	120	–	–	ns
SDI1 hold time	t_{SDH}		–	mode0	10	–	–	ns
			–	mode1	40	–	–	ns
SDO1 output delay time	t_{SDO}	C_L = 15 pF *1	–	mode0	–	–	20	ns
			–	mode1	–	–	80	ns

*1 C_L = Pin load

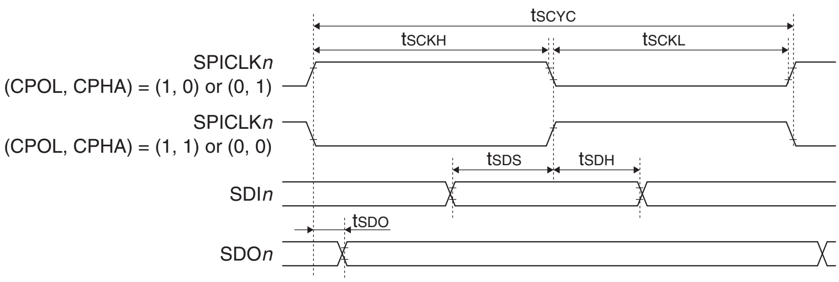
SPIA Ch.1 slave mode

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

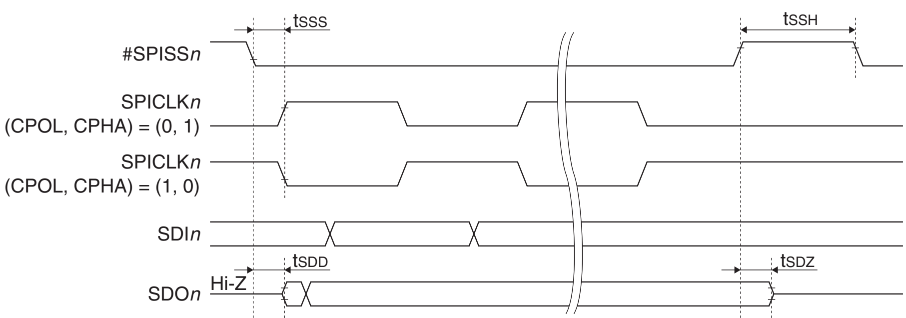
Item	Symbol	Condition	V_{DD}	V_{D1} output	Min.	Typ.	Max.	単位
SPICLK1 cycle time	t_{SCYC}		3.0 to 5.5 V	mode0	120	–	–	ns
			1.8 to 3.0 V	mode0	150	–	–	ns
			–	mode1	480	–	–	ns
SPICLK1 High pulse width	t_{SCKH}		3.0 to 5.5 V	mode0	50	–	–	ns
			1.8 to 3.0 V	mode0	60	–	–	ns
			–	mode1	190	–	–	ns
SPICLK1 Low pulse width	t_{SCKL}		3.0 to 5.5 V	mode0	50	–	–	ns
			1.8 to 3.0 V	mode0	60	–	–	ns
			–	mode1	190	–	–	ns
SDI1 setup time	t_{SDS}		–	mode0	10	–	–	ns
			–	mode1	40	–	–	ns
SDI1 hold time	t_{SDH}		–	mode0	10	–	–	ns
			–	mode1	50	–	–	ns
#SPISS1 setup time	t_{SSS}		–	mode0	10	–	–	ns
			–	mode1	40	–	–	ns
#SPISS1 High pulse width	t_{SSH}		3.0 to 5.5 V	mode0	50	–	–	ns
			1.8 to 3.0 V	mode0	60	–	–	ns
			–	mode1	190	–	–	ns
SDO1 output delay time	t_{SDO}	C_L = 15 pF *1	3.0 to 5.5 V	mode0	–	–	42	ns
			1.8 to 3.0 V	mode0	–	–	52	ns
			–	mode1	–	–	180	ns
SDO1 output start time	t_{SDD}	C_L = 15 pF *1	3.0 to 5.5 V	mode0	–	–	42	ns
			1.8 to 3.0 V	mode0	–	–	52	ns
			–	mode1	–	–	180	ns
SDO1 output stop time	t_{SDZ}	C_L = 15 pF *1	3.0 to 5.5 V	mode0	–	–	42	ns
			1.8 to 3.0 V	mode0	–	–	52	ns
			–	mode1	–	–	180	ns

*1 C_L = Pin load

Master and slave modes



Slave mode

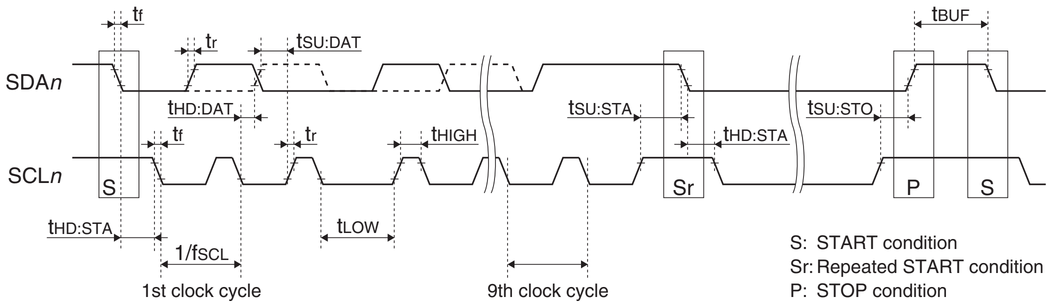


23.11 I²C (I2C) Characteristics

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°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.



23.12 LCD Driver (LCD8D) Characteristics

The LCD driver characteristics varies depending on the panel load (panel size, drive duty, number of display pixels and display contents), so evaluate them by connecting to the actually used LCD panel.

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = 25°C, LCD8DTIM2.BSTC[1:0] bits = 0x1 (Voltage booster clock = 2 kHz), No panel load

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
LCD drive voltage (1/3 bias, V_{C1} reference voltage) V_{DD} = 1.8 to 5.5 V *3 LCD8DPWR.BIASSEL bit = 1 LCD8DPWR.VCSEL bit = 0	V_{C1}	Connect 1 MΩ load resistor between V_{SS} and V_{C1}		0.312 × V_{C3} (Typ.)	–	0.355 × V_{C3} (Typ.)	V
	V_{C2}	Connect 1 MΩ load resistor between V_{SS} and V_{C2}		0.623 × V_{C3} (Typ.)	–	0.710 × V_{C3} (Typ.)	V
	V_{C3}	Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD8DPWR.LC[4:0] bits = 0x00	2.44	2.57	2.70	V
			LCD8DPWR.LC[4:0] bits = 0x01	2.51	2.65	2.79	V
			LCD8DPWR.LC[4:0] bits = 0x02	2.58	2.73	2.87	V
			LCD8DPWR.LC[4:0] bits = 0x03	2.66	2.80	2.95	V
			LCD8DPWR.LC[4:0] bits = 0x04	2.73	2.88	3.03	V
			LCD8DPWR.LC[4:0] bits = 0x05	2.82	2.96	3.10	V
			LCD8DPWR.LC[4:0] bits = 0x06	2.90	3.04	3.18	V
			LCD8DPWR.LC[4:0] bits = 0x07	2.97	3.12	3.26	V
			LCD8DPWR.LC[4:0] bits = 0x08	3.05	3.20	3.35	V
			LCD8DPWR.LC[4:0] bits = 0x09	3.14	3.28	3.41	V
			LCD8DPWR.LC[4:0] bits = 0x0a	3.22	3.35	3.49	V
			LCD8DPWR.LC[4:0] bits = 0x0b	3.37	3.51	3.66	V
			LCD8DPWR.LC[4:0] bits = 0x0c	3.52	3.67	3.82	V
			LCD8DPWR.LC[4:0] bits = 0x0d	3.67	3.82	3.98	V
			LCD8DPWR.LC[4:0] bits = 0x0e	3.82	3.98	4.15	V
			LCD8DPWR.LC[4:0] bits = 0x0f	3.97	4.14	4.31	V
			LCD8DPWR.LC[4:0] bits = 0x10	4.12	4.29	4.47	V
			LCD8DPWR.LC[4:0] bits = 0x11	4.27	4.45	4.63	V
			LCD8DPWR.LC[4:0] bits = 0x12	4.42	4.61	4.80	V
			LCD8DPWR.LC[4:0] bits = 0x13	4.57	4.76	4.96	V

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
LCD drive voltage (1/3 bias, V_{C1} reference voltage) V_{DD} = 1.8 to 5.5 V *3 LCD8DPWR.BIASSEL bit = 1 LCD8DPWR.VCSEL bit = 0	V_{C3}	Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD8DPWR.LC[4:0] bits = 0x14	4.72	4.92	5.12	V
			LCD8DPWR.LC[4:0] bits = 0x15	4.87	5.08	5.29	V
			LCD8DPWR.LC[4:0] bits = 0x16	5.02	5.24	5.45	V
			LCD8DPWR.LC[4:0] bits = 0x17	5.10	5.31	5.53	V
			LCD8DPWR.LC[4:0] bits = 0x18	5.17	5.39	5.61	V
			LCD8DPWR.LC[4:0] bits = 0x19	5.25	5.47	5.69	V
			LCD8DPWR.LC[4:0] bits = 0x1a	5.32	5.55	5.78	V
			LCD8DPWR.LC[4:0] bits = 0x1b	5.40	5.63	5.86	V
			LCD8DPWR.LC[4:0] bits = 0x1c	5.44	5.71	5.97	V
			LCD8DPWR.LC[4:0] bits = 0x1d	5.52	5.78	6.05	V
			LCD8DPWR.LC[4:0] bits = 0x1e	5.59	5.86	6.13	V
			LCD8DPWR.LC[4:0] bits = 0x1f	5.67	5.94	6.21	V
LCD drive voltage (1/3 bias, V_{C2} reference voltage) V_{DD} = 1.8 to 5.5 V *3 LCD8DPWR.BIASSEL bit = 1 LCD8DPWR.VCSEL bit = 1	V_{C1}	Connect 1 MΩ load resistor between V_{SS} and V_{C1}		0.323 × V_{C3} (Typ.)	–	0.344 × V_{C3} (Typ.)	V
	V_{C2}	Connect 1 MΩ load resistor between V_{SS} and V_{C2}		0.646 × V_{C3} (Typ.)	–	0.687 × V_{C3} (Typ.)	V
	V_{C3}	Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD8DPWR.LC[4:0] bits = 0x00	2.48	2.58	2.67	V
			LCD8DPWR.LC[4:0] bits = 0x01	2.56	2.66	2.75	V
			LCD8DPWR.LC[4:0] bits = 0x02	2.63	2.73	2.84	V
			LCD8DPWR.LC[4:0] bits = 0x03	2.72	2.81	2.90	V
			LCD8DPWR.LC[4:0] bits = 0x04	2.80	2.89	2.98	V
			LCD8DPWR.LC[4:0] bits = 0x05	2.88	2.97	3.06	V
			LCD8DPWR.LC[4:0] bits = 0x06	2.95	3.05	3.14	V
			LCD8DPWR.LC[4:0] bits = 0x07	3.03	3.13	3.23	V
			LCD8DPWR.LC[4:0] bits = 0x08	3.10	3.20	3.31	V
			LCD8DPWR.LC[4:0] bits = 0x09	3.18	3.28	3.39	V
			LCD8DPWR.LC[4:0] bits = 0x0a	3.26	3.36	3.47	V
			LCD8DPWR.LC[4:0] bits = 0x0b	3.41	3.52	3.63	V
			LCD8DPWR.LC[4:0] bits = 0x0c	3.56	3.67	3.79	V
			LCD8DPWR.LC[4:0] bits = 0x0d	3.71	3.83	3.95	V
			LCD8DPWR.LC[4:0] bits = 0x0e	3.86	3.99	4.11	V
			LCD8DPWR.LC[4:0] bits = 0x0f	4.01	4.14	4.27	V
			LCD8DPWR.LC[4:0] bits = 0x10	4.17	4.30	4.44	V
			LCD8DPWR.LC[4:0] bits = 0x11	4.32	4.46	4.60	V
			LCD8DPWR.LC[4:0] bits = 0x12	4.47	4.61	4.76	V
			LCD8DPWR.LC[4:0] bits = 0x13	4.62	4.77	4.92	V
			LCD8DPWR.LC[4:0] bits = 0x14	4.78	4.93	5.08	V
			LCD8DPWR.LC[4:0] bits = 0x15	4.93	5.09	5.24	V
			LCD8DPWR.LC[4:0] bits = 0x16	5.08	5.24	5.41	V
			LCD8DPWR.LC[4:0] bits = 0x17	5.16	5.32	5.49	V
			LCD8DPWR.LC[4:0] bits = 0x18	5.23	5.40	5.57	V
			LCD8DPWR.LC[4:0] bits = 0x19	5.31	5.48	5.65	V
			LCD8DPWR.LC[4:0] bits = 0x1a	5.39	5.56	5.73	V
			LCD8DPWR.LC[4:0] bits = 0x1b	5.46	5.64	5.81	V
			LCD8DPWR.LC[4:0] bits = 0x1c	5.54	5.72	5.89	V
			LCD8DPWR.LC[4:0] bits = 0x1d	5.62	5.80	5.97	V
			LCD8DPWR.LC[4:0] bits = 0x1e	5.69	5.87	6.06	V
			LCD8DPWR.LC[4:0] bits = 0x1f	5.77	5.95	6.14	V
LCD drive voltage (1/2 bias, V_{C1} reference voltage) V_{DD} = 1.8 to 5.5 V *3 LCD8DPWR.BIASSEL bit = 0 LCD8DPWR.VCSEL bit = 0	V_{C1}	Connect 1 MΩ load resistor between V_{SS} and V_{C1}(V_{C2})		0.468 × V_{C3} (Typ.)	–	0.533 × V_{C3} (Typ.)	V
	V_{C3}	Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD8DPWR.LC[4:0] bits = 0x00	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x01	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x02	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x03	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x04	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x05	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x06	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x07	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x08	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x09	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x0a	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x0b	2.24	2.34	2.44	V
			LCD8DPWR.LC[4:0] bits = 0x0c	2.34	2.45	2.55	V
			LCD8DPWR.LC[4:0] bits = 0x0d	2.44	2.55	2.66	V
			LCD8DPWR.LC[4:0] bits = 0x0e	2.54	2.65	2.77	V
			LCD8DPWR.LC[4:0] bits = 0x0f	2.64	2.76	2.87	V
			LCD8DPWR.LC[4:0] bits = 0x10	2.74	2.86	2.98	V
			LCD8DPWR.LC[4:0] bits = 0x11	2.84	2.97	3.09	V
			LCD8DPWR.LC[4:0] bits = 0x12	2.94	3.07	3.20	V

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
LCD drive voltage (1/2 bias, V_{C1} reference voltage) V_{DD} = 1.8 to 5.5 V *3 LCD8DPWR.BIASSEL bit = 0 LCD8DPWR.VCSEL bit = 0	V_{C3}	Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD8DPWR.LC[4:0] bits = 0x13	3.04	3.18	3.31	V
			LCD8DPWR.LC[4:0] bits = 0x14	3.14	3.28	3.42	V
			LCD8DPWR.LC[4:0] bits = 0x15	3.25	3.39	3.53	V
			LCD8DPWR.LC[4:0] bits = 0x16	3.35	3.49	3.63	V
			LCD8DPWR.LC[4:0] bits = 0x17	3.40	3.54	3.69	V
			LCD8DPWR.LC[4:0] bits = 0x18	3.45	3.59	3.74	V
			LCD8DPWR.LC[4:0] bits = 0x19	3.50	3.65	3.80	V
			LCD8DPWR.LC[4:0] bits = 0x1a	3.55	3.70	3.85	V
			LCD8DPWR.LC[4:0] bits = 0x1b	3.60	3.75	3.91	V
			LCD8DPWR.LC[4:0] bits = 0x1c	3.63	3.80	3.98	V
			LCD8DPWR.LC[4:0] bits = 0x1d	3.68	3.86	4.03	V
			LCD8DPWR.LC[4:0] bits = 0x1e	3.73	3.91	4.09	V
			LCD8DPWR.LC[4:0] bits = 0x1f	3.78	3.96	4.14	V
LCD drive voltage (1/2 bias, V_{C2} reference voltage) V_{DD} = 1.8 to 5.5 V *3 LCD8DPWR.BIASSEL bit = 0 LCD8DPWR.VCSEL bit = 1	V_{C2}	Connect 1 MΩ load resistor between V_{SS} and V_{C2}(V_{C1})		0.485 × V_{C3} (Typ.)	–	0.515 × V_{C3} (Typ.)	V
	V_{C3}	Connect 1 MΩ load resistor between V_{SS} and V_{C3}	LCD8DPWR.LC[4:0] bits = 0x00	3.31	3.44	3.56	V
			LCD8DPWR.LC[4:0] bits = 0x01	3.41	3.54	3.67	V
			LCD8DPWR.LC[4:0] bits = 0x02	3.51	3.65	3.78	V
			LCD8DPWR.LC[4:0] bits = 0x03	3.63	3.75	3.87	V
			LCD8DPWR.LC[4:0] bits = 0x04	3.73	3.85	3.98	V
			LCD8DPWR.LC[4:0] bits = 0x05	3.84	3.96	4.08	V
			LCD8DPWR.LC[4:0] bits = 0x06	3.94	4.06	4.19	V
			LCD8DPWR.LC[4:0] bits = 0x07	4.04	4.17	4.30	V
			LCD8DPWR.LC[4:0] bits = 0x08	4.14	4.27	4.41	V
			LCD8DPWR.LC[4:0] bits = 0x09	4.24	4.38	4.51	V
			LCD8DPWR.LC[4:0] bits = 0x0a	4.34	4.48	4.62	V
			LCD8DPWR.LC[4:0] bits = 0x0b	4.55	4.69	4.84	V
			LCD8DPWR.LC[4:0] bits = 0x0c	4.75	4.90	5.05	V
			LCD8DPWR.LC[4:0] bits = 0x0d	4.95	5.11	5.27	V
			LCD8DPWR.LC[4:0] bits = 0x0e	5.15	5.32	5.48	V
			LCD8DPWR.LC[4:0] bits = 0x0f	5.35	5.53	5.70	V
			LCD8DPWR.LC[4:0] bits = 0x10	5.56	5.73	5.91	V
			LCD8DPWR.LC[4:0] bits = 0x11	5.76	5.94	6.13	V
			LCD8DPWR.LC[4:0] bits = 0x12	5.96	6.15	6.34	V
			LCD8DPWR.LC[4:0] bits = 0x13	6.17	6.36	6.56	V
			LCD8DPWR.LC[4:0] bits = 0x14	6.37	6.57	6.77	V
			LCD8DPWR.LC[4:0] bits = 0x15	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x16	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x17	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x18	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x19	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x1a	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x1b	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x1c	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x1d	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x1e	–	–	–	V
			LCD8DPWR.LC[4:0] bits = 0x1f	–	–	–	V
Segment/Common output current	I_{SEGH}	SEG0–55, COM0–7, V_{SEGH} = V_{C3}/V_{C2}/V_{C1} - 0.1 V, Ta = -40 to 105°C		–	–	-10	µA
	I_{SEGL}	SEG0–55, COM0–7, V_{SEGL} = V_{C3}/V_{C2}/V_{C1} + 0.1 V, Ta = -40 to 105°C		10	–	–	µA
LCD circuit current (1/3 bias, V_{C2} reference voltage, Waveform B)	I_{LCD1}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 0 *1 *2		–	2.5	5.3	µA
		LCD8DDSP.DSPC[1:0] bits = 0x2 (all on), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 0 *1 *2		–	1.1	2.4	µA
LCD circuit current (1/3 bias, V_{C1} reference voltage, Waveform B)	I_{LCD1}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 0 *1 *2		–	4.8	10.2	µA
		LCD8DDSP.DSPC[1:0] bits = 0x2 (all on), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 0 *1 *2		–	1.9	4.4	µA

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
LCD circuit current (1/2 bias, V_{C2} reference voltage, Waveform B)	I_{LCD2}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 0 *1 *2	–	5.2	10.4	µA
		LCD8DDSP.DSPC[1:0] bits = 0x2 (all on), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 0 *1 *2	–	1.2	2.5	µA
LCD circuit current (1/2 bias, V_{C1} reference voltage, Waveform B)	I_{LCD2}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 0 *1 *2	–	3.9	8.1	µA
		LCD8DDSP.DSPC[1:0] bits = 0x2 (all on), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 0 *1 *2	–	1.1	2.3	µA
LCD circuit current (1/3 bias, V_{C2} reference voltage, Waveform A)	I_{LCD3}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 1 *1 *2	–	4.7	9.6	µA
		LCD8DDSP.DSPC[1:0] bits = 0x2 (all on), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 1 *1 *2	–	5.7	11.9	µA
LCD circuit current (1/3 bias, V_{C1} reference voltage, Waveform A)	I_{LCD3}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 1 *1 *2	–	9.0	18.8	µA
		LCD8DDSP.DSPC[1:0] bits = 0x2 (all on), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 1 *1 *2	–	11.2	23.8	µA
LCD circuit current (1/2 bias, V_{C2} reference voltage, Waveform A)	I_{LCD4}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 1 *1 *2	–	8.3	16.6	µA
		LCD8DDSP.DSPC[1:0] bits = 0x2 (all on), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 1 *1 *2	–	7.0	14.1	µA
LCD circuit current (1/2 bias, V_{C1} reference voltage, Waveform A)	I_{LCD4}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 1 *1 *2	–	6.1	12.3	µA
		LCD8DDSP.DSPC[1:0] bits = 0x2 (all on), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 1 *1 *2	–	5.2	10.7	µA
LCD circuit current in heavy load protection mode (1/3 bias, V_{C2} reference voltage, Waveform B)	I_{LCDH1}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 0, LCD8DPWR.HVLD bit = 1 *1 *2	–	20.4	49.2	µA
LCD circuit current in heavy load protection mode (1/2 bias, V_{C2} reference voltage, Waveform B)	I_{LCDH2}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 0, LCD8DPWR.HVLD bit = 1 *1 *2	–	16.8	38.1	µA
LCD circuit current in heavy load protection mode (1/3 bias, V_{C2} reference voltage, Waveform A)	I_{LCDH3}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 1, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 1, LCD8DPWR.HVLD bit = 1 *1 *2	–	22.5	52.4	µA
LCD circuit current in heavy load protection mode (1/2 bias, V_{C2} reference voltage, Waveform A)	I_{LCDH4}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 0, LCD8DPWR.VCSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 1, LCD8DPWR.HVLD bit = 1 *1 *2	–	19.9	42.6	µA

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
LCD circuit current (1/3 bias, Waveform B, when internal voltage dividing resistors are used)	I_{LCDR1}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 0, LCD8DPWR.RESISEL[1:0] bits = 0x1 *1 *2	–	35.5	72.7	μA
LCD circuit current (1/2 bias, Waveform B, when internal voltage dividing resistors are used)	I_{LCDR2}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 0, LCD8DPWR.RESISEL[1:0] bits = 0x1 *1 *2	–	54.0	111.3	μA
LCD circuit current (1/3 bias, Waveform A, when internal voltage dividing resistors are used)	I_{LCDR3}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 1, LCD8DTIM2.LCDWAVE bit = 1, LCD8DPWR.RESISEL[1:0] bits = 0x1 *1 *2	–	36.8	75.4	μA
LCD circuit current (1/2 bias, Waveform A, when internal voltage dividing resistors are used)	I_{LCDR4}	LCD8DDSP.DSPC[1:0] bits = 0x1 (checker pattern), LCD8DPWR.BIASSEL bit = 0, LCD8DTIM2.LCDWAVE bit = 1, LCD8DPWR.RESISEL[1:0] bits = 0x1 *1 *2	–	55.6	114.5	μA

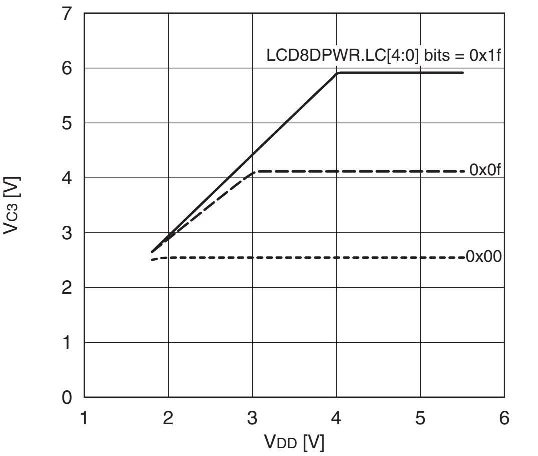
*1 Other LCD driver settings: LCD8DPWR.LC[4:0] bits = 0x1f, CLK_LCD8D = 32 kHz, LCD8DTIM1.FRMCNT[4:0] bits = 0x01 (frame frequency = 64 Hz)

*2 The value is added to the current consumption in HALT/RUN mode. Current consumption increases according to the display contents and panel load.

*3 If the power supply voltage V_{DD} is within the range between 1.8 to 4.0 V (when V_{C2} reference voltage is selected) or 1.8 to 3.0 V (when V_{C1} reference voltage is selected), the LCD drive voltage is decreased more than the value set using the LCD8WR.LC[4:0] bits. For details, refer to the LCD drive voltage-supply voltage characteristic graphs.

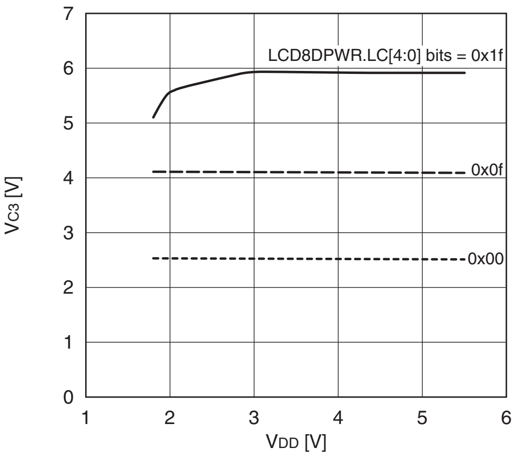
LCD drive voltage-supply voltage characteristic (V_{C2} reference voltage)

Ta = 25°C, Typ. value, when a 1 MΩ load resistor is connected between V_{SS} and V_{C3} (no panel load)



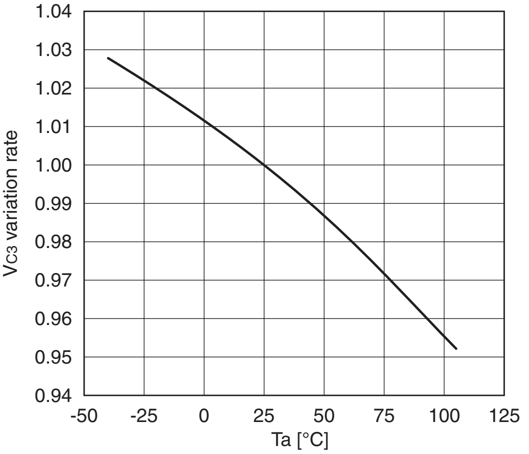
LCD drive voltage-supply voltage characteristic (V_{C1} reference voltage)

Ta = 25°C, Typ. value, when a 1 MΩ load resistor is connected between V_{SS} and V_{C3} (no panel load)



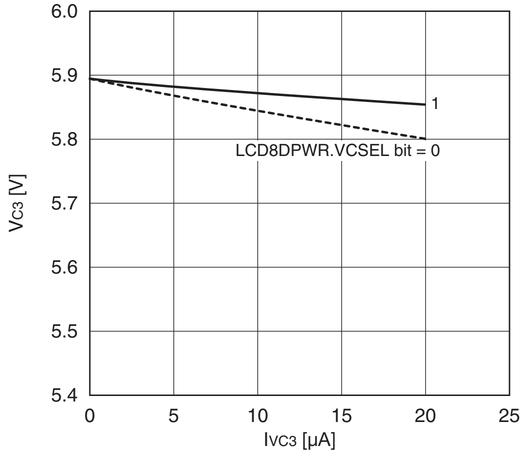
LCD drive voltage-temperature characteristic (V_{C1}/V_{C2} reference voltage)

Typ. value



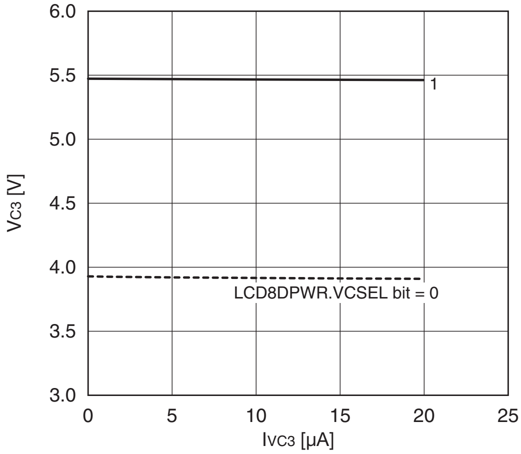
LCD drive voltage-load characteristic (1/3 bias)

V_{DD} = 5.5 V, Ta = 25°C, Typ. value, LCD8DPWR.LC[4:0] bits = 0x1f, when a load is connected to the V_{C3} pin only



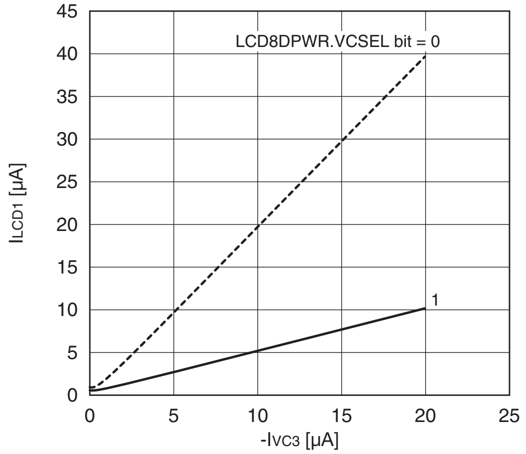
LCD drive voltage-load characteristic (1/2 bias)

V_{DD} = 5.5 V, Ta = 25°C, Typ. value, LCD8DPWR.LC[4:0] bits = 0x0f (V_{C2} reference voltage)/0x1f (V_{C1} reference voltge), when a load is connected to the V_{C3} pin only



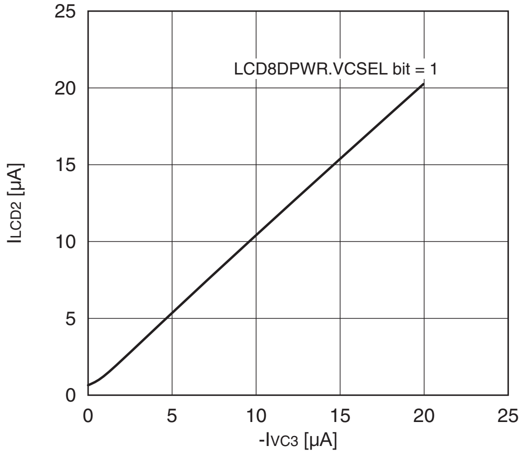
LCD circuit current-load characteristic (1/3 bias)

V_{DD} = 5.5 V, Ta = 25°C, Typ. value, LCD8DPWR.LC[4:0] bits = 0x1f, when a load is connected to the V_{C3} pin only



LCD circuit current-load characteristic (1/2 bias)

V_{DD} = 5.5 V, Ta = 25°C, Typ. value, LCD8DPWR.LC[4:0] bits = 0x0f, when a load is connected to the V_{C3} pin only



23.13 R/F Converter (RFC) Characteristics

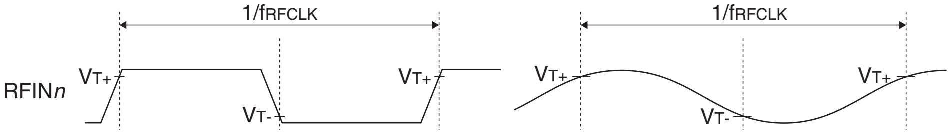
R/F converter characteristics change depending on conditions (board pattern, components used, etc.). Use these characteristic values as a reference and perform evaluation using the actual printed circuit board.

Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°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	$\Delta f_{RFCLK}/\Delta IC$	Ta = 25°C *1		-40	–	40	%
Reference resistor/resistive sensor resistance	R_{REF}, R_{SEN}			10	–	-	kΩ
Reference capacitance	C_{REF}			100	–	–	pF
Time base counter clock frequency	f_{TCCLK}			–	–	33	MHz
High level Schmitt input threshold voltage	V_{T+}			0.5 × V_{DD}	–	0.8 × V_{DD}	V
Low level Schmitt input threshold voltage	V_{T-}			0.2 × V_{DD}	–	0.5 × V_{DD}	V
Schmitt input hysteresis voltage	ΔV_T			165	–	–	mV
R/F converter operating current	I_{RFC}	R_{REF}/R_{SEN} = 100 kΩ,	5.5 V	–	165	230	µA
		C_{REF} = 1,000 pF, Ta = 25°C	3.6 V	–	75	110	µA

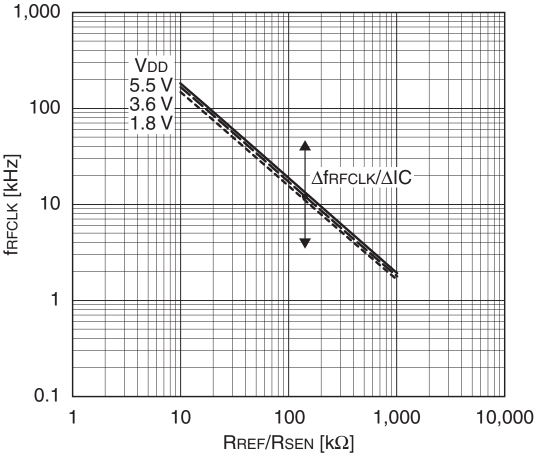
*1 In this characteristic, unevenness between production lots, and variations in measurement board, resistances and capacitances are taken into account.

Waveforms for external clock input mode



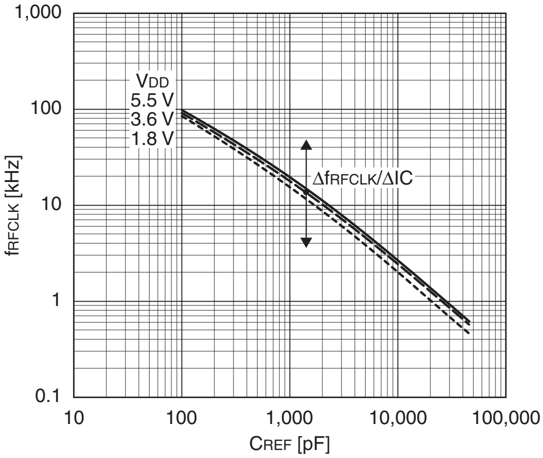
RFC reference/sensor oscillation frequency-resistance characteristic

CREF = 1,000 pF, Ta = 25°C, Typ. value



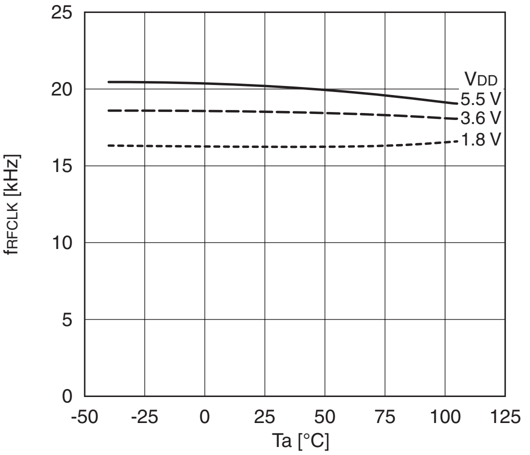
RFC reference/sensor oscillation frequency-capacitance characteristic

RREF/RSEN = 100 kΩ, Ta = 25°C, Typ. value



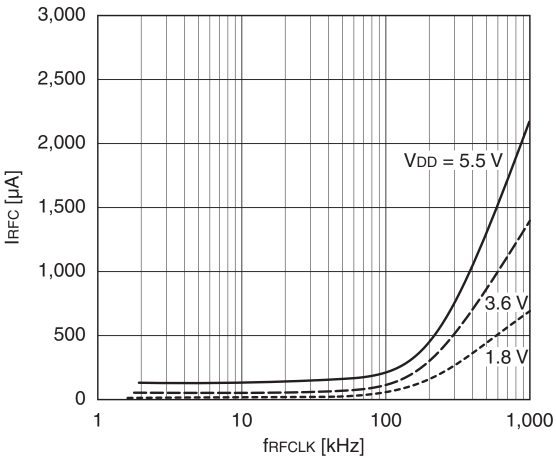
RFC reference/sensor oscillation frequency-temperature characteristic

RREF/RSEN = 100 kΩ, CREF = 1,000 pF, Typ. value



RFC reference/sensor oscillation current consumption-frequency characteristic

CREF = 1,000 pF, Ta = 25°C, Typ. value



23.14 12-bit A/D Converter (ADC12A) Characteristics

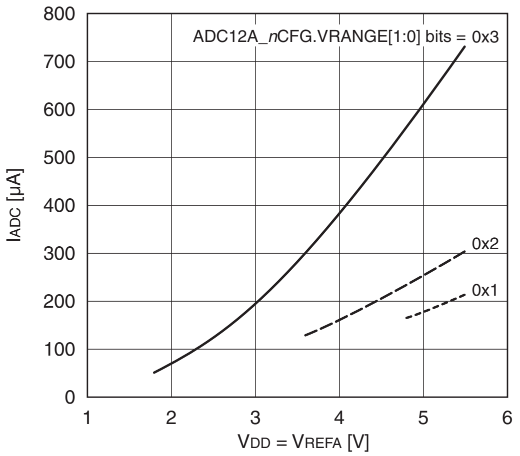
Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, VREFA*n* = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C,
ADC12A_*n*TRG.SMPCLK[2:0] bits = 0x3 (7cycles)

Item	Symbol	Condition	V_{DD}	Ta	Min.	Typ.	Max.	Unit
VREFA*n* voltage range	V_{REFA}		–	–	1.8	–	V_{DD}	V
A/D conversion clock frequency	f_{CLK_ADC12A}		–	–	16	–	2,200	kHz
Sampling rate *1	f_{SMP}		–	–	–	–	100	ksps
Integral nonlinearity *2	INL	V_{DD} = VREFA*n* *3	–	-40 to 85°C	–	–	±3	LSB
			–	-40 to 105°C	–	–	±4	LSB
Differential nonlinearity	DNL	V_{DD} = VREFA*n* *3	–	-40 to 85°C	–	–	±3	LSB
			–	-40 to 105°C	–	–	±4	LSB
Zero-scale error	ZSE	V_{DD} = VREFA*n* *3	–	–	–	–	±5	LSB
Full-scale error	FSE	V_{DD} = VREFA*n* *3	–	–	–	–	±5	LSB
Analog input resistance	R_{ADIN}		–	–	–	–	4	kΩ
Analog input capacitance	C_{ADIN}		–	–	–	–	30	pF
A/D converter circuit current	I_{ADC}	ADC12A_*n*CFG.VRANGE[1:0] bits = 0x3, V_{DD} = V_{REFA}, ADIN = V_{REFA}/2, f_{SMP} = 100 ksps, Ta = 25°C	3.6 V	–	–	400	700	µA
		ADC12A_*n*CFG.VRANGE[1:0] bits = 0x2, V_{DD} = V_{REFA}, ADIN = V_{REFA}/2, f_{SMP} = 100 ksps, Ta = 25°C	4.8 V	–	–	230	470	µA
		ADC12A_*n*CFG.VRANGE[1:0] bits = 0x1, V_{DD} = V_{REFA}, ADIN = V_{REFA}/2, f_{SMP} = 100 ksps, Ta = 25°C	5.5 V	–	–	210	390	µA

*1 The Max. value is the value when the A/D conversion clock frequency f_{CLK_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 VREFA*n*.

A/D converter current consumption-power supply voltage characteristic

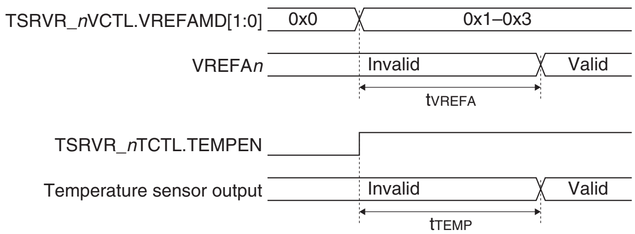
V_{DD} = V_{REFA}, ADIN = V_{REFA}/2, f_{SMP} = 100 ksps, Ta = 25°C, Typ. value



23.15 Temperature Sensor/Reference Voltage Generator (TSRVR) Characteristics

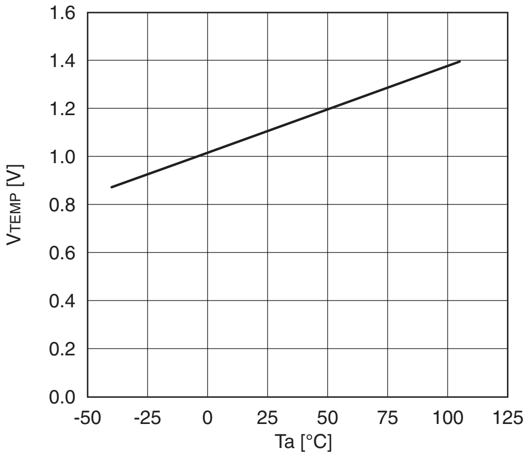
Unless otherwise specified: V_{DD} = 1.8 to 5.5 V, V_{SS} = 0 V, Ta = -40 to 105°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
V_{REFA} (2.5 V) output voltage	V_{VO25}	V_{DD} = 2.7 to 5.5 V	2.4	2.5	2.6	V
V_{REFA} (2.0 V) output voltage	V_{VO20}	V_{DD} = 2.2 to 5.5 V	1.9	2.0	2.1	V
V_{REFA} (V_{DD}) output voltage	V_{VODD}	V_{DD} = 1.8 to 5.5 V	V_{DD} - 0.1	V_{DD}	V_{DD} + 0.1	V
V_{REFA} (2.5/2.0 V) operating current	I_{VO1}	V_{DD} = 5.5 V, Ta = 25°C	25	40	60	µA
V_{REFA} (V_{DD}) operating current	I_{VO2}	V_{DD} = 5.5 V, Ta = 25°C	–	0	0.1	µA
V_{REFA} output voltage stabilization time	t_{VREFA}	C_{VREFA} = 0.1 µF	–	1.5	5	ms
Temperature sensor output voltage	V_{TEMP}	V_{DD} = 2.2 to 5.5 V, Ta = 25°C	1.06	1.09	1.12	V
Temperature sensor output voltage temperature coefficient	ΔV_{TEMP}	V_{DD} = 2.2 to 5.5 V	–	3.6 ± 3%	3.6 ± 7%	mV/°C
Temperature sensor operating current	I_{VTEMP}	V_{DD} = 5.5 V, Ta = 25°C	10	16	22	µA
Temperature sensor output stabilization time	t_{TEMP}		–	–	200	µs

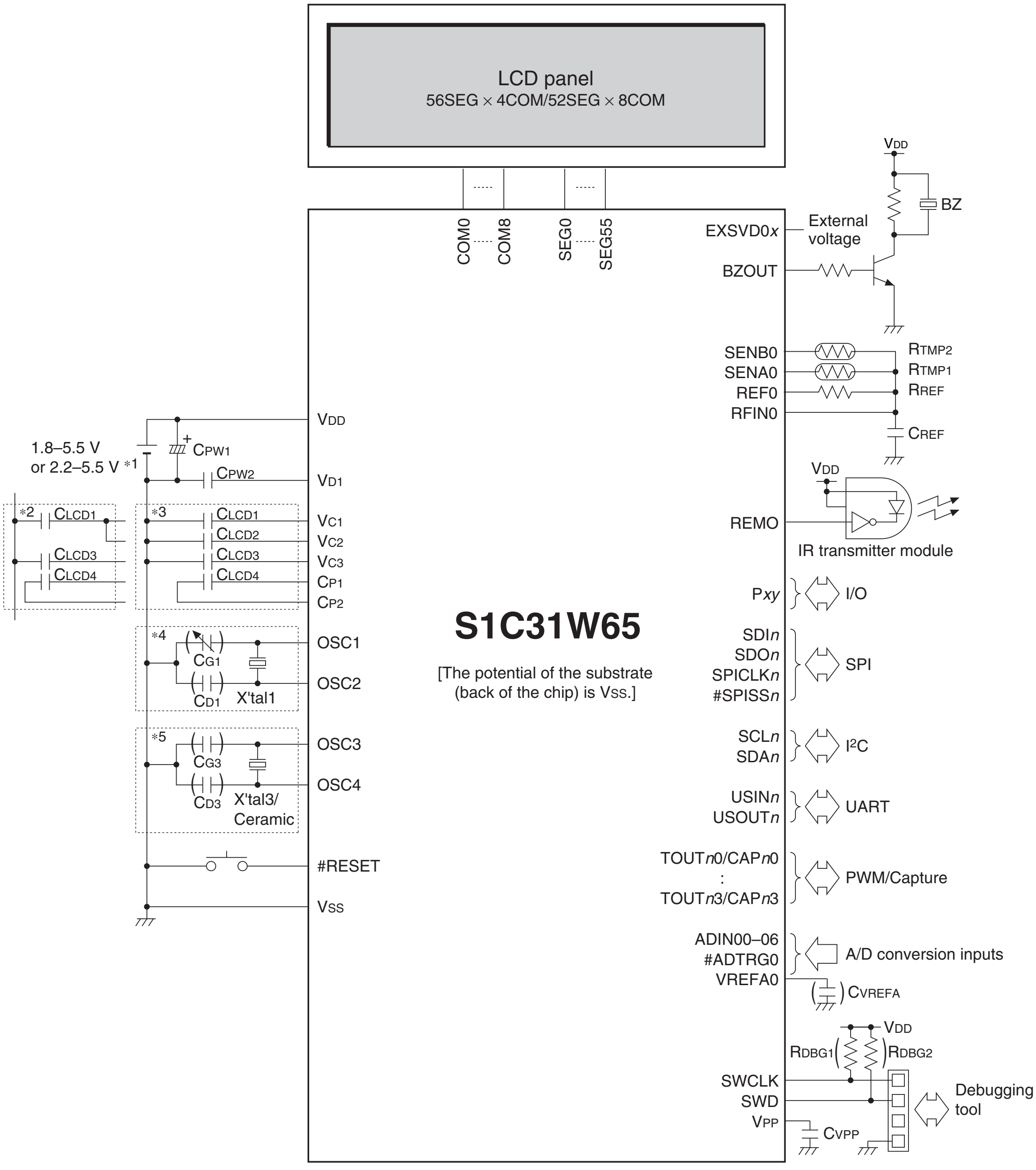


Temperature sensor output voltage-temperature characteristic

V_{DD} = 2.2 to 5.5 V, Typ. value



24 Basic External Connection Diagram



*1: For Flash programming
*3: When 1/2 bias is selected
*3: When 1/3 bias is selected
*4: When OSC1 crystal oscillator is selected
*5: When OSC3 crystal/ceramic oscillator is selected
( ): 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	Seiko Epson product
Ceramic	Ceramic resonator	Murata Manufacturing product
C_{G3}	OSC3 gate capacitor	Ceramic capacitor
C_{D3}	OSC3 drain capacitor	Ceramic capacitor
R_{CR3}	OSC3 oscillating resistor	Thick film chip resistor
C_{PW1}	Bypass capacitor between V_{SS} and V_{DD}	Ceramic capacitor or electrolytic capacitor
C_{PW2}	Capacitors between V_{SS} and V_{D1}	Ceramic capacitor
C_{LCD1-3}	Capacitors between V_{SS} and V_{C1-3}	Ceramic capacitor
C_{LCD4}	Capacitor between C_{P1} and C_{P2}	Ceramic capacitor
BZ	Piezoelectric buzzer	PS1240P02 manufactured by TDK Corporation
R_{DBG1-2}	Debug pin 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
		Humidity sensor C15-M53R manufactured by SHINYEI Technology Co.,Ltd. (* In AC oscillation mode for resistive sensor measurements)
C_{REF}	RFC reference capacitor	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.

25 Package

TQFP15-100PIN (P-TQFP100-1414-0.50)

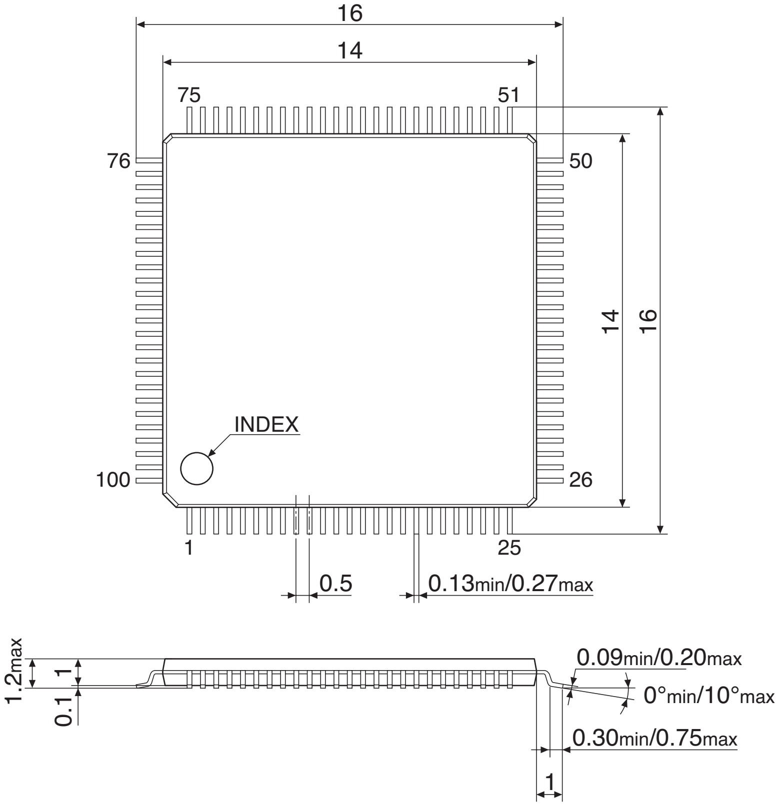


Figure 25.1 TQFP15-100PIN Package Dimensions

Appendix A List of Peripheral Circuit Control Registers

0x4000 0000 System Register (SYS)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0000	SYSPROT (System Protect Register)	15–0	PROT[15:0]	0x0000	H0	R/W	–

0x4000 0020 Power Generator (PWGA)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0020	PWGACTL (PWGA Control Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	REGDIS	0	H0	R/WP	
		4	REGSEL	1	H0	R/WP	
		3–2	–	0x0	–	R	
		1–0	REGMODE[1:0]	0x0	H0	R/WP	

0x4000 0040–0x4000 0050 Clock Generator (CLG)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0040	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	
0x4000 0042	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	
0x4000 0044	CLGIOSC (CLG IOSC Control Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6–5	IOSCWT[1:0]	0x3	H0	R/WP	
		4	IOSCSTM	0	H0	R/WP	
		3	–	0	–	R	
		2–0	IOSCFQ[2:0]	0x3	H0	R/WP	
0x4000 0046	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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0048	CLGOSC3 (CLG OSC3 Control Register)	15–13	–	0x0	–	R	–
		12–10	OSC3FQ[2:0]	0x5	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	–	0	–	R	
		2–0	OSC3WT[2:0]	0x6	H0	R/WP	
0x4000 004c	CLGINTF (CLG Interrupt Flag Register)	15–9	–	0x00	–	R	–
		8	IOSCTERIF	0	H0	R/W	Cleared by writing 1.
		7	–	0	–	R	–
		6	(reserved)	0	H0	R	
		5	OSC1STPIF	0	H0	R/W	Cleared by writing 1.
		4	IOSCTEDIF	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	
0x4000 004e	CLGINTE (CLG Interrupt Enable Register)	15–9	–	0x00	–	R	–
		8	IOSCTERIE	0	H0	R/W	
		7	–	0	–	R	
		6	(reserved)	0	H0	R	
		5	OSC1STPIE	0	H0	R/W	
		4	IOSCTEDIE	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	
0x4000 0050	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	
0x4000 0052	CLGTRIM1 (CLG Oscillation Frequency Trimming Register 1)	15–14	–	0x0	–	R	–
		13–8	IOSCLSAJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.
		7	–	0	–	R	–
		6–0	IOSCHSAJ[6:0]	*	H0	R/WP	* Determined by factory adjustment.
0x4000 0054	CLGTRIM2 (CLG Oscillation Frequency Trimming Register 2)	15	–	0	–	R	–
		14–8	OSC3SAJ[6:0]	*	H0	R/WP	* Determined by factory adjustment.
		7–6	–	0x0	–	R	–
		5–0	OSC1SAJ[5:0]	*	H0	R/WP	* Determined by factory adjustment.

0x4000 0060–0x4000 0062 System Reset Controller (SRC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0060	SRCRESETREQ (SRC Reset Request Flag Register)	15–8	–	0x00	–	R	–
		7–5	–	0x0	–	R	
		4	PORBORREQ	1	H2	R/W	Cleared by writing 1.
		3	XRESETREQ	1	H2	R/W	
		2	WDTRSTREQ	0	H1	R/W	
		1	SVDRSTREQ	0	H1	R/W	
		0	KEYRSTREQ	0	H1	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0062	SRCRESETPCTL (SRC #RESET Port Control Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6–4	(reserved)	0x0	H2	R/WP	Do not write 1.
		3	–	0	–	R	–
		2	(reserved)	0	H2	R/WP	Do not write 1.
		1	PORT_PLUP_EN	1	H2	R/WP	–
		0	PORT_RESET_EN	1	H2	R/WP	

0x4000 00a0–0x4000 00a4 Watchdog Timer (WDT2)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 00a0	WDT2CLK (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	
0x4000 00a2	WDT2CTL (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	WDTRUN[3:0]	0xa	H0	R/WP	–
0x4000 00a4	WDT2CMP (WDT2 Counter Compare Match Register)	15–10	–	0x00	–	R	–
		9–0	CMP[9:0]	0x3ff	H0	R/WP	

0x4000 00c0–0x4000 00d2 Real-time Clock (RTCA)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 00c0	RTCACTLL (RTCA Control Register (Low Byte))	7	–	0	–	R	–
		6	RTCBSY	0	H0	R	
		5	RTCHLD	0	H0	R/W	Cleared by setting the RTCACTLL.RTCRST bit to 1.
		4	RTC24H	0	H0	R/W	–
		3	–	0	–	R	
		2	RTCADJ	0	H0	R/W	Cleared by setting the RTCACTLL.RTCRST bit to 1.
		1	RTCRST	0	H0	R/W	–
		0	RTCRUN	0	H0	R/W	
0x4000 00c1	RTCACTLH (RTCA Control Register (High Byte))	7	RTCTRMBSY	0	H0	R	–
		6–0	RTCTRM[6:0]	0x00	H0	W	Read as 0x00.
0x4000 00c2	RTCAALM1 (RTCA 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	
0x4000 00c4	RTCAALM2 (RTCA 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	
0x4000 00c6	RTCASWCTL (RTCA 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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 00c8	RTCASEC (RTCA Second/1Hz Register)	15	–	0	–	R	–
		14–12	RTCSH[2:0]	0x0	H0	R/W	
		11–8	RTCSL[3:0]	0x0	H0	R/W	
		7	RTC1HZ	0	H0	R	Cleared by setting the RTCACTLL.RTCRST bit to 1.
		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	
0x4000 00ca	RTCAHUR (RTCA 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	
0x4000 00cc	RTCAMON (RTCA 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	
0x4000 00ce	RTCAYAR (RTCA 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	
0x4000 00d0	RTCAINTF (RTCA 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	–
		8	ALARMIF	0	H0	R/W	Cleared by writing 1.
		7	T1DAYIF	0	H0	R/W	
		6	T1HURIF	0	H0	R/W	
		5	T1MINIF	0	H0	R/W	
		4	T1SECIF	0	H0	R/W	
		3	T1_2SECIF	0	H0	R/W	
		2	T1_4SECIF	0	H0	R/W	
		1	T1_8SECIF	0	H0	R/W	
		0	T1_32SECIF	0	H0	R/W	
0x4000 00d2	RTCAINTE (RTCA 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	T1DAYIE	0	H0	R/W	
		6	T1HURIE	0	H0	R/W	
		5	T1MINIE	0	H0	R/W	
		4	T1SECIE	0	H0	R/W	
		3	T1_2SECIE	0	H0	R/W	
		2	T1_4SECIE	0	H0	R/W	
		1	T1_8SECIE	0	H0	R/W	
		0	T1_32SECIE	0	H0	R/W	

0x4000 0100–0x4000 0106 Supply Voltage Detector (SVD4) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0100	SVD4_0CLK (SVD4 Ch.0 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	
0x4000 0102	SVD4_0CTL (SVD4 Ch.0 Control Register)	15	VDSEL	0	H1	R/WP	–
		14–13	SVDSC[1:0]	0x0	H0	R/WP	Writing takes effect when the SVD4_0CTL.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	EXSEL	0	H1	R/W	
		2–1	SVDMD[1:0]	0x0	H0	R/W	
		0	MODEN	0	H1	R/W	
0x4000 0104	SVD4_0INTF (SVD4 Ch.0 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.
0x4000 0106	SVD4_0INTE (SVD4 Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	SVDIE	0	H0	R/W	

0x4000 0140–0x4000 014c 16-bit Timer (T16) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0140	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	
0x4000 0142	T16_0MOD (T16 Ch.0 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4000 0144	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	
0x4000 0146	T16_0TR (T16 Ch.0 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4000 0148	T16_0TC (T16 Ch.0 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x4000 014a	T16_0INTF (T16 Ch.0 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x4000 014c	T16_0INTE (T16 Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x4000 01b0 Flash Controller (FLASHC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 01b0	FLASHCWAIT (FLASHC Flash Read Cycle Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1–0	RDWAIT[1:0]	0x1	H0	R/WP	

0x4000 0200–0x4000 02e2 I/O Ports (PPORT)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0200	PPORTP0DAT (P0 Port Data Register)	15–8	P0OUT[7:0]	0x00	H0	R/W	–
		7–0	P0IN[7:0]	0x00	H0	R	
0x4000 0202	PPORTP0IOEN (P0 Port Enable Register)	15–8	P0IEN[7:0]	0x00	H0	R/W	–
		7–0	P0OEN[7:0]	0x00	H0	R/W	
0x4000 0204	PPORTP0RCTL (P0 Port Pull-up/down Control Register)	15–8	P0PDPU[7:0]	0x00	H0	R/W	–
		7–0	P0REN[7:0]	0x00	H0	R/W	
0x4000 0206	PPORTP0INTF (P0 Port Interrupt Flag Register)	15–8	P0FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
		7–0	P0RIF[7:0]	0x00	H0	R/W	
0x4000 0208	PPORTP0INTCTL (P0 Port Interrupt Control Register)	15–8	P0FIE[7:0]	0x00	H0	R/W	–
		7–0	P0RIE[7:0]	0x00	H0	R/W	
0x4000 020a	PPORTP0CHATEN (P0 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
		7–0	P0CHATEN[7:0]	0x00	H0	R/W	
0x4000 020c	PPORTP0MODSEL (P0 Port Mode Select Register)	15–8	–	0x00	–	R	–
		7–0	P0SEL[7:0]	0x00	H0	R/W	
0x4000 020e	PPORTP0FNCSEL (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	
0x4000 0210	PPORTP1DAT (P1 Port Data Register)	15–8	P1OUT[7:0]	0x00	H0	R/W	–
		7–0	P1IN[7:0]	0x00	H0	R	
0x4000 0212	PPORTP1IOEN (P1 Port Enable Register)	15–8	P1IEN[7:0]	0x00	H0	R/W	–
		7–0	P1OEN[7:0]	0x00	H0	R/W	
0x4000 0214	PPORTP1RCTL (P1 Port Pull-up/down Control Register)	15–8	P1PDPU[7:0]	0x00	H0	R/W	–
		7–0	P1REN[7:0]	0x00	H0	R/W	
0x4000 0216	PPORTP1INTF (P1 Port Interrupt Flag Register)	15–8	P1FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
		7–0	P1RIF[7:0]	0x00	H0	R/W	
0x4000 0218	PPORTP1INTCTL (P1 Port Interrupt Control Register)	15–8	P1FIE[7:0]	0x00	H0	R/W	–
		7–0	P1RIE[7:0]	0x00	H0	R/W	
0x4000 021a	PPORTP1CHATEN (P1 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
		7–0	P1CHATEN[7:0]	0x00	H0	R/W	
0x4000 021c	PPORTP1MODSEL (P1 Port Mode Select Register)	15–8	–	0x00	–	R	–
		7–0	P1SEL[7:0]	0x00	H0	R/W	
0x4000 021e	PPORTP1FNCSEL (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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0220	PPORTP2DAT (P2 Port Data Register)	15–8	P2OUT[7:0]	0x00	H0	R/W	–
		7–0	P2IN[7:0]	0x00	H0	R	
0x4000 0222	PPORTP2IOEN (P2 Port Enable Register)	15–8	P2IEN[7:0]	0x00	H0	R/W	–
		7–0	P2OEN[7:0]	0x00	H0	R/W	
0x4000 0224	PPORTP2RCTL (P2 Port Pull-up/down Control Register)	15–8	P2PDPU[7:0]	0x00	H0	R/W	–
		7–0	P2REN[7:0]	0x00	H0	R/W	
0x4000 0226	PPORTP2INTF (P2 Port Interrupt Flag Register)	15–8	P2FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
		7–0	P2RIF[7:0]	0x00	H0	R/W	
0x4000 0228	PPORTP2INTCTL (P2 Port Interrupt Control Register)	15–8	P2FIE[7:0]	0x00	H0	R/W	–
		7–0	P2RIE[7:0]	0x00	H0	R/W	
0x4000 022a	PPORTP2CHATEN (P2 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
		7–0	P2CHATEN[7:0]	0x00	H0	R/W	
0x4000 022c	PPORTP2MODSEL (P2 Port Mode Select Register)	15–8	–	0x00	–	R	–
		7–0	P2SEL[7:0]	0x00	H0	R/W	
0x4000 022e	PPORTP2FNCSEL (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	
0x4000 0230	PPORTP3DAT (P3 Port Data Register)	15–8	P3OUT[7:0]	0x00	H0	R/W	–
		7–0	P3IN[7:0]	0x00	H0	R	
0x4000 0232	PPORTP3IOEN (P3 Port Enable Register)	15–8	P3IEN[7:0]	0x00	H0	R/W	–
		7–0	P3OEN[7:0]	0x00	H0	R/W	
0x4000 0234	PPORTP3RCTL (P3 Port Pull-up/down Control Register)	15–8	P3PDPU[7:0]	0x00	H0	R/W	–
		7–0	P3REN[7:0]	0x00	H0	R/W	
0x4000 0236	PPORTP3INTF (P3 Port Interrupt Flag Register)	15–8	P3FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
		7–0	P3RIF[7:0]	0x00	H0	R/W	
0x4000 0238	PPORTP3INTCTL (P3 Port Interrupt Control Register)	15–8	P3FIE[7:0]	0x00	H0	R/W	–
		7–0	P3RIE[7:0]	0x00	H0	R/W	
0x4000 023a	PPORTP3CHATEN (P3 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
		7–0	P3CHATEN[7:0]	0x00	H0	R/W	
0x4000 023c	PPORTP3MODSEL (P3 Port Mode Select Register)	15–8	–	0x00	–	R	–
		7–0	P3SEL[7:0]	0x00	H0	R/W	
0x4000 023e	PPORTP3FNCSEL (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	
0x4000 0240	PPORTP4DAT (P4 Port Data Register)	15–8	P4OUT[7:0]	0x00	H0	R/W	–
		7–0	P4IN[7:0]	0x00	H0	R	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0242	PPORTP4IOEN (P4 Port Enable Register)	15–8	P4IEN[7:0]	0x00	H0	R/W	–
		7–0	P4OEN[7:0]	0x00	H0	R/W	
0x4000 0244	PPORTP4RCTL (P4 Port Pull-up/down Control Register)	15–8	P4PDPU[7:0]	0x00	H0	R/W	–
		7–0	P4REN[7:0]	0x00	H0	R/W	
0x4000 0246	PPORTP4INTF (P4 Port Interrupt Flag Register)	15–8	P4FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
		7–0	P4RIF[7:0]	0x00	H0	R/W	
0x4000 0248	PPORTP4INTCTL (P4 Port Interrupt Control Register)	15–8	P4FIE[7:0]	0x00	H0	R/W	–
		7–0	P4RIE[7:0]	0x00	H0	R/W	
0x4000 024a	PPORTP4CHATEN (P4 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
		7–0	P4CHATEN[7:0]	0x00	H0	R/W	
0x4000 024c	PPORTP4MODSEL (P4 Port Mode Select Register)	15–8	–	0x00	–	R	–
		7–0	P4SEL[7:0]	0x00	H0	R/W	
0x4000 024e	PPORTP4FNCSEL (P4 Port Function Select Register)	15–14	P47MUX[1:0]	0x3	H0	R/W	–
		13–12	P46MUX[1:0]	0x3	H0	R/W	
		11–10	P45MUX[1:0]	0x3	H0	R/W	
		9–8	P44MUX[1:0]	0x3	H0	R/W	
		7–6	P43MUX[1:0]	0x3	H0	R/W	
		5–4	P42MUX[1:0]	0x3	H0	R/W	
		3–2	P41MUX[1:0]	0x3	H0	R/W	
		1–0	P40MUX[1:0]	0x3	H0	R/W	
0x4000 0250	PPORTP5DAT (P5 Port Data Register)	15–8	P5OUT[7:0]	0x00	H0	R/W	–
		7–0	P5IN[7:0]	0x00	H0	R	
0x4000 0252	PPORTP5IOEN (P5 Port Enable Register)	15–8	P5IEN[7:0]	0x00	H0	R/W	–
		7–0	P5OEN[7:0]	0x00	H0	R/W	
0x4000 0254	PPORTP5RCTL (P5 Port Pull-up/down Control Register)	15–8	P5PDPU[7:0]	0x00	H0	R/W	–
		7–0	P5REN[7:0]	0x00	H0	R/W	
0x4000 0256	PPORTP5INTF (P5 Port Interrupt Flag Register)	15–8	P5FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
		7–0	P5RIF[7:0]	0x00	H0	R/W	
0x4000 0258	PPORTP5INTCTL (P5 Port Interrupt Control Register)	15–8	P5FIE[7:0]	0x00	H0	R/W	–
		7–0	P5RIE[7:0]	0x00	H0	R/W	
0x4000 025a	PPORTP5CHATEN (P5 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
		7–0	P5CHATEN[7:0]	0x00	H0	R/W	
0x4000 025c	PPORTP5MODSEL (P5 Port Mode Select Register)	15–8	–	0x00	–	R	–
		7–0	P5SEL[7:0]	0x00	H0	R/W	
0x4000 025e	PPORTP5FNCSEL (P5 Port Function Select Register)	15–14	P57MUX[1:0]	0x3	H0	R/W	–
		13–12	P56MUX[1:0]	0x3	H0	R/W	
		11–10	P55MUX[1:0]	0x3	H0	R/W	
		9–8	P54MUX[1:0]	0x3	H0	R/W	
		7–6	P53MUX[1:0]	0x3	H0	R/W	
		5–4	P52MUX[1:0]	0x3	H0	R/W	
		3–2	P51MUX[1:0]	0x3	H0	R/W	
		1–0	P50MUX[1:0]	0x3	H0	R/W	
0x4000 0260	PPORTP6DAT (P6 Port Data Register)	15–8	P6OUT[7:0]	0x00	H0	R/W	–
		7–0	P6IN[7:0]	0x00	H0	R	
0x4000 0262	PPORTP6IOEN (P6 Port Enable Register)	15–8	P6IEN[7:0]	0x00	H0	R/W	–
		7–0	P6OEN[7:0]	0x00	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0264	PPORTP6RCTL (P6 Port Pull-up/down Control Register)	15–8	P6PDPU[7:0]	0x00	H0	R/W	–
		7–0	P6REN[7:0]	0x00	H0	R/W	
0x4000 0266	PPORTP6INTF (P6 Port Interrupt Flag Register)	15–8	P6FIF[7:0]	0x00	H0	R/W	Cleared by writing 1.
		7–0	P6RIF[7:0]	0x00	H0	R/W	
0x4000 0268	PPORTP6INTCTL (P6 Port Interrupt Control Register)	15–8	P6FIE[7:0]	0x00	H0	R/W	–
		7–0	P6RIE[7:0]	0x00	H0	R/W	
0x4000 026a	PPORTP6CHATEN (P6 Port Chattering Filter Enable Register)	15–8	–	0x00	–	R	–
		7–0	P6CHATEN[7:0]	0x00	H0	R/W	
0x4000 026c	PPORTP6MODSEL (P6 Port Mode Select Register)	15–8	–	0x00	–	R	–
		7–0	P6SEL[7:0]	0x00	H0	R/W	
0x4000 026e	PPORTP6FNCSEL (P6 Port Function Select Register)	15–14	P67MUX[1:0]	0x3	H0	R/W	–
		13–12	P66MUX[1:0]	0x3	H0	R/W	
		11–10	P65MUX[1:0]	0x3	H0	R/W	
		9–8	P64MUX[1:0]	0x3	H0	R/W	
		7–6	P63MUX[1:0]	0x3	H0	R/W	
		5–4	P62MUX[1:0]	0x3	H0	R/W	
		3–2	P61MUX[1:0]	0x3	H0	R/W	
		1–0	P60MUX[1:0]	0x3	H0	R/W	
0x4000 02d0	PPORTPDDAT (Pd Port Data Register)	15–8	PDOUT[7:0]	0x00	H0	R/W	–
		7–5	PDIN[7:5]	0x0	H0	R	
		4	(reserved)	0	–	R	
		3–0	PDIN[3:0]	0x00	H0	R	
0x4000 02d2	PPORTPDIOEN (Pd Port Enable Register)	15–13	PDIEN[7:5]	0x0	H0	R/W	–
		12	(reserved)	0	H0	R/W	
		11–8	PDIEN[3:0]	0x0	H0	R/W	
		7–0	PDOEN[7:0]	0x10	H0	R/W	
0x4000 02d4	PPORTPDRCTL (Pd Port Pull-up/down Control Register)	15–13	PDPDPU[7:5]	0x0	H0	R/W	–
		12	(reserved)	0	H0	R/W	
		11–8	PDPDPU[3:0]	0x0	H0	R/W	
		7–5	PDREN[7:5]	0x0	H0	R/W	
		4	(reserved)	0	H0	R/W	
		3–0	PDREN[3:0]	0x0	H0	R/W	
0x4000 02dc	PPORTPDMODSEL (Pd Port Mode Select Register)	15–8	–	0x00	–	R	–
		7–0	PDSEL[7:0]	0x23	H0	R/W	
0x4000 02de	PPORTPDFNCSEL (Pd Port Function Select Register)	15–14	PD7MUX[1:0]	0x0	H0	R/W	–
		13–12	PD6MUX[1:0]	0x0	H0	R/W	
		11–10	PD5MUX[1:0]	0x2	H0	R/W	
		9–8	(reserved)	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	
0x4000 02e0	PPORTCLK (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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 02e2	PPORTINTFGRP (P Port Interrupt Flag Group Register)	15–8	–	0x00	–	R	–
		7	–	0	–	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	

0x4000 0300–0x4000 031e Universal Port Multiplexer (UPMUX)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0300	UPMUXP0MUX0 (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	
0x4000 0302	UPMUXP0MUX1 (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	
0x4000 0304	UPMUXP0MUX2 (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	
0x4000 0306	UPMUXP0MUX3 (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	
0x4000 0308	UPMUXP1MUX0 (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	
0x4000 030a	UPMUXP1MUX1 (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	
0x4000 030c	UPMUXP1MUX2 (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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 030e	UPMUXP1MUX3 (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	
0x4000 0310	UPMUXP2MUX0 (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	
0x4000 0312	UPMUXP2MUX1 (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	
0x4000 0314	UPMUXP2MUX2 (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	
0x4000 0316	UPMUXP2MUX3 (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	
0x4000 0318	UPMUXP3MUX0 (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	
0x4000 031a	UPMUXP3MUX1 (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	
0x4000 031c	UPMUXP3MUX2 (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	
0x4000 031e	UPMUXP3MUX3 (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	

0x4000 0380–0x4000 0394 UART (UART3) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0380	UART3_0CLK (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	
0x4000 0382	UART3_0MOD (UART3 Ch.0 Mode Register)	15–13	–	0x00	–	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	
0x4000 0384	UART3_0BR (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	
0x4000 0386	UART3_0CTL (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	
0x4000 0388	UART3_0TXD (UART3 Ch.0 Transmit Data Register)	15–8	–	0x00	–	R	–
		7–0	TXD[7:0]	0x00	H0	R/W	
0x4000 038a	UART3_0RXD (UART3 Ch.0 Receive Data Register)	15–8	–	0x00	–	R	–
		7–0	RXD[7:0]	0x00	H0	R	
0x4000 038c	UART3_0INTF (UART3 Ch.0 Status and Interrupt Flag Register)	15–10	–	0x00	–	R	–
		9	RBSY	0	H0/S0	R	
		8	TBSY	0	H0/S0	R	
		7	–	0	–	R	
		6	TENDIF	0	H0/S0	R/W	Cleared by writing 1.
		5	FEIF	0	H0/S0	R/W	Cleared by writing 1 or reading the UART3_0RXD 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 UART3_0RXD register.
		1	RB1FIF	0	H0/S0	R	
		0	TBEIF	1	H0/S0	R	Cleared by writing to the UART3_0TXD register.
0x4000 038e	UART3_0INTE (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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0390	UART3_0 TBEDMAEN (UART3 Ch.0 Transmit Buffer Empty DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	TBEDMAEN[3:0]	0x0	H0	R/W	
0x4000 0392	UART3_0 RB1FDMAEN (UART3 Ch.0 Receive Buffer One Byte Full DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	RB1FDMAEN[3:0]	0x0	H0	R/W	
0x4000 0394	UART3_0CAWF (UART3 Ch.0 Carrier Waveform Register)	15–8	–	0x00	–	R	–
		7–0	CRPER[7:0]	0x00	H0	R/W	

0x4000 03a0–0x4000 03ac 16-bit Timer (T16) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 03a0	T16_1CLK (T16 Ch.1 Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	0	H0	R/W	
		7–4	CLKDIV[3:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x4000 03a2	T16_1MOD (T16 Ch.1 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4000 03a4	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	
0x4000 03a6	T16_1TR (T16 Ch.1 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4000 03a8	T16_1TC (T16 Ch.1 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x4000 03aa	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.
0x4000 03ac	T16_1INTE (T16 Ch.1 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x4000 03b0–0x4000 03be Synchronous Serial Interface (SPIA) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 03b0	SPIA_0MOD (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	
0x4000 03b2	SPIA_0CTL (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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 03b4	SPIA_0TXD (SPIA Ch.0 Transmit Data Register)	15–0	TXD[15:0]	0x0000	H0	R/W	–
0x4000 03b6	SPIA_0RXD (SPIA Ch.0 Receive Data Register)	15–0	RXD[15:0]	0x0000	H0	R	–
0x4000 03b8	SPIA_0INTF (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 SPIA_0RXD register.
		0	TBEIF	1	H0/S0	R	Cleared by writing to the SPIA_0TXD register.
0x4000 03ba	SPIA_0INTE (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	
0x4000 03bc	SPIA_0TBEDMAEN (SPIA Ch.0 Transmit Buffer Empty DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	TBEDMAEN[3:0]	0x0	H0	R/W	
0x4000 03be	SPIA_0RBFDMAEN (SPIA Ch.0 Receive Buffer Full DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	RBFDMAEN[3:0]	0x0	H0	R/W	

0x4000 03c0–0x4000 03d6 I²C (I2C) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 03c0	I2C_0CLK (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	
0x4000 03c2	I2C_0MOD (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	
0x4000 03c4	I2C_0BR (I2C Ch.0 Baud-Rate Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6–0	BRT[6:0]	0x7f	H0	R/W	
0x4000 03c8	I2C_0OADR (I2C Ch.0 Own Address Register)	15–10	–	0x00	–	R	–
		9–0	OADR[9:0]	0x000	H0	R/W	
0x4000 03ca	I2C_0CTL (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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 03cc	I2C_0TXD (I2C Ch.0 Transmit Data Register)	15–8	–	0x00	–	R	–
		7–0	TXD[7:0]	0x00	H0	R/W	
0x4000 03ce	I2C_0RXD (I2C Ch.0 Receive Data Register)	15–8	–	0x00	–	R	–
		7–0	RXD[7:0]	0x00	H0	R	
0x4000 03d0	I2C_0INTF (I2C Ch.0 Status and Interrupt Flag Register)	15–13	–	0x0	–	R	–
		12	SDALOW	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 I2C_0RXD register.
		0	TBEIF	0	H0/S0	R	Cleared by writing to the I2C_0TXD register.
0x4000 03d2	I2C_0INTE (I2C Ch.0 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7	BYTEENDIE	0	H0	R/W	
		6	GCIE	0	H0	R/W	
		5	NACKIE	0	H0	R/W	
		4	STOPIE	0	H0	R/W	
		3	STARTIE	0	H0	R/W	
		2	ERRIE	0	H0	R/W	
		1	RBFIE	0	H0	R/W	
		0	TBEIE	0	H0	R/W	
0x4000 03d4	I2C_0TBEDMAEN (I2C Ch.0 Transmit Buffer Empty DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	TBEDMAEN[3:0]	0x0	H0	R/W	
0x4000 03d6	I2C_0RBFDMAEN (I2C Ch.0 Receive Buffer Full DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	RBFDMAEN[3:0]	0x0	H0	R/W	

0x4000 0400–0x4000 042c 16-bit PWM Timer (T16B) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0400	T16B_0CLK (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	
0x4000 0402	T16B_0CTL (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	
0x4000 0404	T16B_0MC (T16B Ch.0 Max Counter Data Register)	15–0	MC[15:0]	0xffff	H0	R/W	–

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0406	T16B_0TC (T16B Ch.0 Timer Counter Data Register)	15–0	TC[15:0]	0x0000	H0	R	–
0x4000 0408	T16B_0CS (T16B Ch.0 Counter Status Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPI3	0	H0	R	
		4	CAPI2	0	H0	R	
		3	CAPI1	0	H0	R	
		2	CAPI0	0	H0	R	
		1	UP_DOWN	1	H0	R	
		0	BSY	0	H0	R	
0x4000 040a	T16B_0INTF (T16B Ch.0 Interrupt Flag Register)	15–10	–	0x00	–	R	–
		9	CAPOW3IF	0	H0	R/W	Cleared by writing 1.
		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	
0x4000 040c	T16B_0INTE (T16B Ch.0 Interrupt Enable Register)	15–10	–	0x00	–	R	–
		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	
0x4000 040e	T16B_0MZDMAEN (T16B Ch.0 Counter Max/Zero DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	MZDMAEN[3:0]	0x0	H0	R/W	
0x4000 0410	T16B_0CCCTL0 (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	
		0	CCMD	0	H0	R/W	
0x4000 0412	T16B_0CCR0 (T16B Ch.0 Compare/Capture 0 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 0414	T16B_0CC0DMAEN (T16B Ch.0 Compare/Capture 0 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC0DMAEN[3:0]	0x0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0418	T16B_0CCCTL1 (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	
		0	CCMD	0	H0	R/W	
0x4000 041a	T16B_0CCR1 (T16B Ch.0 Compare/ Capture 1 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 041c	T16B_0CC1DMAEN (T16B Ch.0 Compare/ Capture 1 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC1DMAEN[3:0]	0x0	H0	R/W	
0x4000 0420	T16B_0CCCTL2 (T16B Ch.0 Compare/ Capture 2 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	
0x4000 0422	T16B_0CCR2 (T16B Ch.0 Compare/ Capture 2 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 0424	T16B_0CC2DMAEN (T16B Ch.0 Compare/ Capture 2 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC2DMAEN[3:0]	0x0	H0	R/W	
0x4000 0428	T16B_0CCCTL3 (T16B Ch.0 Compare/ Capture 3 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	
0x4000 042a	T16B_0CCR3 (T16B Ch.0 Compare/ Capture 3 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 042c	T16B_0CC3DMAEN (T16B Ch.0 Compare/ Capture 3 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC3DMAEN[3:0]	0x0	H0	R/W	

0x4000 0440–0x4000 046c 16-bit PWM Timer (T16B) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0440	T16B_1CLK (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	
0x4000 0442	T16B_1CTL (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	
0x4000 0444	T16B_1MC (T16B Ch.1 Max Counter Data Register)	15–0	MC[15:0]	0xffff	H0	R/W	–
0x4000 0446	T16B_1TC (T16B Ch.1 Timer Counter Data Register)	15–0	TC[15:0]	0x0000	H0	R	–
0x4000 0448	T16B_1CS (T16B Ch.1 Counter Status Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPI3	0	H0	R	
		4	CAPI2	0	H0	R	
		3	CAPI1	0	H0	R	
		2	CAPI0	0	H0	R	
		1	UP_DOWN	1	H0	R	
		0	BSY	0	H0	R	
0x4000 044a	T16B_1INTF (T16B Ch.1 Interrupt Flag Register)	15–10	–	0x00	–	R	–
		9	CAPOW3IF	0	H0	R/W	Cleared by writing 1.
		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	
0x4000 044c	T16B_1INTE (T16B Ch.1 Interrupt Enable Register)	15–10	–	0x00	–	R	–
		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	
0x4000 044e	T16B_1MZDMAEN (T16B Ch.1 Counter Max/Zero DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	MZDMAEN[3:0]	0x0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0450	T16B_1CCCTL0 (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	
0x4000 0452	T16B_1CCR0 (T16B Ch.1 Compare/ Capture 0 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 0454	T16B_1CC0DMAEN (T16B Ch.1 Compare/ Capture 0 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC0DMAEN[3:0]	0x0	H0	R/W	
0x4000 0458	T16B_1CCCTL1 (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	
0x4000 045a	T16B_1CCR1 (T16B Ch.1 Compare/ Capture 1 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 045c	T16B_1CC1DMAEN (T16B Ch.1 Compare/ Capture 1 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC1DMAEN[3:0]	0x0	H0	R/W	
0x4000 0460	T16B_1CCCTL2 (T16B Ch.1 Compare/ Capture 2 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	
0x4000 0462	T16B_1CCR2 (T16B Ch.1 Compare/ Capture 2 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 0464	T16B_1CC2DMAEN (T16B Ch.1 Compare/ Capture 2 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC2DMAEN[3:0]	0x0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0468	T16B_1CCCTL3 (T16B Ch.1 Compare/ Capture 3 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	
0x4000 046a	T16B_1CCR3 (T16B Ch.1 Compare/ Capture 3 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 046c	T16B_1CC3DMAEN (T16B Ch.1 Compare/ Capture 3 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC3DMAEN[3:0]	0x0	H0	R/W	

0x4000 0480–0x4000 048c 16-bit Timer (T16) Ch.3

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0480	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	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x4000 0482	T16_3MOD (T16 Ch.3 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4000 0484	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	
		0	MODEN	0	H0	R/W	
0x4000 0486	T16_3TR (T16 Ch.3 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4000 0488	T16_3TC (T16 Ch.3 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x4000 048a	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.
0x4000 048c	T16_3INTE (T16 Ch.3 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x4000 04a0–0x4000 04ac 16-bit Timer (T16) Ch.4

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 04a0	T16_4CLK (T16 Ch.4 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	
0x4000 04a2	T16_4MOD (T16 Ch.4 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 04a4	T16_4CTL (T16 Ch.4 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	
0x4000 04a6	T16_4TR (T16 Ch.4 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4000 04a8	T16_4TC (T16 Ch.4 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x4000 04aa	T16_4INTF (T16 Ch.4 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x4000 04ac	T16_4INTE (T16 Ch.4 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x4000 04c0–0x4000 04cc 16-bit Timer (T16) Ch.5

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 04c0	T16_5CLK (T16 Ch.5 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	
0x4000 04c2	T16_5MOD (T16 Ch.5 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4000 04c4	T16_5CTL (T16 Ch.5 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	
0x4000 04c6	T16_5TR (T16 Ch.5 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4000 04c8	T16_5TC (T16 Ch.5 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x4000 04ca	T16_5INTF (T16 Ch.5 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x4000 04cc	T16_5INTE (T16 Ch.5 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x4000 0600–0x4000 0614 UART (UART3) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0600	UART3_1CLK (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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0602	UART3_1MOD (UART3 Ch.1 Mode Register)	15–13	–	0x00	–	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	
0x4000 0604	UART3_1BR (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	
0x4000 0606	UART3_1CTL (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	
0x4000 0608	UART3_1TXD (UART3 Ch.1 Transmit Data Register)	15–8	–	0x00	–	R	–
		7–0	TXD[7:0]	0x00	H0	R/W	
0x4000 060a	UART3_1RXD (UART3 Ch.1 Receive Data Register)	15–8	–	0x00	–	R	–
		7–0	RXD[7:0]	0x00	H0	R	
0x4000 060c	UART3_1INTF (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 UART3_1RXD 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 UART3_1RXD register.
		1	RB1FIF	0	H0/S0	R	
		0	TBEIF	1	H0/S0	R	Cleared by writing to the UART3_1TXD register.
0x4000 060e	UART3_1INTE (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	
0x4000 0610	UART3_1 TBEDMAEN (UART3 Ch.1 Transmit Buffer Empty DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	TBEDMAEN[3:0]	0x0	H0	R/W	
0x4000 0612	UART3_1 RB1FDMAEN (UART3 Ch.1 Receive Buffer One Byte Full DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	RB1FDMAEN[3:0]	0x0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0614	UART3_1CAWF (UART3 Ch.1 Carrier Waveform Register)	15–8	–	0x00	–	R	–
		7–0	CRPER[7:0]	0x00	H0	R/W	

0x4000 0660–0x4000 066c 16-bit Timer (T16) Ch.6

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0660	T16_6CLK (T16 Ch.6 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	
0x4000 0662	T16_6MOD (T16 Ch.6 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4000 0664	T16_6CTL (T16 Ch.6 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	
0x4000 0666	T16_6TR (T16 Ch.6 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4000 0668	T16_6TC (T16 Ch.6 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x4000 066a	T16_6INTF (T16 Ch.6 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x4000 066c	T16_6INTE (T16 Ch.6 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x4000 0670–0x4000 067e Synchronous Serial Interface (SPIA) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0670	SPIA_1MOD (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	
0x4000 0672	SPIA_1CTL (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	
0x4000 0674	SPIA_1TXD (SPIA Ch.1 Transmit Data Register)	15–0	TXD[15:0]	0x0000	H0	R/W	–
0x4000 0676	SPIA_1RXD (SPIA Ch.1 Receive Data Register)	15–0	RXD[15:0]	0x0000	H0	R	–

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0678	SPIA_1INTF (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 SPIA_1RXD register.
		0	TBEIF	1	H0/S0	R	Cleared by writing to the SPIA_1TXD register.
0x4000 067a	SPIA_1INTE (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	
0x4000 067c	SPIA_1TBEDMAEN (SPIA Ch.1 Transmit Buffer Empty DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	TBEDMAEN[3:0]	0x0	H0	R/W	
0x4000 067e	SPIA_1RBFDMAEN (SPIA Ch.1 Receive Buffer Full DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	RBFDMAEN[3:0]	0x0	H0	R/W	

0x4000 0680–0x4000 068c 16-bit Timer (T16) Ch.2

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0680	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	
0x4000 0682	T16_2MOD (T16 Ch.2 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4000 0684	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	
0x4000 0686	T16_2TR (T16 Ch.2 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4000 0688	T16_2TC (T16 Ch.2 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x4000 068a	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.
0x4000 068c	T16_2INTE (T16 Ch.2 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x4000 06c0–0x4000 06d6 I²C (I2C) Ch.1

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 06c0	I2C_1CLK (I2C 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	
0x4000 06c2	I2C_1MOD (I2C Ch.1 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	
0x4000 06c4	I2C_1BR (I2C Ch.1 Baud-Rate Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6–0	BRT[6:0]	0x7f	H0	R/W	
0x4000 06c8	I2C_1OADR (I2C Ch.1 Own Address Register)	15–10	–	0x00	–	R	–
		9–0	OADR[9:0]	0x000	H0	R/W	
0x4000 06ca	I2C_1CTL (I2C Ch.1 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	
0x4000 06cc	I2C_1TXD (I2C Ch.1 Transmit Data Register)	15–8	–	0x00	–	R	–
		7–0	TXD[7:0]	0x00	H0	R/W	
0x4000 06ce	I2C_1RXD (I2C Ch.1 Receive Data Register)	15–8	–	0x00	–	R	–
		7–0	RXD[7:0]	0x00	H0	R	
0x4000 06d0	I2C_1INTF (I2C Ch.1 Status and Interrupt Flag Register)	15–13	–	0x0	–	R	–
		12	SDALOW	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 I2C_1RXD register.
		0	TBEIF	0	H0/S0	R	Cleared by writing to the I2C_1TXD register.
0x4000 06d2	I2C_1INTE (I2C Ch.1 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7	BYTEENDIE	0	H0	R/W	
		6	GCIE	0	H0	R/W	
		5	NACKIE	0	H0	R/W	
		4	STOPIE	0	H0	R/W	
		3	STARTIE	0	H0	R/W	
		2	ERRIE	0	H0	R/W	
		1	RBFIE	0	H0	R/W	
		0	TBEIE	0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 06d4	I2C_1TBEDMAEN (I2C Ch.1 Transmit Buffer Empty DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	TBEDMAEN[3:0]	0x0	H0	R/W	
0x4000 06d6	I2C_1RBFDMAEN (I2C Ch.1 Receive Buffer Full DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	RBFDMAEN[3:0]	0x0	H0	R/W	

0x4000 0700–0x4000 070c Sound Generator (SNDA)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0700	SNDACLK (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	
0x4000 0702	SNDASEL (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	
0x4000 0704	SNDACTL (SNDA Control Register)	15–9	–	0x00	–	R	–
		8	SSTP	0	H0	R/W	
		7–1	–	0x00	–	R	
		0	MODEN	0	H0	R/W	
0x4000 0706	SNDADAT (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	
0x4000 0708	SNDAINTF (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 SNDADAT register.
		0	EDIF	0	H0	R/W	Cleared by writing 1 or writing to the SNDADAT register.
0x4000 070a	SNDAINTE (SNDA Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	EMIE	0	H0	R/W	
		0	EDIE	0	H0	R/W	
0x4000 070c	SNDAEMDMAEN (SNDA Sound Buffer Empty DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	EMDMAEN[3:0]	0x0	H0	R/W	

0x4000 0720–0x4000 0732 IR Remote Controller (REMC3)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0720	REMC3CLK (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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0722	REMC3DBCTL (REMC3 Data Bit Counter Control Register)	15–10	–	0x00	–	R	–
		9	PRESET	0	H0/S0	R/W	Cleared by writing 1 to the REMC3DBCTL.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	
0x4000 0724	REMC3DBCNT (REMC3 Data Bit Counter Register)	15–0	DBCNT[15:0]	0x0000	H0/S0	R	Cleared by writing 1 to the REMC3DBCTL.REMCRST bit.
0x4000 0726	REMC3APLEN (REMC3 Data Bit Active Pulse Length Register)	15–0	APLEN[15:0]	0x0000	H0	R/W	Writing enabled when REMC3DBCTL.MODEN bit = 1.
0x4000 0728	REMC3DBLEN (REMC3 Data Bit Length Register)	15–0	DBLEN[15:0]	0x0000	H0	R/W	Writing enabled when REMC3DBCTL.MODEN bit = 1.
0x4000 072a	REMC3INTF (REMC3 Status and Interrupt Flag Register)	15–11	–	0x00	–	R	–
		10	DBCNTRUN	0	H0/S0	R	Cleared by writing 1 to the REMC3DBCTL.REMCRST bit.
		9	DBLENBSY	0	H0	R	Effective when the REMC3DBCTL.BUFEN bit = 1.
		8	APLENBSY	0	H0	R	
		7–2	–	0x00	–	R	–
		1	DBIF	0	H0/S0	R/W	Cleared by writing 1 to this bit or the REMC3DBCTL. REMCRST bit.
		0	APIF	0	H0/S0	R/W	
0x4000 072c	REMC3INTE (REMC3 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	DBIE	0	H0	R/W	
		0	APIE	0	H0	R/W	
0x4000 0730	REMC3CARR (REMC3 Carrier Waveform Register)	15–8	CRDTY[7:0]	0x00	H0	R/W	–
		7–0	CRPER[7:0]	0x00	H0	R/W	
0x4000 0732	REMC3CCTL (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	

0x4000 0740–0x4000 076c 16-bit PWM Timer (T16B) Ch.2

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0740	T16B_2CLK (T16B 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	–	0	–	R	
		2–0	CLKSRC[2:0]	0x0	H0	R/W	
0x4000 0742	T16B_2CTL (T16B Ch.2 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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0744	T16B_2MC (T16B Ch.2 Max Counter Data Register)	15–0	MC[15:0]	0xffff	H0	R/W	–
0x4000 0746	T16B_2TC (T16B Ch.2 Timer Counter Data Register)	15–0	TC[15:0]	0x0000	H0	R	–
0x4000 0748	T16B_2CS (T16B Ch.2 Counter Status Register)	15–8	–	0x00	–	R	–
		7–6	–	0x0	–	R	
		5	CAPI3	0	H0	R	
		4	CAPI2	0	H0	R	
		3	CAPI1	0	H0	R	
		2	CAPI0	0	H0	R	
		1	UP_DOWN	1	H0	R	
		0	BSY	0	H0	R	
0x4000 074a	T16B_2INTF (T16B Ch.2 Interrupt Flag Register)	15–10	–	0x00	–	R	–
		9	CAPOW3IF	0	H0	R/W	Cleared by writing 1.
		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	
0x4000 074c	T16B_2INTE (T16B Ch.2 Interrupt Enable Register)	15–10	–	0x00	–	R	–
		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	
0x4000 074e	T16B_2MZDMAEN (T16B Ch.2 Counter Max/Zero DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	MZDMAEN[3:0]	0x0	H0	R/W	
0x4000 0750	T16B_2CCCTL0 (T16B Ch.2 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	
0x4000 0752	T16B_2CCR0 (T16B Ch.2 Compare/Capture 0 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 0754	T16B_2CC0DMAEN (T16B Ch.2 Compare/Capture 0 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC0DMAEN[3:0]	0x0	H0	R/W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0758	T16B_2CCCTL1 (T16B Ch.2 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	
0x4000 075a	T16B_2CCR1 (T16B Ch.2 Compare/ Capture 1 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 075c	T16B_2CC1DMAEN (T16B Ch.2 Compare/ Capture 1 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC1DMAEN[3:0]	0x0	H0	R/W	
0x4000 0760	T16B_2CCCTL2 (T16B Ch.2 Compare/ Capture 2 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	
0x4000 0762	T16B_2CCR2 (T16B Ch.2 Compare/ Capture 2 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 0764	T16B_2CC2DMAEN (T16B Ch.2 Compare/ Capture 2 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC2DMAEN[3:0]	0x0	H0	R/W	
0x4000 0768	T16B_2CCCTL3 (T16B Ch.2 Compare/ Capture 3 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	
0x4000 076a	T16B_2CCR3 (T16B Ch.2 Compare/ Capture 3 Data Register)	15–0	CC[15:0]	0x0000	H0	R/W	–
0x4000 076c	T16B_2CC3DMAEN (T16B Ch.2 Compare/ Capture 3 DMA Request Enable Register)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	CC3DMAEN[3:0]	0x0	H0	R/W	

0x4000 0780–0x4000 078c 16-bit Timer (T16) Ch.7

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0780	T16_7CLK (T16 Ch.7 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	
0x4000 0782	T16_7MOD (T16 Ch.7 Mode Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	TRMD	0	H0	R/W	
0x4000 0784	T16_7CTL (T16 Ch.7 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	
0x4000 0786	T16_7TR (T16 Ch.7 Reload Data Register)	15–0	TR[15:0]	0xffff	H0	R/W	–
0x4000 0788	T16_7TC (T16 Ch.7 Counter Data Register)	15–0	TC[15:0]	0xffff	H0	R	–
0x4000 078a	T16_7INTF (T16 Ch.7 Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIF	0	H0	R/W	Cleared by writing 1.
0x4000 078c	T16_7INTE (T16 Ch.7 Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	UFIE	0	H0	R/W	

0x4000 07a0–0x4000 07bc 12-bit A/D Converter (ADC12A) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 07a2	ADC12A_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	
0x4000 07a4	ADC12A_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	
		2–0	SMPCLK[2:0]	0x7	H0	R/W	
0x4000 07a6	ADC12A_0CFG (ADC12A Ch.0 Configuration Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1–0	VRANGE[1:0]	0x0	H0	R/W	
0x4000 07a8	ADC12A_0INTF (ADC12A Ch.0 Interrupt Flag Register)	15–9	–	0x00	–	R	–
		8	OVIF	0	H0	R/W	Cleared by writing 1.
		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	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 07aa	ADC12A_0INTE (ADC12A Ch.0 Interrupt Enable Register)	15–9	–	0x00	–	R	–
		8	OVIE	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	
0x4000 07ac	ADC12A_0DMAEN0 (ADC12A Ch.0 DMA Request Enable Register 0)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	ADCDMAEN[3:0]	0x0	H0	R/W	
0x4000 07ae	ADC12A_0DMAEN1 (ADC12A Ch.0 DMA Request Enable Register 1)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	ADCDMAEN[3:0]	0x0	H0	R/W	
0x4000 07b0	ADC12A_0DMAEN2 (ADC12A Ch.0 DMA Request Enable Register 2)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	ADCDMAEN[3:0]	0x0	H0	R/W	
0x4000 07b2	ADC12A_0DMAEN3 (ADC12A Ch.0 DMA Request Enable Register 3)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	ADCDMAEN[3:0]	0x0	H0	R/W	
0x4000 07b4	ADC12A_0DMAEN4 (ADC12A Ch.0 DMA Request Enable Register 4)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	ADCDMAEN[3:0]	0x0	H0	R/W	
0x4000 07b6	ADC12A_0DMAEN5 (ADC12A Ch.0 DMA Request Enable Register 5)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	ADCDMAEN[3:0]	0x0	H0	R/W	
0x4000 07b8	ADC12A_0DMAEN6 (ADC12A Ch.0 DMA Request Enable Register 6)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	ADCDMAEN[3:0]	0x0	H0	R/W	
0x4000 07ba	ADC12A_0DMAEN7 (ADC12A Ch.0 DMA Request Enable Register 7)	15–8	–	0x00	–	R	–
		7–4	–	0x0	–	R	
		3–0	ADCDMAEN[3:0]	0x0	H0	R/W	
0x4000 07bc	ADC12A_0ADD (ADC12A Ch.0 Result Register)	15–0	ADD[15:0]	0x0000	H0	R	–

0x4000 07c0–0x4000 07c2 Temperature Sensor/Reference Voltage Generator (TSRVR) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 07c0	TSRVR_0TCTL (TSRVR Ch.0 Temperature Sensor Control Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	H0	R	
		0	TEMPEN	0	H0	R/W	
0x4000 07c2	TSRVR_0VCTL (TSRVR Ch.0 Reference Voltage Generator Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	H0	R	
		1–0	VREFAMD[1:0]	0x0	H0	R/W	

0x4000 0800–0x4000 0812 LCD Driver (LCD8D)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0800	LCD8DCLK (LCD8D Clock Control Register)	15–9	–	0x00	–	R	–
		8	DBRUN	1	H0	R/W	
		7	–	0	–	R	
		6–4	CLKDIV[2:0]	0x0	H0	R/W	
		3–2	–	0x0	–	R	
		1–0	CLKSRC[1:0]	0x0	H0	R/W	
0x4000 0802	LCD8DCTL (LCD8D Control Register)	15–8	–	0x00	–	R	–
		7–2	–	0x00	–	R	
		1	LCDDIS	0	H0	R/W	
		0	MODEN	0	H0	R/W	
0x4000 0804	LCD8DTIM1 (LCD8D Timing Control Register 1)	15–13	–	0x0	–	R	–
		12–8	FRMCNT[4:0]	0x01	H0	R/W	
		7–6	–	0x0	–	R	
		5	(reserved)	0	H0	R/W	
		4–3	–	0x0	–	R	
		2–0	LDUTY[2:0]	0x1f	H0	R/W	
0x4000 0806	LCD8DTIM2 (LCD8D Timing Control Register 2)	15	LCDWAVE	0	H0	R/W	–
		14–10	–	0x00	–	R	
		9–8	BSTC[1:0]	0x1	H0	R/W	
		7–3	–	0x00	–	R	
		2–0	NLINE[2:0]	0x0	H0	R/W	
0x4000 0808	LCD8DPWR (LCD8D Power Control Register)	15	EXVCSEL	1	H0	R/W	–
		14–13	RESISEL[1:0]	0x0	H0	R/W	
		12–8	LC[4:0]	0x00	H0	R/W	
		7–5	–	0x0	–	R	
		4	BSTEN	0	H0	R/W	
		3	BIASSEL	1	H0	R/W	
		2	HVLD	0	H0	R/W	
		1	VCSEL	0	H0	R/W	
		0	VCEN	0	H0	R/W	
0x4000 080a	LCD8DDSP (LCD8D Display Control Register)	15–8	–	0x00	–	R	–
		7	–	0	–	R	
		6	SEGREV	1	H0	R/W	
		5	COMREV	1	H0	R/W	
		4	DSPREV	1	H0	R/W	
		3	–	0	–	R	
		2	DSPAR	0	H0	R/W	
		1–0	DSPC[1:0]	0x0	H0	R/W	
0x4000 080c	LCD8DCOMC0 (LCD8D COM Pin Control Register 0)	15–8	–	0x00	–	R	–
		7	COM7DEN	1	H0	R/W	
		6	COM6DEN	1	H0	R/W	
		5	COM5DEN	1	H0	R/W	
		4	COM4DEN	1	H0	R/W	
		3	COM3DEN	1	H0	R/W	
		2	COM2DEN	1	H0	R/W	
		1	COM1DEN	1	H0	R/W	
		0	COM0DEN	1	H0	R/W	
0x4000 0810	LCD8DINTF (LCD8D Interrupt Flag Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	FRMIF	0	H0	R/W	Cleared by writing 1.
0x4000 0812	LCD8DINTE (LCD8D Interrupt Enable Register)	15–8	–	0x00	–	R	–
		7–1	–	0x00	–	R	
		0	FRMIE	0	H0	R/W	

0x4000 0840–0x4000 0850 R/F Converter (RFC) Ch.0

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 0840	RFC_0CLK (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	
0x4000 0842	RFC_0CTL (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	(reserved)	0x0	–	R/W	
		3–1	–	0x0	–	R	
		0	MODEN	0	H0	R/W	
0x4000 0844	RFC_0TRG (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	
0x4000 0846	RFC_0MCL (RFC Ch.0 Measurement Counter Low Register)	15–0	MC[15:0]	0x0000	H0	R/W	–
0x4000 0848	RFC_0MCH (RFC Ch.0 Measurement Counter High Register)	15–8	–	0x00	–	R	–
		7–0	MC[23:16]	0x00	H0	R/W	
0x4000 084a	RFC_0TCL (RFC Ch.0 Time Base Counter Low Register)	15–0	TC[15:0]	0x0000	H0	R/W	–
0x4000 084c	RFC_0TCH (RFC Ch.0 Time Base Counter High Register)	15–8	–	0x00	–	R	–
		7–0	TC[23:16]	0x00	H0	R/W	
0x4000 084e	RFC_0INTF (RFC Ch.0 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	
0x4000 0850	RFC_0INTE (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	

0x4000 1000–0x4000 2014 DMA Controller (DMAC)

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 1000	DMACSTAT (DMAC Status Register)	31–24	–	0x00	–	R	–
		23–21	–	0x0	–	R	
		20–16	CHNLS[4:0]	*	H0	R	* Number of channels implemented - 1
		15–8	–	0x00	–	R	–
		7–4	STATE[3:0]	0x0	H0	R	
		3–1	–	0x0	–	R	
		0	MSTENSTAT	0	H0	R	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 1004	DMACCFG (DMAC Configuration Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–1	–	0x00	–	R	
		0	MSTEN	–	–	W	
0x4000 1008	DMACCPTR (DMAC Control Data Base Pointer Register)	31–7	CPTR[31:7]	0x000 0000	H0	R/W	–
		6–0	CPTR[6:0]	0x00	H0	R	
0x4000 100c	DMACACPTR (DMAC Alternate Control Data Base Pointer Register)	31–0	ACPTR[31:0]	–	H0	R	–
0x4000 1014	DMACSWREQ (DMAC Software Request Register)	31–24	–	–	–	R	–
		23–16	–	–	–	R	
		15–8	–	–	–	R	
		7–4	–	–	–	R	
		3–0	SWREQ[3:0]	–	–	W	
0x4000 1020	DMACRMSET (DMAC Request Mask Set Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–4	–	0x0	–	R	
		3–0	RMSET[3:0]	0x0	H0	R/W	
0x4000 1024	DMACRMCLR (DMAC Request Mask Clear Register)	31–24	–	–	–	R	–
		23–16	–	–	–	R	
		15–8	–	–	–	R	
		7–4	–	–	–	R	
		3–0	RMCLR[3:0]	–	–	W	
0x4000 1028	DMACENSET (DMAC Enable Set Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–4	–	0x0	–	R	
		3–0	ENSET[3:0]	0x0	H0	R/W	
0x4000 102c	DMACENCLR (DMAC Enable Clear Register)	31–24	–	–	–	R	–
		23–16	–	–	–	R	
		15–8	–	–	–	R	
		7–4	–	–	–	R	
		3–0	ENCLR[3:0]	–	–	W	
0x4000 1030	DMACPASET (DMAC Primary-Alternate Set Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–4	–	0x0	–	R	
		3–0	PASET[3:0]	0x0	H0	R/W	
0x4000 1034	DMACPACLR (DMAC Primary-Alternate Clear Register)	31–24	–	–	–	R	–
		23–16	–	–	–	R	
		15–8	–	–	–	R	
		7–4	–	–	–	R	
		3–0	PACLR[3:0]	–	–	W	
0x4000 1038	DMACPRSET (DMAC Priority Set Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–4	–	0x0	–	R	
		3–0	PRSET[3:0]	0x0	H0	R/W	
0x4000 103c	DMACPRCLR (DMAC Priority Clear Register)	31–24	–	–	–	R	–
		23–16	–	–	–	R	
		15–8	–	–	–	R	
		7–4	–	–	–	R	
		3–0	PRCLR[3:0]	–	–	W	

Address	Register name	Bit	Bit name	Initial	Reset	R/W	Remarks
0x4000 104c	DMACERRIF (DMAC Error Interrupt Flag Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–1	–	0x00	–	R	
		0	ERRIF	0	H0	R/W	Cleared by writing 1.
0x4000 2000	DMACENDIF (DMAC Transfer Completion Interrupt Flag Register)	31–24	–	0x00	–	R	
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–4	–	0x0	–	R	
		3–0	ENDIF[3:0]	0x0	H0	R/W	Cleared by writing 1.
0x4000 2008	DMACENDIESET (DMAC Transfer Completion Interrupt Enable Set Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–4	–	0x0	–	R	
		3–0	ENDIESET[3:0]	0x0	H0	R/W	
0x4000 200c	DMACENDIECLR (DMAC Transfer Completion Interrupt Enable Clear Register)	31–24	–	–	–	R	–
		23–16	–	–	–	R	
		15–8	–	–	–	R	
		7–4	–	–	–	R	
		3–0	ENDIECLR[3:0]	–	–	W	
0x4000 2010	DMACERRIESET (DMAC Error Interrupt Enable Set Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–1	–	0x00	–	R	
		0	ERRIESET	0	H0	R/W	
0x4000 2014	DMACERRIECLR (DMAC Error Interrupt Enable Clear Register)	31–24	–	0x00	–	R	–
		23–16	–	0x00	–	R	
		15–8	–	0x00	–	R	
		7–1	–	0x00	–	R	
		0	ERRIECLR	–	–	W	

Appendix B Power Saving

Current consumption will vary dramatically, depending on CPU operating mode, operation clock frequency, peripheral circuits being operated, and V_{D1} regulator 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	↑	Economy	OFF	OFF	OFF	SLEEP	I_{SLP1-2}
Clock counting	Low		ON		ON	SLEEP with OSC1SLPC	I_{SLP3-4}
Low-speed processing						OSC1 RUN	I_{RUN6-9}
Peripheral circuit operations		Normal		ON		SLEEP or HALT	$I_{HALT1-2}$
High-speed processing	High ↓					IOSC/OSC3/EXOSC RUN	I_{RUN1-5}

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.

PWGACTL.REGMODE[1:0] bits of the power generator

If the PWGACTL.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 PWGACTL.REGMODE[1:0] bits to 0x3 (economy mode) or 0x0 (automatic mode) before placing the CPU into SLEEP mode.

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 CPU enters SLEEP mode. 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 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.

B.2 Other Power Saving Methods

Supply voltage detector configuration

Continuous operation mode (SVD4_*n*CTL.SVDMD[1:0] bits = 0x0) always detects the power supply voltage, therefore, it increases current consumption. Set the supply voltage detector to intermittent operation mode or turn it on only when required.

LCD driver configurations

- Setting the LCD voltage regulator to operate with the V_{C1} reference voltage (LCD8DPWR.VCSEL bit = 0) increases current consumption. If a desired LCD drive voltage can be obtained, operating with V_{C2} reference voltage (LCD8DPWR.VCSEL bit = 1) is recommended.
- The lower booster clock frequency setting (LCD8DTIM2.BSTC[1:0] bits) for the LCD voltage booster decreases current consumption. Note, however, that the load characteristic becomes worse.
- Setting the LCD voltage regulator into heavy load protection mode (LCD8DPWR.HVLD bit = 1) increases current consumption. Heavy load protection mode should be set only when the display becomes unstable.

Appendix C Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

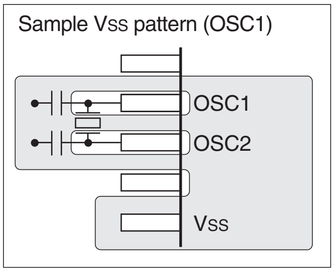
OSC1/OSC3 oscillator circuit

- 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 V_{SS} 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.

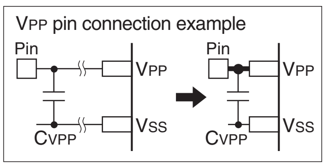
Failure to observe precautions (1) to (3) adequately may lead to noise in OSC1CLK and jitter in OSC3CLK. Noise in the OSC1CLK will destabilize timers that use OSC1CLK as well as CPU operations. Jitter in the OSC3 output will reduce operating frequencies.

#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.

V_{PP} pin

Connect a capacitor C_{VPP} between the V_{SS} and V_{PP} pins to suppress fluctuations within V_{PP} ± 1 V. The C_{VPP} should be placed as close to the V_{PP} pin as possible and use a sufficiently thick wiring pattern that allows current of several tens of mA to flow.

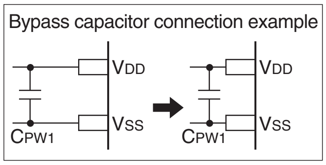


Power supply circuit

Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

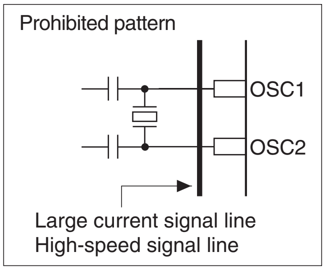
(1) Connections from the power supply to the V_{DD} and V_{SS} pins should be implemented via the shortest, thickest patterns possible.

(2) If a bypass capacitor is connected between V_{DD} and V_{SS}, connections between the V_{DD} and V_{SS} pins should be as short as possible.



Signal line location

- 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.
- The SEG/COM lines and voltage boost capacitor drive lines are more likely to generate noise, therefore keep a distance between the lines and pins susceptible to noise.



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).

(3) V_{C1-3}, C_{P1-2}, SEG*x*, and COM*x* pins
If the LCD driver is not used, these pins should be left open. The control registers should be fixed at the initial status (display off). The unused SEG*x* and COM*x* pins that are not required to connect should be left open even if the LCD driver is used.

Miscellaneous

Minor variations over time may result in electrical damage arising from disturbances in the form of voltages exceeding the absolute maximum rating when mounting the product in addition to physical damage. The following factors can give rise to these variations:

(1) Electromagnetically-induced noise from industrial power supplies used in mounting reflow, reworking after mounting, and individual characteristic evaluation (testing) processes
(2) Electromagnetically-induced noise from a solder iron when soldering

In particular, during soldering, take care to ensure that the soldering iron GND (tip potential) has the same potential as the IC GND.

Appendix D Measures Against Noise

To improve noise immunity, take measures against noise as follows:

Noise Measures for V_{DD} and V_{SS} Power Supply Pins

When noise falling below the rated voltage is input, an IC malfunction may occur. If desired operations cannot be achieved, take measures against noise on the circuit board, such as designing close patterns for circuit board power supply circuits, adding noise-filtering decoupling capacitors, and adding surge/noise prevention components on the power supply line.
For the recommended patterns on the circuit board, see “Mounting Precautions” in Appendix.

Noise Measures for #RESET Pin

If noise is input to the #RESET pin, the IC may be reset. Therefore, the circuit board must be designed properly taking noise measures into consideration.
For the recommended patterns on the circuit board, see “Mounting Precautions” in Appendix.

Noise Measures for 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 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.

Revision History

Code No.	Page	Contents
414063300	All	New establishment
414063301	Whole manual	Corrected the Cortex®-M0+ register names. System control register → Cortex®-M0+ System Control Register or Cortex®-M0+ Application Interrupt and Reset Control Register Vector table offset register → Cortex®-M0+ Vector Table Offset Register (VTOR) System handler priority registers → Cortex®-M0+ System Handler Priority Registers Interrupt priority registers → Cortex®-M0+ Interrupt Priority Registers Correcred the Cortex®-M0+ manual names. Cortex®-M0+ Technical Reference Manual → ARM®v6-M Architecture Reference Manual、 Cortex®-M0+Technical Reference Manual、 or the documents introduced in Section 3.4, such as “Cortex®-M0+ Devices Generic User Guide” COU core → CPU
	1-1 to 3	1.1 Features Added the following annotations to Table 1.1.1. I²C (I2C) *1 *1 The input filter in I2C (SDA and SCL inputs) does not comply with the standard for removing noise spikes less than 50 ns. *2 SLEEP mode refers to deep sleep mode in the Cortex®-M0+ processor. The RAM retains data even in SLEEP mode. Modified Table 1.1.1. Instruction cache: Deleted Package: The JEITA package name was corrected, LQFP → TQFP.
	1-3	1.2 Block Diagram Deleted Cache controller and Cache RAM from Figure 1.2.1.
	2-16	2.4.2 Transition between Operating Modes SLEEP mode Added the following description: The RAM retains data even in SLEEP mode.
	3-2	Added a new section. 3.4 Reference Documents
	4-2	4.3.1 Flash Memory Pin Deleted the following description and note: ~~For the VPP voltage, refer to “Recommended Operating Conditions, Flash programming voltage VPP” in the “Electrical Characteristics” chapter.~~ ~~Note: Always leave the VPP pin open except when programming the Flash memory.~~
	4-3	4.6 Peripheral Circuit Control Registers Deleted the Cache controller register from Table 4.6.1.
	4-9	4.7 Instruction Cache Deleted the section.
	4-10	4.8 Control Registers FLASHC Flash Read Cycle Register Added a not to the RDWAIT[1:0] bit. Notes: ... • When the FLASHCWAIT.RDWAIT[1:0] bit setting is altered from 0x2 to 0x1, add two NOP instructions immediately after that. Program example: FLASHC->WAIT_b.RDWAIT = 1; asm(“NOP”); asm(“NOP”); CLG->OSC_b.IOSCEN = 0;
	9-4	9.4 Control Registers WDT2 Control Register Corrected the description of the WDTRUN[3:0] bit. 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

REVISION HISTORY

Code No.	Page	Contents
414063301	10-4	10.4.2 Real-Time Clock Counter Operations Corrective operation when a value out of the effective range is set Added a note. Note: Do not set the RTCMON.RTCMOL[3:0] bits to 0x0 if the RTCMON.RTCMOH bit = 0.
	10-11	10.6 Control Registers RTCA Month/Day Register Bit 12 RTCMOH Bits 11–8 RTCMOL[3:0] Added a note. Notes: ... • Be sure to avoid setting the RTCAMON.RTCMOH/RTCMOL[3:0] bits to 0x00.
	15-1	15.1 Overview Added the following description: • The input filter for the SDA and SCL inputs does not comply with the standard for removing noise spikes less than 50 ns.
	15-7, 9	15.4.3 Data Reception in Master Mode Data receiving procedure Added Step 1. (The old step numbers were carried down in order.) 1. When receiving one-byte data, write 1 to the I2C_*n*CTL.TXNACK bit. Modified Figure 15.4.3.2. A flow for Step 1 was added.
	15-9	15.4.3 Data Reception in Master Mode Data reception using DMA Corrected the description. This automates the data receiving procedure Steps 6, 8, and 10 described above.
	15-13 to 14	15.4.6 Data Reception in Slave Mode Data receiving procedure Added Step 1. (The old step numbers were carried down in order.) 1. When receiving one-byte data, write 1 to the I2C_*n*CTL.TXNACK bit. Modified Figure 15.4.6.2. A flow for Step 1 was added.
	16-5	16.4.2 Counter Block Operations MAX counter data register Added a note. 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.
	23-1	23.2 Recommended Operating Conditions Added “(V_{SS} = 0 V) ∗1” and the following annotations: ∗1 The potential variation of the V_{SS} voltage should be suppressed to within ±0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count). ∗6 The component values should be determined after evaluating operations using an actual mounting board.
	23-7	23.6 Flash Memory Characteristics Added an annotation. ∗1 The potential variation of the V_{SS} voltage should be suppressed to within ±0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count).
	25-1	25 Package The JEITA package name was corrected, LQFP → TQFP.
	AP-A-3	Appendix A List of Peripheral Circuit Control Registers Deleted the CACHE Control Register.
	AP-D-1	Appendix D Measures Against Noise Added a description. Noise Measures for Input Pins Connected to Signal with High Driving Capability Such As Power Supply
414063302	Cover	Replaced the EPSON logo.
	Back of cover	Replaced the NOTICE.
	1-3, 1-4	Changed the packge name. TQFP14 was deleted and TQFP15 was added.
	25-1	Changed the package name and replaced the figure. TQFP14 was deleted and TQFP15 was added.

EPSON

International Sales Operations

America

Epson America, Inc.

Headquarter:
3131 Katella Ave.,
Los Alamitos, CA 90720, USA
Phone: +1-800-463-7766

San Jose Office:
2860 Zanker Road Suite 204,
San Jose, CA 95134, USA
Phone: +1-800-463-7766

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 601-603, Building A One East, No. 325 East Longhua
Road, Shanghai 200023, 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: +86-755-3299-0588 Fax: +86-755-3299-0560

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.

438B Alexandra Road,
Block B Alexandra TechnoPark, #04-01/04, Singapore 119968
Phone: +65-6586-5500 Fax: +65-6271-7066

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

MD Sales & Marketing Department

JR Shinjuku Miraina Tower, 4-1-6 Shinjuku, Shinjuku-ku,
Tokyo 160-8801, Japan

Document Code: 414063302
First Issue August 2020
Revised May 2023 in JAPAN Ⓛ